UNIVERSITÀ DEGLI STUDI DI PISA

Facoltà di Scienze Matematiche, Fisiche e Naturali

Corso di Laurea in Informatica

IDENTIFICAZIONE AUTOMATICA DI MACRO-ATTIVITÀ NEL TRAFFICO DI RETE

Candidato
Emanuele Faranda

Relatore
Luca Deri

Anno Accademico 2015/2016

Indice generale

Capitolo 1 Tirocinio e Obiettivi

1.1 Contesto

Nella società moderna, le informazioni viaggiano in tempo reale da una parte all'altra del pianeta. Le infrastrutture che permettono il transito e lo smistamento dei dati fanno uso di tecnologie sempre più veloci e complesse. Dal piccolo ambito privato, al medio-grande contesto aziendale, fino al vasto ambito di fornitura di servizi a livello mondiale, l'attività di monitoraggio di rete è essenziale per poter individuare guasti, malfunzionamenti, intrusioni e per garantire un'appropriata qualità del servizio.

Oggi buona parte del traffico applicativo che viaggia su Internet è cifrato con algoritmi di crittografia forte, grazie ai quali è possibile lo scambio sicuro di dati tra client e server. La diffusione dell'utilizzo di tecniche di cifratura dei dati, tuttavia, rappresenta un ostacolo per chi si trovi a dover monitorare le attività di rete. Le tradizionali tecnologie DPI, Deep Packet Introspection [59], che utilizzano i dati presenti nei pacchetti di rete per identificarne il protocollo, risultano spesso inefficaci. È quindi indispensabile, per l'immediato futuro, ideare strategie più dinamiche di caratterizzazione del traffico.

Un altro tema cruciale per il monitoraggio è la grande quantità d'informazione che viaggia sulla rete. Dal punto di vista di un amministratore di rete, questa quantità d'informazione, per poter essere correttamente interpretata, deve essere in qualche modo correlata e sintetizzata. L'approccio tradizione prevede l'aggregazione dei pacchetti in flussi di rete. Questo garantisce sì la correlazione tra i dati, ma presenta ancora un elevato grado di complessità, che si traduce, per l'amministratore di rete, in ulteriore lavoro d'interpretazione.

1.2 Obiettivi

Il tirocinio ha come obiettivo lo sviluppo di metodologie, basate sui flussi, per la sintesi del traffico di rete in macro-attività e per la sua caratterizzazione, con particolare attenzione al traffico cifrato. Le informazioni ricavate devono essere opportunamente storicizzate, al fine di permetterne un'elaborazione futura.

Queste funzionalità devono permettere all'amministratore di rete di avere una visione globale dell'attività di un host locale, per poter meglio discernere tra la sua normale attività e potenziali intrusioni o malfunzionamenti.

Sono qui brevemente indicati gli obiettivi prefissati e quelli effettivamente raggiunti.

- ✔ Sintesi generica dei flussi in macro-attività
- ✔ Separazione generica del traffico di rete attivo dal rumore di fondo
- ✔ Rilevamento del protocollo OpenVPN

- ✔ Caratterizzazione attività su flussi IMAPS
- ✔ Rilevamento di traffico multimediale (tramite MIME type)
- ✔ Rilevamento di traffico di navigazione web all'interno di flussi generici SSL
- ✔ Caratterizzazione dell'integrazione di Facebook e Twitter in pagine web
- ✔ Storicizzazione delle informazioni sulle attività
- ✔ Implementazione all'interno di nDPI e ntopng
- ✔ Visualizzazione su grafico delle attività ricavate
- ✗ Rilevamento di traffico VPN generico
- ✗ Caratterizzazione generica di flussi multimediali

Le tecniche di riconoscimento elaborate sono descritte nel dettaglio nel Capitolo 5.

1.3 Modalità di Svolgimento

Nella prima fase del tirocinio sono stati delineati gli obiettivi da conseguire e si è discusso, dopo opportuna documentazione, su quali tecnologie potessero essere impiegate per il conseguimento degli stessi. La seconda fase si è svolta principalmente per via telematica. Tramite posta elettronica si è continuato valutare le scelte proposte e ci si è coordinati per l'implementazione dei primi prototipi. Per discutere di scelte implementative più complesse si è usato il software Skype per la videoconferenza. Si è scelto infine di usare Dropbox come piattaforma di condivisione di contenuti, in particolare grafici e file di cattura di traffico da analizzare.

Un ruolo molto importante ha avuto l'utilizzo della piattaforma github.com, tramite la quale è stato possibile leggere e modificare il codice di ntopng [19] e di nDPI [20]. Dopo aver implementato una quantità consona di funzionalità correlate, infatti, i cambiamenti venivano proposti tramite pull request e successivamente integrati nel progetto software principale.

Capitolo 2 Monitoraggio e Strumenti

2.1 Architettura di Rete

Per poter comprendere meglio il contesto in cui gli strumenti di monitoraggio operano, è necessario introdurre in che modo avviene la comunicazione tra le applicazioni che fanno uso dei servizi di rete. L'infrastruttura volta ad adempiere questo compito viene definita architettura di rete. Grazie al grande lavoro di standardizzazione delle architetture di rete su Internet, oggi si può far riferimento al modello unico TCP/IP, ampliamento diffuso, che ha soppiantato le architetture proprietarie ed il più complesso modello ISO/OSI [62]. Il modello TCP/IP è organizzato in una serie di livelli indipendenti, in quello che viene chiamato stack di rete.

Il livello più basso dello stack di rete è quello fisico. Esso comprende tutte le tecnologie ed i mezzi fisici che trasportano i dati in forma di segnali elettromagnetici. Questo è l'unico livello tangibile dell'architettura. Il livello superiore, detto di datalink, ha ruoli di controllo e di mediazione tra il livello fisico ed il livello superiore. Un livello molto importante nell'ambito del monitoraggio delle reti è quello di rete, supportato ed implementato da tutti i dispositivi attivi di interconnessione. Il quarto strato dell'architettura TCP/IP è il livello di trasporto, che si occupa dell'integrità dell'informazione, dal mittente al destinatario. Infine vi è il livello applicativo, destinato appunto alla definizione di protocolli di alto livello, spesso testuali, per la comunicazione delle applicazioni reali.

I software di monitoraggio delle reti possono trarre grande vantaggio dalla struttura a livelli. Il passaggio da un livello superiore ad uno inferiore, infatti, avviene per mezzo dell'incapsulamento dei dati dello strato superiore, chiamati payload o carico utile, con l'aggiunta di intestazioni specifiche del protocollo. Il software di monitoraggio, in esecuzione su un sistema operativo che utilizza l'architettura TCP/IP, per avere accesso ai dati di cui ha bisogno non deve far altro che spacchettare via via i contenitori di livello inferiore, risalendo idealmente fino al livello d'interesse. Questo compito, sebbene possa sembrare relativamente semplice, nella realtà si complica a causa delle innumerevoli implementazioni esistenti.

Gli strumenti di monitoraggio si appoggiano alle funzionalità offerte dal sistema operativo che, comunicando opportunamente con le schede di rete, mette a disposizione le informazioni di cui questi hanno bisogno.

2.2 Flussi di Rete

Un flusso rappresenta un'aggregazione logica del traffico di rete. I pacchetti di rete, infatti, spesso fanno parte di una più ampia parte di comunicazione di rete. I flussi possono essere considerati l'equivalente artificiale di una chiamata o di una connessione [35]. Per correlare i pacchetti di rete, gli strumenti di monitoraggio basati sui flussi tengono in considerazione alcune informazioni contenute nei pacchetti catturati.

Spesso si aggrega considerando la quadrupla composta da

- interfaccia di rete su cui è avvenuta la cattura
- indirizzo di rete sorgente o destinazione
- porta del livello di trasporto, sorgente o destinazione
- identificatore di VLAN

Anche il traffico UDP, sebbene la sua natura non sia esplicitamente quella di una connessione, può essere aggregato allo stesso modo, considerando la porta su cui i pacchetti transitano. Uno dei parametri da dosare quando si fa aggregazione è il tempo massimo di inattività del flusso prima che la comunicazione sia considerata terminata.

2.3 Infrastruttura Esistente

Il software nDPI [20] implementa la tecnologia di Deep Packet Introspection [59], grazie alla quale è possibile estrarre informazioni dai flussi di rete. nDPI nasce come un'estensione della libreria OpenDPI per supportare nuovi protocolli, estendere la sua compatibilità, adattarla alle nuove esigenze di cattura di pacchetti di rete. Il codice è disponibile online [20] sotto licenza LGPv3. nDPI attualmente supporta più di 210 protocolli e servizi tra quelli più conosciuti ed utilizzati.

Ntopng [19] è un software di monitoraggio delle reti per la cattura e l'elaborazione di pacchetti ad alta velocità ideato da Luca Deri. Il software nasce dalla necessità di un cambiamento strutturale al precedente prodotto ntop, sviluppato dallo stesso autore, per adattarlo alle nuove esigenze di monitoraggio.

Buona parte del codice di ntopng è disponibile online sotto licenza GPLv3. Funzionalità aggiuntive come la possibilità di generare report avanzati in formato HTML e PDF, di filtrare il traffico indesiderato con regole ad-hoc, di salvare in maniera efficiente su disco statistiche di utilizzo della rete, richiedono invece l'acquisto di una licenza e sono soggette a copyright.

Il codice sviluppato durante questo tirocinio è interamente libero e soggetto alla licenze libere dei rispettivi progetti.

2.4 Software Utilizzato

Oltre ai già citati software ntopng e nDPI, sono stati utilizzati altri software a supporto dell'attività di cattura ed analisi dei pacchetti.

- tcpdump [39] è stato utilizzato per la cattura del traffico di rete.

- Il software Wireshark [33] è stato usato per la cattura, l’analisi e la visualizzazione dei singoli pacchetti di rete e delle metriche associate. Wireshark è stato di particolare aiuto soprattutto per l’implementazione ed il debug dei dissector di rete sviluppati o estesi durante il tirocinio.
- La libreria python pandas è stata utilizzata in combinazione a gnuplot per visualizzare grafici di throughput e altre metriche relative ai flussi analizzati nella fase iniziale del tirocinio.

Infine, il motore di ricerca Google Scholar e l’enciclopedia libera Wikipedia sono stati fondamentali per la ricerca di informazioni, documenti, e riferimenti qui riportati.

2.5 Normative sul Monitoraggio

Gli strumenti di monitoraggio delle reti rappresentano di fatto strumenti d’intercettazione. Prima di utilizzare questi strumenti all’interno di una rete, bisogna avere un permesso esplicito da parte del suo gestore, in cui venga indicato chiaramente il fine di tale attività.

Dal punto di vista legale, l’Articolo 617 Quater [64] definisce il reato d’intercettazione fraudolenta che prevede, a seguito di querela dalla parte offesa, da sei mesi a quattro anni di reclusione. In caso la parte offesa sia un organo statale, si prosegue invece d’ufficio e la pena va da uno a cinque anni.

Ancora, l’articolo Articolo 617 Quinquies [63] regola l’attività d’installazione di strumenti per l’intercettazione, come ad esempio quelli di monitoraggio delle reti, e prevede, in caso di azione fraudolenta, da uno a quattro anni di reclusione.

Capitolo 3 Casi di Studio

In questo capitolo verranno introdotti i contesti che sono stati analizzati nel dettaglio per operare la caratterizzazione del traffico.

3.1 SSL

Il protocollo crittografico SSL, conosciuto come TLS nelle sue nuove versioni, fornisce un servizio di trasporto in sicurezza dei dati applicativi. SSL può essere usato per uno o più dei seguenti scopi:

- cifratura di messaggio
- autenticazione di un messaggio volta a garantirne la provenienza
- autenticazione di un messaggio volta ad assicurarne l'integrità
- autenticazione di un host

L'autenticazione avviene tramite un opportuno MAC, Message Authentication Code, realizzato come firma dell'hash del messaggio.

Il protocollo trova oggi applicazione in moltissimi contesti, come la navigazione web, la posta elettronica, la messaggistica istantanea e le comunicazioni VoIP. In ambito web, recenti iniziative quali HTTPS Everywhere [53] hanno sensibilizzato l'opinione pubblica al problema della privacy e della sicurezza della navigazione, promuovendo l'impiego di tecnologie volte all'utilizzo estensivo della crittografia end-to-end per le comunicazioni.

Una connessione SSL si articola in due fasi:

1. handshake SSL
2. comunicazione cifrata

Durante l'handshake SSL, il server presenta il proprio certificato e, tramite un meccanismo a chiave asimmetrica, viene generata la chiave simmetrica da usare per la comunicazione. Una seconda modalità prevede che anche il client presenti il proprio certificato, permettendo ad entrambi di accertare l'identità dell'altra parte.

L'utilizzo della cifratura simmetrica ha come scopo immediato quello di velocizzare la comunicazione, richiedendo meno risorse di elaborazione. Inoltre, in base al protocollo utilizzato per la generazione della chiave simmetrica, può essere o meno garantita una proprietà molto importante chiamata forward secrecy [57]. La forward secrecy garantisce, nel caso in cui la chiave asimmetrica utilizzata per iniziare tutte le comunicazioni arrivi in possesso di un attaccante, che non sia comunque possibile decifrare le comunicazioni precedenti. Alcuni tra i servizi che utilizzano oggi algoritmi con supporto alla forward secrecy sono Gmail, Google Docs, Twitter e WhatsApp.

Esistono diverse versioni del protocollo SSL/TLS. Le versioni precedenti a TLS 1.0 sono da considerare insicure. Le versioni TLS 1.1 e TLS 1.2 più utilizzate oggi, in base all'algoritmo di cifratura impiegato, sono considerate dagli esperti più o meno sicure. La nuova versione TLS 1.3 è ancora solo una bozza [36].

Nel tempo sono stati studiati diversi attacchi, spesso molto articolati, volti a compromettere la sicurezza delle comunicazioni SSL. Questi attacchi utilizzano principalmente tecniche "attive" di monitoraggio e richiedono dunque l'alterazione del flusso di rete. In questo contesto, le CA, Certification Authority, hanno il ruolo d'estrema responsabilità di garanti della sicurezza mondiale. Recenti scandali [58] [56] hanno dimostrato come il connubio tra CA e organizzazioni con scopi illeciti possa vanificare tutti gli sforzi volti a difendere le comunicazioni.

3.1.1 SSL e Deep Packet Introspection

L'adozione di SSL oggi in molti servizi ha limitato di fatto l'efficacia degli strumenti di monitoraggio di rete basati su tecnologie DPI in quanto i dati nei flussi cifrati non sono più accessibili. Le nuove tecniche DPI per il riconoscimento dei servizi su flussi cifrati utilizzano le poche informazioni disponibili in chiaro, come numero di porta, indirizzo del server e i metadati presenti nei certificati scambiati durante l'handshake SSL.

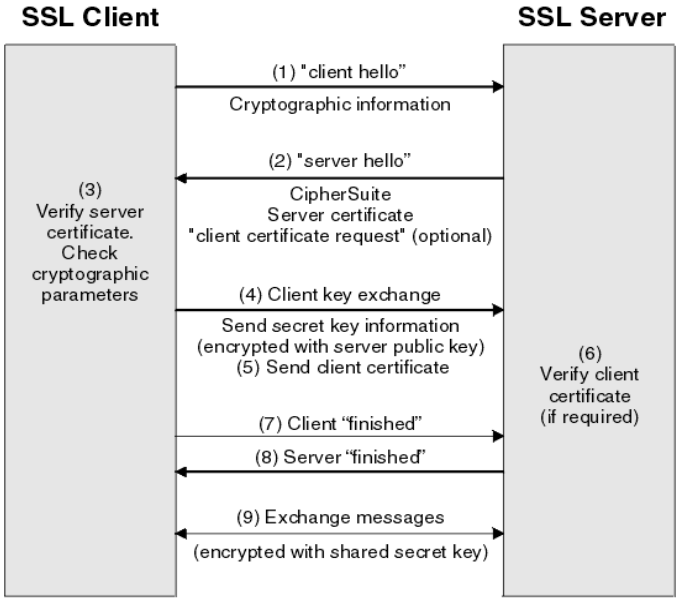

Figura 1: SSL handshake – ibm.com

In figura sono riportate le fasi dell'handshake SSL.

Il dissector SSL presente in nDPI effettua il riconoscimento di SSL basandosi sul primo messaggio di "client hello". Questo messaggio contiene varie sezioni obbligatorie e altre opzionali, indicate col nome di estensioni, insieme alla lista di cifrari disponibili tra cui il server dovrà scegliere. In particolare,

l’estensione SNI, Server Name Indication, nella pratica quasi sempre presente, contiene il nome del server a cui connettersi, un informazione molto importante usata in nDPI per il riconoscimento di servizi noti.

3.2 VPN

Una VPN, Virtual Private Network, è una rete di telecomunicazioni privata instaurata al di sopra di un protocollo di trasmissione pubblico.

Le necessità che hanno dato origine alle attuali tecnologie VPN sono emerse, inizialmente, nel contesto aziendale. L’obiettivo era quello di permettere un accesso remoto alla propria rete privata “intranet” per distaccamenti, partner e impiegati geograficamente dislocati senza compromettere la sicurezza e la confidenzialità delle informazioni.

Il modello tradizionale di rete imponeva il passaggio di collegamenti fisici dedicati tra i singoli punti di smistamento dei dati, con tutti i costi di messa in opera e di manutenzione correlati. Le tecnologie VPN, invece, permettono la creazione di reti logiche, che utilizzano le infrastrutture di rete private e pubbliche esistenti per lo smistamento dei pacchetti di rete.

Successivamente, l’utilizzo di reti VPN si è esteso anche all’ambito privato. Tra le motivazioni rilevanti vi è la necessità di tutelare la propria privacy, a fronte di una maggiore consapevolezza dell’opinione pubblica sull’utilizzo di software spia da parte di alcuni governi ed organizzazioni private. Altre motivazioni riguardano la possibilità di aggirare la censura, limitazioni geografiche o anche di accedere a contenuti nascosti e normalmente non accessibili da internet, il cosiddetto “deep web” [38].

3.2.1 OpenVPN

OpenVPN è un software libero che implementa la tecnologia VPN a livello 2 a livello 3 dello stack di rete. Vi sono due modalità d’instradamento dei pacchetti:

- point-to-point, simula una comunicazione diretta tra due pc
- site-to-site, collega assieme due reti fisicamente distinte

In base alla configurazione del client, inoltre è possibile specificare se instradare tutte o solo alcune delle connessioni in uscita dal dispositivo locale.

Sebbene sia possibile utilizzare il software anche solo nella sua parte libera, la stessa organizzazione che ha ideato OpenVPN fornisce l’Access server, uno strumento proprietario per facilitare la configurazione e la gestione della componente server, insieme a licenze e servizi per aziende e privati.

Il software OpenVPN, a differenza di altre soluzioni VPN, lavora in spazio utente, non richiede dunque un supporto kernel dedicato. Tuttavia il kernel deve supportare le interfacce TUN/TAP, supporto abilitato di default nella maggior parte delle configurazioni. La flessibilità ottenuta lavorando in spazio

utente ha permesso a OpenVPN di ottenere un ampio grado di portabilità e configurabilità e di impiegare tecniche per funzionare anche su dispositivi che impiegano NAT, Network Address Translation.

3.2.2 Configurazione OpenVPN in Modalità TLS

La configurazione automatica della componente server è abbastanza semplice. Lanciato il comando ovpn-init viene avviato un ambiente interattivo per la configurazione. È necessario impostare una password per l'utente openvpn con cui accedere all'interfaccia web, porta 943. Dalla stessa interfaccia è possibile scaricare i file contenente i parametri di configurazione da copiare nel client OpenVPN per la connessione.

La configurazione manuale del servizio server, invece, richiede maggiore attenzione. Per prima cosa è necessaria la creazione di una PKI, Public Key Infrastructure, ovvero un'infrastruttura composta da una CA, Certification Authority, che firmi i certificati client e server che verranno generati. Lo strumento EasyRsa è spesso consigliato per creare velocemente la PKI e le chiavi necessarie ad un uso privato della VPN. A partire da una configurazione di base, è poi necessario creare manualmente il file d'impostazione del server, avendo cura d'impostare lo schema di autenticazione ed il protocollo di trasporto desiderato. Successivamente, bisogna unire i certificati utente, la chiave utente, il certificato della CA e la chiave Diffie Hellman, insieme ad un'appropriata configurazione del client, in un unico file, da copiare poi sul dispositivo client.

3.2.3 OpenVPN Trasporto e Sicurezza

OpenVPN supporta due modalità di autenticazione: a chiave statica PSK o in modalità TLS [45] [46].

- Il primo modo si basa sull'utilizzo di un segreto condiviso, conosciuto quindi da entrambe le parti, composto da 4 chiavi indipendenti per l'autenticazione e la cifratura del traffico in ingresso e in uscita. Questo metodo ha lo svantaggio di richiedere la distribuzione su tutti i client del segreto.
- Il secondo modo si basa invece sull'utilizzo di certificati SSL in ognuna delle due direzioni. Durante la fase di handshake, client e server si scambiano i certificati insieme a dei byte casuali, il Random Material, la cui sorgente software può essere configurata, che verrà usato per la generazione delle chiavi.

In realtà è presente una terza modalità, che utilizza inizialmente la modalità PSK per cifrare ed autenticare l'handshake SSL e, successivamente, instaura una connessione SSL. In questo modo è possibile nascondere completamente i certificati scambiati e scartare a priori i pacchetti non autenticati, possibilmente provenienti da un attaccante che tenti in qualche modo di sfruttare vulnerabilità del servizio. Verrà di seguito analizzata la sola modalità di autenticazione TLS, attiva di default e sicuramente più diffusa.

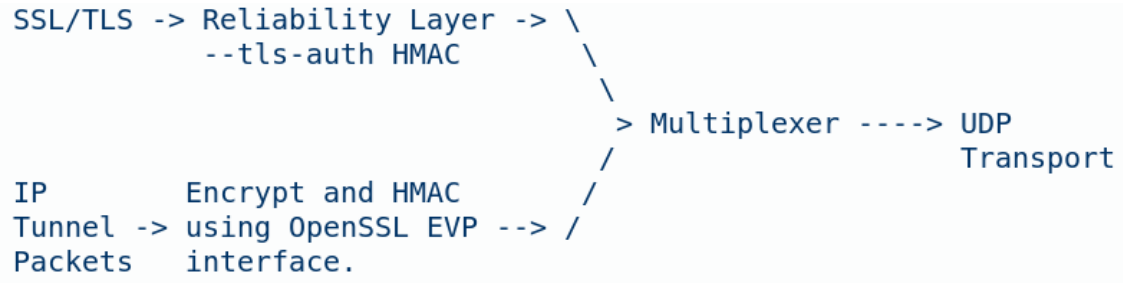

Figura 2: OpenVPN e trasporto – openvpn.net

OpenVPN può usare indistintamente sia UDP sia TCP come protocolli di livello trasporto. All'interno del trasporto, OpenVPN invia alternativamente pacchetti di autenticazione e pacchetti dati, secondo lo schema riportato in Figura 3. Per gestire le rispedizioni di pacchetti di autenticazione è implementato un ulteriore strato di affidabilità. I pacchetti dati vengono invece passati direttamente al protocollo di livello 3.

Sebbene l'utilizzo di TCP come protocollo di trasporto possa sembrare una scelta saggia, è bene ricordare che all'interno della VPN viaggiano a loro volta altri pacchetti TCP e UDP. Scegliendo TCP come protocollo di livello 3 in realtà si creano due diversi strati TCP, uno è quello del trasporto effettivo, l'altro è proprio al traffico inoltrato. Studi hanno messo in evidenza che usare TCP su TCP generi un effettivo decremento delle prestazioni di rete [34]. Il protocollo UDP è quindi da preferire.

3.3 IMAPS

IMAP è il protocollo standard usato per interagire con le caselle di posta elettronica presenti su Internet. IMAP supera le limitazioni di POP3, il suo predecessore, permettendo una gestione completa della posta remota. E' possibile, ad esempio, collegarsi contemporaneamente con più client, effettuare ricerche di messaggi lato server, prelevare solo l'intestazione o specifiche parti dei messaggi così come gestirne gli attributi di stato.

La versione del protocollo maggiormente in uso oggi è IMAPv4, definita nella RFC 3501 [50]. Il dialogo tra server e client avviene per mezzo di comandi e risposte in formato testuale, inviate all'interno del protocollo di trasporto TCP e separate dai terminatori "\r\n". Ogni comando inviato da un client è preceduto da un identificatore alfanumerico chiamato "tag" che deve essere differente per ogni comando inviato.

L'interazione tra client e server, al collegamento, segue normalmente questa sequenza di passi:

1. Il client si collega al server sulla porta standard 143 (IMAP) oppure 993 (IMAPS)
2. Il server invia una risposta OK di servizio pronto
3. Il client invia il comando AUTHENTICATE ed inizia lo scambio di dati per l'autenticazione

4. Il client invia il comando ID ed il server risponde con le proprie generalità
5. Il client invia una serie di comandi LIST ed il server risponde con la struttura della casella di posta
6. Il client invia il comando SELECT per impostare la cartella corrente dentro la casella di posta
7. Il client invia il comando FETCH per ottenere informazioni sui nuovi messaggi

In IMAP la conversazione tra server e client è in chiaro e pertanto insicura. L'utilizzo di SSL per incapsulare i dati applicativi IMAP garantendone la confidenzialità viene indicato col nome IMAPS.

3.3.1 Ricezione Nuove Email

Il protocollo IMAP originale richiede un'azione esplicita del client, un polling periodico, per la ricezione di nuovi messaggi. Il client, dopo aver effettuato le sue operazioni, chiude la connessione.

In una successiva RFC [51], ed in particolare in IMAPv4, è invece disponibile il comando IDLE che, tramite un meccanismo di push su una connessione in attesa, permette la notifica automatica al client di nuovi messaggi. Questo nuovo modello è stato poi esteso da diverse compagnie che hanno proposto il loro protocollo, proprietario o meno, per far fronte a varie necessità, prima tra tutte la limitazione del consumo energetico su dispositivi mobili. Alcuni di queste tecnologie sono state poi implementate all'interno di piattaforme più complesse come ad esempio iCloud di Apple, Direct Push di Microsoft e Google Cloud Messaging.

3.3.2 Opportunistic TLS

Alcuni server che supportano IMAP, nella sua versione non cifrata, mettono a disposizione il comando STARTTLS, definito nella RFC 2595 [6]. Questo comando permette di instaurare una connessione SSL a partire da una connessione IMAP standard. Alla ricezione del comando, se il server supporta tale estensione, viene inviato un riscontro positivo ed inizia la negoziazione SSL, al termine della quale è possibile scambiare dati in modo cifrato.

È bene puntualizzare che questo meccanismo è solo un tentativo opportunistico di instaurare una connessione sicura, in quanto il server, o un attaccante che effettua un attacco di MITM, Man In The Middle, potrebbe comunque declinare la richiesta di connessione sicura, con conseguente utilizzo di caratteri in chiaro. Attacchi di questo tipo vengono definiti TLS downgrade attacks.

Diverse fonti riportano come questi attacchi siano stati effettivamente applicati su larga scala da alcuni ISP [55] [49].

3.4 Web e Traffico Multimediale

Dall'invenzione del WWW, World Wide Web, nel 1989 ad opera dello scienziato Tim Berners-Lee, il web ha subito una grande crescita fino ad affermasi come il principale mezzo di accesso alle informazioni. L'infrastruttura a scambio di ipertesti, che vede in HTML e HTTP i maggiori protocolli, si è evoluta negli anni, abbracciando nuovi linguaggi quali Javascript e CSS ma non solo. L'utilizzo del web come piattaforma, iniziata con lo sviluppo delle applicazioni web, è ormai un elemento affermato della comunicazione moderna. Questo processo di cambiamento è stato ulteriormente amplificato dalla diffusione degli smartphone e dei social networks.

Le pagine web oggi sono applicazioni dinamiche, spesso complesse, che interagiscono in tempo reale con i client dei dispositivi usati dagli utenti. Questo avviene comunemente con tecnologie Ajax o tramite i web sockets che, al contrario di HTTP, mettono a disposizione una connessione bidirezionale per i dati.

Con l'ampliamento e lo sviluppo delle infrastrutture di rete, tramite il web oggi è anche possibile accedere a contenuti multimediali tramite tecnologie di streaming audio e video. Servizi per l'intrattenimento permettono di vedere film on demand e di ascoltare musica a tema. Il nuovo protocollo WebRTC estende anche al web la possibilità di effettuare videoconferenze mentre l'utilizzo di tecnologie innovative come asm.js permette di sviluppare videogiochi da eseguire all'interno del web browser, ponendo così ottime basi per il supporto multi OS.

3.5 Social Networks

I social networks sono oggi la piattaforma principale per lo scambio di informazioni multimediali. L'ostacolo che si pone al corretto riconoscimento di questo traffico è dovuto all'integrazione, all'interno dei siti web, di API per l'accesso alle funzionalità social.

Condividere un articolo presente in rete, effettuare il login tramite un account Facebook, mettere un "mi piace" su un post, genera, a livello di rete, dei flussi con certificato SSL collegato al social networks in questione. Questo certificato, grazie all'utilizzo di tecniche di DPI, permette di associare il flusso al un servizio specifico.

Nella caratterizzazione del traffico di rete, il problema che si presenta è quello di differenziare il traffico generato sul portale principale da quello dovuto all'integrazione delle funzionalità citate in altri siti web.

Capitolo 4 Letteratura

Sono disponibili in rete molti documenti scientifici inerenti all'argomento del riconoscimento del traffico di rete. Le tecniche prese in considerazione verranno qui analizzate e raggruppate in base alla metodologia d'analisi prevalentemente usata, con eventuale riferimento al loro effettivo utilizzo all'interno del lavoro di tirocinio.

Nonostante i vari documenti scientifici, non è stato trovato alcuno strumento opensource di monitoraggio di rete che impieghi con successo tecniche diverse da quelle tradizionali. Software proprietari come Cisco NBAR/NBAR2 [10] e Ipoque PACE [15], documentano l'utilizzo di tecniche di riconoscimento più intelligenti e avanzate.

Un concetto ricorrente è quello delle metriche di rete. Per metrica di rete s'intende una misura inerente al traffico di rete, ricavata direttamente dalle informazioni dei pacchetti oppure elaborata in qualche modo a partire da essi.

4.1 Fingerprint del Protocollo

Le tecniche basate su fingerprint, o impronta digitale, cercano d'identificare, per ogni protocollo, quali siano i valori delle metriche di rete che lo caratterizzano, una specie di "firma" del protocollo. Un strumento opensource chiamato fl0p è l'esempio di un analizzatore passivo di traffico di rete che utilizza questa tecnologia [25].

fl0p considera come metriche rilevanti:

- i cambiamenti di ruolo nell'invio di pacchetti
- la dimensione relativa del payload
- lo IAT, inter packet arrival time, ovvero l'intervallo di arrivo tra i pacchetti

Le metriche che utilizza questo strumento sono molto interessanti ai fini del riconoscimento del traffico. Il concetto di cambiamento di ruolo è riferito al numero di pacchetti consecutivi inviati da una stessa entità client o server. Nei protocolli reali i ruoli sono più o meno fissi. In molti protocolli che per la richiesta di contenuti remoti, ad esempio, la comunicazione prevede l'invio di una richiesta e la successiva risposta con il contenuto in oggetto. Normalmente il ruolo prevalente, in questo caso, è quello del server, che si troverà a spedire un numero di pacchetti di dimensione rilevante in risposta alla richiesta del client.

L'autore di fl0p mette anche in evidenza come eccessive differenze tra le metriche rilevate e quelle attese possano essere un campanello d'allarme per il rilevamento di attacchi informatici. Un timeout eccessivo su una connessione TCP in fase di handshake, ad esempio, è talvolta sintomo di un port scan, così come un ritardo nell'invio di pacchetti SMTP può rivelare l'immissione manuale di comandi da parte di una persona umana.

fl0p fa affidamento ad un database delle firme, implementato come un file di testo. Questo database, tuttavia, deve essere compilato manualmente protocollo per protocollo dopo aver individuato le caratteristiche distintive dello stesso. La richiesta di un intervento manuale ad-hoc per ogni protocollo è la principale limitazione del metodo a fingerprint, soprattutto in vista della grande varietà di configurazioni possibili per i servizi forniti nel mondo reale.

Altri utilizzi delle tecniche di fingerprinting sono discusse in Moore et al. [3] e Sen et al. [28]. Software che utilizzano tecniche simili sono l7-filter [65], tramite le espressioni regolari sul payload, e libprotoident [30], che effettua il fingerprint sui primi 4 bytes dei pacchetti.

4.2 Apprendimento Automatico

Oggi esistono tecniche di ML, apprendimento automatico o Machine Learning, per riuscire a ricavare in modo automatico i corretti valori delle metriche di rete associate ai vari protocolli. Il documento scientifico di Barnaille et al. [18], descrive un metodo per la classificazione del traffico di rete che usa tecniche di clustering automatico quali K-Means e metodi bayesiani.

La soluzione proposta prende in considerazione i primi 4 pacchetti di ogni flusso. Le metriche considerate sono:

- la dimensione dei pacchetti
- lo IAT
- il jitter
- la direzione dei pacchetti

Gli algoritmi basati su ML richiedono dei dati da utilizzare durante la fase di training per il corretto dimensionamento dei parametri. Gli autori riportano di aver analizzato manualmente tre traccie dati provenienti dalla rete Paris 6, indicate come campione significativo di dati. Le euristiche sui cluster sono ideate in modo tale da privilegiare i pattern dominanti, corrispondenti ai protocolli applicativi reali. Dai risultati riportati nel documento scientifico, l’accuratezza su traffico sconosciuto sempre su rete Paris 6 non è molto alta, con valori inferiori al 70% anche per protocolli molto utilizzati come SMTP e HTTP, e inferiori al 80% per HTTPS.

Altre tecniche similari sono descritte in Zander et al. [27], Erman et al. [16] e McGregor et al. [4]. Livadas et al. [7] e Gu et al. [14] utilizzano ML per il rilevamento di botnet nel traffico di rete, anche cifrato. Alshammari e Zincir-Heywood [26] mostrano come applicare efficacemente ML nel contesto di riconoscimento di traffico SSH e Skype, ottenendo un’accuratezza spesso superiore al 95%.

4.3 Analisi Statistica

Numerose sono le proposte d'impiego di metodi statistici per il riconoscimento del traffico di rete. Korczyński et al. [22] propone ad esempio un metodo basato sulle catene di Markov. Gli autori analizzano i vari messaggi descritti dallo standard SSL e li confrontano con le implementazioni reali. Secondo il loro studio, il riconoscimento del traffico di rete può essere effettuato proprio grazie alle piccole differenze implementative dei protocolli che ogni servizio ha in uso. I risultati ottenuti sono particolarmente promettenti soprattutto per quanto riguarda il traffico cifrato.

Crotti et al. [23] propone invece l'utilizzo di tecniche statistiche avanzate, basate sull'estrapolazione di funzioni di densità, punteggi di anomalia e filtri gaussiani. Il metodo proposto, come indicato, richiede però una pre-classificazione iniziale, operata da strumenti esterni.

Dainotti et al. [1] propone invece l'utilizzo di modelli di Markov nascosti. Basandosi sulla dimensione dei pacchetti e sullo IAT, il metodo è implementato con la costruzione di classificatori statistici, ognuno relativo ad un protocollo da analizzare. Questi classificatori ricevono in ingresso i pacchetti catturati e danno una stima della probabilità che il flusso appartenga al proprio protocollo. L'analisi ed i test effettuati si limitano però a considerare solo protocolli non cifrati e questa risulta un'ovvia limitazione.

Un interessante progetto chiamato Joy [11], sviluppato all'interno del gruppo di ricerca di Cisco, si avvicina molto, nello spirito e nell'architettura, a nDPI. Il progetto si presenta come un framework generico per l'estrapolazione di dati dai flussi e la loro codifica in formato JSON. L'utilizzo appropriato di questi dati è lasciato all'applicazione specifica. Joy analizza diverse metriche di rete, come lo IAT, la dimensione dei pacchetti ed i metadati dei flussi. Inoltre, calcola la distribuzione empirica e l'entropia del payload, delle metriche derivate che potrebbero portare ad un riconoscimento più accurato.

Le tecniche statistiche discusse, con opportune variazioni, sono descritte in altri numerosi articoli scientifici, come Wright et al. [8], Zuev et al. [2], Paxson et al. [31], Soule et al. [5] e Campos et al. [12].

L'utilizzo di queste tecniche va ben ponderato perché richiede solidi fondamenti teorici e una conoscenza dettagliata dei contesti pratici, senza i quali i risultati ottenuti rischiano di essere forvianti o non trovare un'applicazione concreta.

4.4 Traffico SSL e Web

Per l'identificazione del protocollo o del servizio associato ad un flusso SSL, nDPI fa uso di due informazioni: la SNI, Server Name Indication, e l'indirizzo IP del server. La SNI è estraibile dal primo pacchetto inviato da un client durante l'handshake SSL. nDPI cerca di effettuare il match tra le sottostringhe salvate nel suo database e la stringa SNI estratta. In alternativa, effettua una ricerca dell'indirizzo IP del server all'interno dei range di indirizzi nel database locale.

La corrispondenza tramite indirizzo IP sembra, a prima vista, un buon modo di riconoscimento. Nella pratica, tuttavia, è richiesto un intervento manuale per l'estrapolazione del giusto range di indirizzi, range che non è fisso ma varia continuamente nel tempo. Negli ultimi anni si è molto diffuso l'utilizzo dei CDN, Content Delivery Network, servizi di hosting di contenuti utilizzati da molti siti web per fornire un accesso più veloce ai propri contenuti. La diffusione dei CDN aggiunge un ulteriore difficoltà alla caratterizzazione del traffico.

Alcuni ricercatori italiani del Politecnico di Torino [24] hanno mostrato come, utilizzando tecniche di ML ed opportuni classificatori, sia possibile, in maniera abbastanza efficace, raccogliere gli indirizzi usati nei vari domini in grandi gruppi di indirizzi, propri dei servizi analizzati.

Un altro documento dei ricercatori del Politecnico di Torino [21], più interessante dal punto di vista di questo tirocinio, propone invece una tecnica per riconoscere azioni esplicite di richiesta di una pagina web da parte di un utente umano. Il meccanismo si basa sul tracciamento delle pagine web visitate utilizzando a proprio vantaggio il campo referer contenuto nelle richieste HTTP. Questa strategia fa ancora uso di ML per sopperire alle diverse implementazioni dei web browser, che evidentemente manifestano comportamenti diversi uno dall'altro. Non è chiara l'applicabilità della tecnica proposta nel contesto di flussi HTTPS, dove cioè il campo referer è cifrato. Un'analisi così fortemente legata ad informazioni in chiaro, in vista di una presenza sempre maggiore su internet di traffico SSL, risulta pertanto limitativa.

Capitolo 5 Strategie Proposte

Sono di seguito analizzate le strategie elaborate durante il tirocinio per l'identificazione e la caratterizzazione del traffico di rete.

5.1 Definizione delle Macro-Attività

I flussi di rete sono stati classificati in varie macro-attività. La definizione delle macro-attività d'interesse ha tenuto conto sia della necessità di sintesi, essenziale in un contesto così ampio, sia della rilevanza che certi protocolli hanno rispetto ad altri nel definire nel modo migliore il comportamento di un utente nella rete.

Con queste premesse, il traffico è stati suddiviso in diversi profili, indicati nella tabella che segue.

Attività	Traffico attivo	Rumore
Condivisione file	Traffico P2P associato a grandi quantità di dati	Collegamento ai peer
Controllo remoto	Invio di comandi o ricezione di output da computer remoto	Keepalive periodico
Giochi online	Gioco	Servizi di supporto al gioco
Invio email	Spedizione di email	–
Messaggistica e comunicazione	Messaggi istantanei, chiamate VoIP e videoconferenze	Traffico di controllo
Multimediale	Flussi audio o video, anche in streaming	Traffico di controllo
Navigazione web	Navigazione per mezzo del browser web	Servizi web accessori
Sincronizzazione email	Ricezione email e sincronizzazione	Sincronizzazione periodica
Social Networks	Utilizzo del portale	Integrazione in siti web
Trasferimento file	Scaricamento di file via web o sincronizzazione su dispositivi remoti	Traffico di controllo
VPN	Traffico offuscato sotto rete VPN	Keepalive periodico
Altro	Traffico rilevante	Traffico non rilevante

Il traffico di ogni attività viene diviso in traffico attivo e rumore di fondo. Per traffico attivo s'intende il traffico che può essere considerato diretta conseguenza di un'azione umana. Per rumore di fondo s'intende invece il traffico generato dalle applicazioni per operare sincronizzazioni o comunque per il

funzionamento interno dell'applicazione. Il traffico attivo rappresenta quindi il miglior indicatore per stabilire il comportamento effettivo di un host. Il profilo "Altro" indica tutto quel traffico non altrimenti classificato.

Va puntualizzato che la tabella mostra la classificazione ideale del traffico così come è stata ideata. Nella pratica, la complessità dei casi reali ha spesso impedito una perfetta corrispondenza.

5.2 Rilevamento OpenVPN

nDPI fornisce un dissector in grado di riconoscere OpenVPN nella sua configurazione di default, ovvero modalità TLS base con trasporto UDP porta 1194 o con TCP porta 443. La porta 443, in particolare, è anche la porta standard assegnata ad HTTPS. Questo significa che è necessario distinguere in qualche modo il traffico HTTPS da quello OpenVPN su TCP.

La prima idea per operare questa distinzione è stata quella di basarsi sulle metriche del flusso in relazione a quelle che sono le caratteristiche del traffico HTTPS. Come metrica rilevante è stata inizialmente scelta la direzione dei pacchetti di rete nel tempo. Da un normale flusso HTTPS, infatti, ci si aspetta un andamento che predilige la direzione verso il client, a parte i pacchetti della fase di handshake SSL. Il client normalmente effettua una richiesta HTTP, racchiusa in pochi pacchetti TCP, a cui segue una risposta del server ben più grande.

Dai test effettuati, tuttavia, non è risultato evidente il comportamento atteso. In effetti il protocollo HTTP viene usato come trasporto da una moltitudine di protocolli applicativi. API che utilizzano REST o SOAP su HTTP, così come protocolli proprietari, utilizzano spesso SSL sulla porta 443 per la fruizione di servizi web, mischiando dunque i ruoli di server e client. Inoltre, lo standard HTTP 1.1, ampiamente adottato, prevede l'invio di richieste multiple su una stessa connessione TCP, rendendo vane le considerazioni fatte.

In una seconda analisi si è invece riflettuto su quali potessero essere gli indicatori di una VPN. Se, ad esempio, l'unico flusso attivo di un host locale è un flusso SSL con una durata rilevante, ci sono delle possibilità che si tratti di una VPN che instrada tutto il traffico dell'host oppure di un flusso per il controllo remoto. In altri casi, oltre che al solo flusso SSL, potrebbero esserci anche dei flussi DNS poiché spesso il client OpenVPN non gestisce opportunamente l'aggiornamento dei server DNS, le cui richieste vengono invece inviate in modo non cifrato. O ancora, l'utilizzo di particolari configurazioni evidenziate dallo scambio dei certificati potrebbe indurre qualche sospetto [22]. Fare un'analisi basata su queste caratteristiche, tuttavia, rischia di essere imprecisa e soggetta ad errori.

La scelta finale per il riconoscimento di traffico OpenVPN è infine ricaduta su un metodo tradizionale. Facendo uso della tecnologia DPI, sono state estratte dai flussi le informazioni che caratterizzano lo strato di trasporto proprio dell'handshake OpenVPN.

L'implementazione di questo metodo è discussa più avanti nel Capitolo 6.3. È stata quindi momentaneamente esclusa la possibilità di una caratterizzazione più generale del traffico VPN.

5.3 Caratterizzazione IMAPS

Il traffico attivo per il protocollo IMAP/S è rappresentato da tutto quel traffico che interessa lo scaricamento di contenuto dal server di posta elettronica.

Questo tipo d'interazione può essere individuata nell'invio, da parte del client, di un comando di FETCH, seguito da una risposta del server che contiene i dati richiesti. Quasi tutti gli altri comandi disponibili in IMAP sono invece di sincronizzazione o servono a gestire metadati. Questi devono essere classificati come rumore di fondo.

Una caratterizzazione basata sui comandi può funzionare per IMAP ma non per IMAPS, essendo quest'ultimo un protocollo cifrato. Si è pensato dunque a come poter estendere il concetto a IMAPS.

Una caratteristica del protocollo SSL è che la lunghezza dei dati non viene mascherata durante la fase di cifratura. Questa peculiarità, se unita alla conoscenza del funzionamento del protocollo applicativo, può essere utilizzata per estrarre varie informazioni dal flusso cifrato. Wright et al. [9] ha mostrato ad esempio come è possibile riconoscere, entro un certo margine, la lingua parlata all'interno di una connessione VoIP cifrata che usa l'encoder Speex. Il segnale, dopo la cifratura, ha una grande similarità con quello originale.

Per operare la classificazione del traffico IMAPS si è cercato di riconoscere, nelle sequenze di pacchetti cifrati IMAPS, l'inizio e la fine di un comando IMAP. Una volta stabiliti i limiti temporali di ogni comando, la caratterizzazione del traffico si è basata su alcune metriche:

- bytes totali del comando di richiesta
- bytes totali della risposta
- numero di pacchetti della risposta
- numero totale di comandi nel flusso
- il fatto che il server abbia o meno inviato un pacchetto vuoto in risposta al primo pacchetto inviato dal client (ACK* in Figura 3)

La presenza di un pacchetto di solo ACK indica che il server non ha potuto soddisfare immediatamente la richiesta del client. Dai test effettuati, infatti, è emerso che spesso i server rispondono immediatamente ad alcuni comandi IMAP, come quelli di autenticazione, ed inviano invece solo un riscontro in risposta ai comandi più complessi come FETCH e SEARCH. Questa caratteristica torna utile al fine della caratterizzazione.

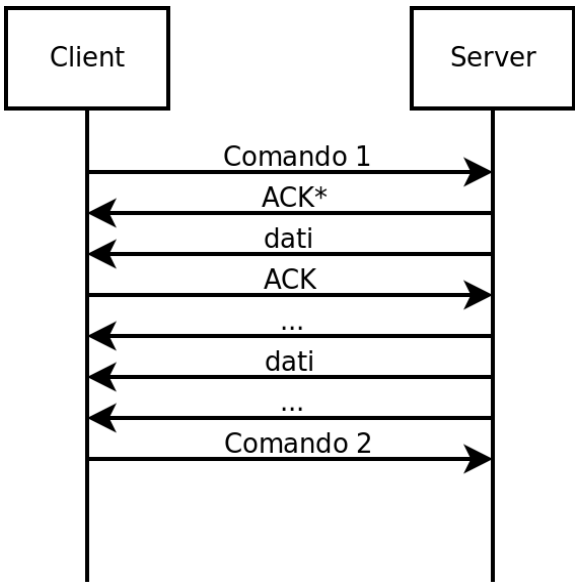

Figura 3

La Figura 3 mostra un'interazione ideale tra client e server, presa come base per l'ideazione della tecnica. Una sequenza di pacchetti di un comando IMAP/S ha inizio con un pacchetto non vuoto del client, dove con "non vuoto" s'intende un pacchetto con dimensione payload TCP superiore a 0 bytes. La sequenza continua fintanto che il server invia pacchetti non vuoti ed il client al più risponde solo con pacchetti vuoti. Non appena il client invia un pacchetto non vuoto la sequenza precedente si considera terminata ed il comando compiuto. La sequenza viene interrotta anche nel caso in cui passi un certo tempo dall'ultimo pacchetto non vuoto ricevuto. Questo aiuta, ad esempio, a evitare che comandi di push su connessione IDLE vengano considerati parte della sequenza precedente.

Il tutto si basa sull'assunzione che il client invii un singolo comando IMAP e attenda la risposta del server senza inviare altri comandi in parallelo. L'invio di comandi in parallelo, sebbene espressa nella RFC 3501 [50], ha un'applicazione limitata a causa della complessità degli algoritmi necessari a garantire la non ambiguità della sequenza di comandi.

5.4 Rilevamento di Traffico Web

Il rilevamento di traffico web all'interno di flussi SSL o HTTPS parte dalla constatazione pratica che, in molti dei flussi usati per trasportare ipertesti e contenuti web, possono essere individuate alcune caratteristiche distintive. L'analisi che segue tiene in considerazione solo i pacchetti di payload di eventuali flussi SSL.

All'inizio di una comunicazione, un client HTTP invia una richiesta al server e questo risponde con la pagina web corrispondente all'indirizzo specificato. Considerando i primi pacchetti scambiati, sono state elaborate le seguenti assunzioni:

- l’intervallo massimo tra i pacchetti è limitato
- vi deve essere un numero minimo di bytes scambiati
- il numero di bytes inviati dal client è inferiore a quelli inviati dal server

Verificando queste assunzioni sui primi pacchetti di flussi SSL, si possono identificare almeno i flussi che mostrano un comportamento assimilabile ai flussi di navigazione web.

5.5 Caratterizzazione del Traffico di Social Networks

nDPI, tramite il certificato SSL, è in grado di riconoscere efficacemente flussi inerenti ai social networks come Facebook e Twitter. Il problema, come già accennato, è quello di filtrare in modo adeguato il traffico dovuto all’integrazione di questi servizi all’interno di altri siti web. Vengono di seguito analizzate due situazioni reali d’integrazione.

Il sito www.informateci.org contiene al suo interno un form per il login attraverso l’account di Facebook. Non appena viene visitata la pagina web, è possibile individuare diversi flussi HTTP con URL “graph.facebook.com”. Gli indirizzi contengono le miniature degli utenti visualizzati nella pagina che probabilmente hanno effettuato il login tramite account Facebook. Vi è anche un flusso SSL di tipo “scontent”, la cui durata però è inferiore al secondo. In caso si abbia una sessione attiva su Facebook, i flussi in questione sono due e la durata è di qualche secondo.

Nel sito www.cinemamultimedia.it sono invece presenti ben quattro flussi SSL con certificato Facebook e due flussi HTTP, i cui URL fanno riferimento ad una versione della SDK di Facebook. Il volume di traffico, in questo caso, è abbastanza elevato e la discriminazione tra traffico attivo e rumore non sembra ovvia.

Una prima analisi ha tenuto in considerazione le seguenti metriche di rete:

- bytes scambiati
- IAT
- direzione dei pacchetti
- durata complessiva

Quest’analisi, tuttavia, non ha portato a risultati positivi. In particolare, i flussi Facebook generati dall’integrazione presentano, in alcuni casi, un alto volume di traffico, con tempi e direzioni compatibili con quelle del portale principale.

La tecnica che viene proposta ed implementata fa uso, invece, di informazioni provenienti da più flussi. Secondo questa strategia, quanti più flussi inerenti al protocollo in esame generano traffico in un certo intervallo di tempo, tanto è più probabile che quel traffico avvenga effettivamente all’interno del

portale del social network. La visualizzazione di foto, video e altri contenuti multimediali tipici dell'attività di navigazione su social networks, contribuiscono ad aumentare il volume del traffico nel dato istante.

Le metriche ricavate presuppongono un'aggregazione per host, sulla base del server remoto. I flussi, possibili concorrenti per la rilevazione dell'attività, devono inoltre essere mantenuti in qualche modo in memoria in una sorta di finestra temporale. Come verrà spiegato nel capitolo relativo all'implementazione, se si vuole evitare l'allocazione dinamica di memoria, ed incrementare così le performance, è necessario stabilire anche la politica d'inserimento ed eliminazione dei flussi nella finestra. Questi temi sono analizzati in dettaglio nel paragrafo 6.5.

5.6 Caratterizzazione basata sui Bytes

Una tecnica di caratterizzazione generica applicata in vari contesti per operare la distinzione tra traffico attivo e rumore di fondo fa uso degli algoritmi di SMA e WMA [47], opportunamente modificati per adattarsi alle esigenze specifiche. Gli algoritmi usano delle soglie parametriche, configurabili per ogni protocollo, che permettono in alcuni casi di stabilire la corretta natura del traffico.

SMA è stato modificato per tenere in considerazione la distanza tra i pacchetti. WMA è abbastanza diverso dall'implementazione originale in quanto utilizza proprio lo IAT come fattore di scala dei valori. La loro implementazione verrà discussa nel capitolo successivo.

Capitolo 6 Implementazione

6.1 Framework delle Attività

Nella fase iniziale del tirocinio, era stato proposto un prototipo di framework che usava classi specializzate, ognuna per il riconoscimento di una macro-attività. Questo framework, tuttavia, aveva i seguenti problemi:

- duplicazione dei dati contenuti nei flussi, con conseguente aumento di memoria
- inefficienza, a causa dell'allocazione dinamica di oggetti per ogni flusso
- mancanza di flessibilità di configurazione
- formato binario non standard per il salvataggio delle attività che, seppure efficiente, richiede codice specifico per la scrittura e lettura delle informazioni

Per questi motivi il prototipo è stato accantonato in favore di un framework maggiormente integrato con il codice di ntopng.

Nell'ideazione del framework si è deciso di separare la logica di caratterizzazione dei flussi, implementata in linguaggio C, dalla configurazione specifica di ogni protocollo. Questa seconda parte è stata infatti implementa in linguaggio Lua, il cui supporto era già presente all'interno di ntopng. Grazie a questa separazione, è adesso possibile modificare i protocolli associati ai profili delle attività senza dover ricompilare il codice, o addirittura senza terminare il processo di ntopng per mezzo del segnale SIGHUP.

I due concetti fondamentali nel framework sono i profili attività ed i filtri attività.

- I profili corrispondono alla definizione data nel paragrafo 5.1. Sono quindi le categorie in cui il traffico è catalogato.
- I filtri, invece, hanno il compito di operare la caratterizzazione del traffico, ovvero stabilire se questo è da considerare traffico attivo oppure rumore di fondo.

I filtri ricevono in ingresso, per ogni pacchetto di rete, le seguenti informazioni:

- dimensione del pacchetto in bytes
- direzione del pacchetto, dal client al server o viceversa
- tempo di ricezione del pacchetto
- flusso di cui il pacchetto fa parte

Grazie a questi dati, i filtri ricavano le metriche di rete, dirette o derivate, che useranno per adempiere al loro compito. Tutti i filtri espongono un'interfaccia comune e, fatta eccezione di alcuni casi particolari, sono del tutto generici ed indipendenti dal profilo assegnato al flusso. Le regole su come associare profili e filtri ai flussi di rete sono configurabili lato Lua. Questo facilita notevolmente l'attività di tuning dei parametri dei vari filtri.

La caratterizzazione del traffico di rete avviene quindi con il seguente procedimento:

1. I pacchetti di rete vengono catturati dall'interfaccia di rete
2. ntopng aggrega i pacchetti in flussi
3. nDPI riconosce il protocollo o il servizio sottostante
4. Il flusso viene marcato con un profilo e gli viene assegnato un filtro
5. Per ogni nuovo pacchetto, il filtro si occupa di discriminare il traffico attivo dal rumore
6. Contatori delle attività vengono continuamente aggiornati e, periodicamente, scritti su disco

I punti da 4 a 6 sono quelli implementati nel corso del tirocinio.

6.1.1 Implementazione Lua

Lua è un linguaggio di scripting molto leggero, utilizzato spesso in progetti complessi per fornire, in modo veloce, accesso alle funzionalità interne del software principale. La fase di progettazione della sua integrazione ha richiesto particolare attenzione per garantire un corretto isolamento tra le strutture interne di ntopng e quelle Lua.

Per effettuare l'integrazione è necessario per prima cosa creare un ambiente in cui l'interprete Lua possa mantenere il proprio stato, la Lua VM. L'ambiente va poi popolato con le variabili che definiscono l'interfaccia tramite la quale è possibile invocare funzioni C direttamente dallo script Lua. L'invocazione dello script avviene invece da C chiamando opportune funzioni per l'inserimento nello stack Lua dei parametri da passare alla funzione da invocare.

Sono state definite le seguenti callback Lua:

- flowCreate – invocata alla creazione di un nuovo flusso
- flowDelete – invocata alla distruzione di un flusso
- flowUpdate – invocata ad intervalli regolari, tipicamente di 5 secondi, quando in ntopng avviene l'aggiornamento di statistiche interne
- flowProtocolDetected – invocata non appena viene riconosciuto il protocollo del flusso o quando nuove informazioni possono essere usate per delineare meglio il carattere del flusso

Le prime tre sono destinate ad un uso futuro e non verranno ulteriormente discusse.

La callback flowProtocolDetected contiene il codice che si occupa dell'assegnamento del profilo, del filtro e relativi parametri per la maggior parte dei protocolli. Viene eseguita all'interno della thread principale di ntopng, che contiene tutte le principali strutture dati ed è imputato alla cattura dei pacchetti. Questo permette di evitare i problemi di sincronizzazione ma pone anche il rischio di una possibile perdita di pacchetti. Le operazioni effettuate nella flowProtocolDetected sono comunque molto leggere e non hanno presentato problemi del genere.

È interessante notare come all'interno dell'ambiente Lua non venga mantenuto alcuno stato tra le diverse invocazioni. Qualunque informazione sul flusso, sul suo profilo attuale così come l'impostazione dei vari parametri avviene tramite la API esposta a Lua da C. Questo ha il doppio vantaggio di evitare il passaggio di proprietà di oggetti tra i due ambienti, sintomo di una cattiva architettura, e di permettere di ricaricare al volo l'ambiente Lua.

6.1.2 Implementazione dei Filtri

I filtri del framework hanno a disposizione due strutture dati su cui lavorare:

- activity_filter_config serve a contenere i parametri che sono stati passati al filtro da Lua. Non può essere modificata dal filtro
- activity_filter_status mantiene invece tutti i dati sullo stato attuale del filtro, necessari al suo funzionamento

Queste strutture sono implementate come union C e vengono allocate staticamente. Questa implementazione ha il vantaggio di ridurre al minimo l'overhead in termini di memoria e tempo di esecuzione.

Si presenta qui una panoramica dei filtri implementati con riferimento alla metrica calcolata da ognuno.

- SMA con legame temporale – media mobile con inserimento di valori nulli negli intervalli temporali, discussa più avanti
- WMA con legame temporale – media pesata sulla base dell'intervallo temporale
- Sequenza di comandi – numero di interazioni effettuate lato server in risposta a quelle del client
- Web – filtro specifico per il riconoscimento di attività web
- Rapporto – rapporto tra i bytes inviati e ricevuti o viceversa
- Inter-flusso – filtro specifico per il riconoscimento di attività di social networks

6.2 Distinguere Handshake SSL dai Dati

In caso di flussi SSL è utile considerare distintamente i pacchetti che contengono l'handshake SSL da quelli che contengono dati. Molte delle tecniche elaborate durante il tirocinio si basano sull'assunzione che i pacchetti considerati contengano il payload effettivo dei protocolli. Questo permette di sapere, ad esempio, chi tra client e server abbia inviato il primo pacchetto applicativo.

nDPI termina il suo lavoro al riconoscimento del pacchetto di "client hello", mentre tutti i successivi pacchetti sono, normalmente processati da ntopng come dati. Vista l'importanza di effettuare la distinzione citata, all'interno del lavoro di tirocinio è stato implementata una funzione per il riconoscimento dello stato di progresso attuale dell'handshake.

All'interno del flusso SSL viene salvato in modo distinto il progresso del client e quello del server. Gli stati di questo progresso sono indicati come "unknown", "hello", "ccs".

- unknown è lo stato in cui non è stato ancora identificato alcun pacchetto dell'handshake
- hello indica che l'entità, client o server, ha inviato un messaggio di "hello"
- ccs indica che l'entità ha inviato il messaggio di Change Cypher Spec

Dopo il Change Cypher Spec, l'entità ha terminato l'handshake e può iniziare la sua parte della comunicazione cifrata. Un messaggio cifrato corrisponde a dati applicativi reali. L'algoritmo implementato permette l'avanzamento dei singoli progressi purché questo rispetti le regole del protocollo SSL. Ad esempio, lo stato di "hello" per il server deve essere necessariamente preceduto da uno di "hello" del client.

La scelta di trattare indipendentemente il progresso client è server è stata dettata dalla constatazione della possibilità d'invio di pacchetti dati quando ancora l'handshake è in corso. Questo può accadere quando il client ha già inviato il CCS e necessita di spedire immediatamente dati, mentre ancora il server non ha inviato il CCS.

Sebbene sia possibile adesso identificare correttamente i pacchetti di handshake, la differenziazione del traffico SSL è, tuttavia, ancora incompleta. SSL prevede infatti la possibilità di inviare messaggi SSL anche durante la normale conversazione, per esempio i messaggi di Alert [37]. L'unico modo per discriminare correttamente tutti questi messaggi è quello di effettuare la ricostruzione completa del flusso SSL estraendo la lunghezza del record successivo, indicata nelle intestazioni dei record di tipo DATA. Fare questo in modo appropriato richiede a sua volta la ricostruzione del flusso TCP basata sui numeri di sequenza. Tutto questo lavoro, in un prodotto per il monitoraggio delle reti, rischia di essere fuori luogo e comprometterne la performance.

Le modifiche apportate sono stata integrate all'interno della classe Flow di ntopng, che adesso dispone di metodi per l'interrogazione sullo stato generale dell'handshake così come sulla natura del pacchetto SSL. L'infrastruttura per il riconoscimento delle attività implementata nel tirocinio può adesso disporre di questi dati per effettuare analisi più accurate.

6.3 Dissector OpenVPN Indipendente dalla Porta

Il dissector originale di OpenVPN in nDPI è stato sostituito con il dissector implementato in questo tirocinio. Grazie al rilevamento del particolare protocollo usato da OpenVPN, è adesso possibile riconoscere flussi OpenVPN in modo indipendente dalla porta usata. Attualmente il dissector supporta la sola modalità TLS.

L'identificazione si basa sul riconoscimento di alcuni byte caratteristici dei messaggi utilizzati durante la fase di handshake di OpenVPN. La struttura di un messaggio di handshake è riportata in figura.

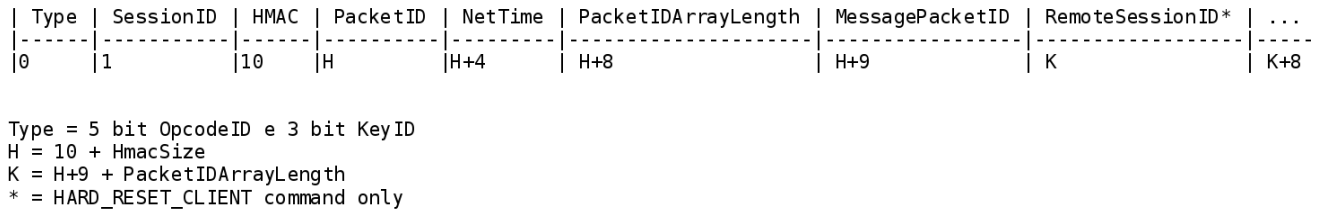

Figura 4: Payload UDP pacchetto di handshake OpenVPN

Il parametro HmacSize è la grandezza in bytes del digest della funzione crittografica usata per l'autenticazione. Nella configurazione di default viene utilizzato SHA1, con un digest di 160 bit, dunque in questo caso HmacSize vale 20. Alcune funzioni di hash popolari che usano un digest della stessa grandezza sono RSA-SHA1 e DSA-SHA1. Funzioni hash che invece usano un digest di 128 bit (HmacSize vale 16) sono invece RSA-MD5, MD5, RSA-MD4, MD4.

L'algoritmo inizia col la ricerca di un pacchetto HARD_RESET_CLIENT e cerca di determinare la grandezza del digest calcolando l'offset per il campo MessagePacketID nei casi di digest a 128 bit e 160 bit. Se viene identificato nel campo un valore pari ad 1, che identifica il primo pacchetto della sequenza di handshake, il pacchetto viene considerato valido e viene copiato in memoria il SessionID.

Il secondo messaggio che viene ricercato è quello con OpcodeID pari a HARD_RESET_SERVER. Anche in questo caso si cerca di determinare la lunghezza del digest allo stesso modo. Stavolta però l'offset interessante da calcolare è quello del campo RemoteSessionID poiché, in una sessione OpenVPN, questo campo deve avere valore uguale al SessionID comunicato dal client. Se questa corrispondenza viene rilevata, si conclude che i pacchetti analizzati fanno parte effettivamente di un handshake OpenVPN.

Il dissector implementato è stato in grado di riconoscere correttamente tutti i flussi OpenVPN con le configurazioni di default più comuni. La probabilità di ottenere un falso positivo è molto bassa in quanto, entro la soglia massima di pacchetti impostata (attualmente 5 pacchetti), per poter avere esito

positivo vi deve essere una corrispondenza con i valori dei campi OpcodeID a 5 bit (32 combinazioni), il PacketID a 8 bit (256 combinazioni) e tra il SessionID ed il RemoteSessionID a 64 bit (19esima potenza di 10).

6.4 Riconoscimento del MIME type HTTP

Il MIME (Multipurpose Internet Mail Extensions) type [48], anche conosciuto come media type, è un identificatore che specifica il tipo di contenuto presente in un qualche oggetto del web. L'identificatore è composto da due parti, separate da un carattere "slash". La prima parte, chiamata tipo, identifica la categoria generica come audio, video o testo. La seconda parte, o sottotipo, contiene invece il formato specifico del contenuto in questione. Una terza parte opzionale, separata da punto e virgola, indica infine la codifica del documento. Nel protocollo HTTP 1.1, la rfc2616 [54] definisce il formato delle intestazioni del protocollo tra cui il Content-Type, che deve contenere un valore di MIME type opportuno in base al formato della richiesta o risposta HTTP.

Dal Content-Type di una risposta HTTP è possibile estrarre il tipo esatto del contenuto inviato. Tra i contenuti maggiormente usati, sono stati considerati indicatori di traffico multimediale quelli con tipo "audio", "video" e "application" con sottotipo "x-shockwave-flash". Per quanto riguarda il traffico web, in presenza del tipo "text" o del tipo "application" con sottotipi "javascript", "x-javascript" e "ecmascript", tutto il traffico viene immediatamente classificato come traffico attivo senza usare il filtro web precedentemente introdotto. Le informazioni per mappare il Content-Type a profili e filtri sono specificate all'interno dello script Lua discusso in precedenza.

6.5 Filtro Inter-flusso

Uno dei filtri per la caratterizzazione del traffico implementati è il filtro inter-flusso. Il suo nome deriva dalla sua peculiarità di utilizzare informazioni presenti in più flussi per determinare se vi sia o meno traffico attivo.

Il filtro necessita di strutture aggiuntive per mantenere lo stato dei flussi d'interesse. Per ogni servizio che utilizza questo filtro, attualmente solo Facebook e Twitter, è stata quindi creata una tabella all'interno della classe Host, con un numero fisso di entrate. La tabella ha per chiave l'indirizzo del flusso e per colonne alcune informazioni di stato.

Quando un nuovo pacchetto viene passato al filtro inter-flusso, si cerca di aggiungerlo all'interno della tabella. Nel caso in cui il flusso associato al pacchetto esista già nella tabella, si procede al solo aggiornamento delle statistiche. In caso contrario, uno nuovo record viene utilizzato.

Ogni record contiene le seguenti informazioni:

- puntatore al flusso, la chiave
- tempo di arrivo del primo pacchetto

- tempo di arrivo dell'ultimo pacchetto
- numero totale di pacchetti

L'algoritmo, in particolare, tiene conto di vari fattori per eliminare dalla tabella eventuali flussi non più utili. Per ogni slot della tabella viene infatti calcolato un valore di "inutilità". All'inserimento nella tabella, il record più inutile viene scelto per essere sovrascritto dal nuovo record. Nel caso in cui la tabella sia piena di flussi, tutti relativamente utili, il nuovo record viene invece scartato.

L'indice di inutilità a cui si è fatto riferimento è rappresentato da un valore float. Nel caso di uno slot vuoto, il valore di inutilità vale 1. Altrimenti, viene espresso dalla formula

$$(tempo_{attuale} - tempo_{ultimopacchetto}) / intervallo_{max} - pacchetti / 100$$

Il parametro d'intervallo massimo è una costante che rappresenta la massima differenza attesa tra un pacchetto del flusso ed il suo successivo, attualmente impostato a 5 secondi. Più il flusso è vecchio, più il primo termine della formula è elevato, ovvero il flusso ha alta inutilità. Il secondo termine tiene invece conto dei pacchetti accumulati per il flusso. Più pacchetti ha il flusso, più il valore tende ad abbassare l'indice di inutilità, ovvero il flusso è utile.

Può quindi accadere che, anche in presenza di slot liberi nella tabella, si preferisca comunque inserire un nuovo record al posto di uno già esistente ma troppo vecchio e poco utile; questo accade quando l'indice l'inutilità è superiore a 1. Quando il valore d'inutilità è inferiore a zero il nuovo flusso viene invece scartato e la tabella rimane immutata.

Il filtro inter-flusso, dopo aver aggiornato le statistiche per host nel modo appena descritto, decide, sulla base del numero di flussi attivi e sul numero di pacchetti scambiati, come classificare il traffico. Ogni flusso presente nella tabella viene conteggiato nel contatore di flussi attivi solo se l'intervallo di tempo dall'ultimo pacchetto ricevuto è inferiore a 5 secondi. Inoltre, ogni flusso contribuisce al calcolo del numero totale di pacchetti scambiati solo se l'intervallo di tempo dall'ultimo pacchetto ricevuto è inferiore ad parametro, definito di "continuità", impostato attualmente a 20 secondi. Il filtro, ottenute queste statistiche, per determinare il tipo di traffico, non fa altro che dei semplici confronti tra questi valori ed i parametri di soglia passati tramite lo script Lua.

6.6 Filtro SMA

Il filtro si basa sul concetto di media mobile [47]. Per il suo funzionamento, è richiesto il mantenimento in memoria di una finestra di valori, che viene utilizzata per calcolare, all'arrivo di ogni pacchetto, il valore attuale della media dei bytes. Questo valore viene confrontato con una soglia, configurabile via Lua, per operare la caratterizzazione.

Dei parametri permettono di regolare il numero minimo di campioni richiesti ed il tempo massimo per l'arrivo dei pacchetti. Ogni qual volta questo tempo viene superato, all'interno della media vengono inseriti valori fittizi nulli per far in modo di abbassare il valore della media, possibilmente catalogando come rumore il traffico successivo.

La dimensione della finestra va stabila tenendo conto dell'occupazione della memoria. Una finestra grande permette infatti di ottenere una misura più significativa ma richiede più valori da dover salvare, aumentando la memoria richiesta da ogni singolo flusso. La dimensione attuale della finestra è di 10 valori. Essa non è configurabile da script perché fa parte delle strutture allocate staticamente per ogni flusso.

6.7 Traffico YouTube

YouTube è la piattaforma più usata per la condivisione nel web di contenuti multimediali. Come Facebook e Twitter, anche questa piattaforma presenta il problema dell'integrazione all'interno di altri siti web. La presenza di flussi YouTube, quindi, non è sempre indice di un'effettiva attività multimediale.

Dallo studio dei flussi di rete, tuttavia, è emerso che praticamente tutti i flussi video YouTube, attualmente, presentano un certificato SSL con indirizzo contenente la stringa "googlevideo.com". Data questa constatazione, si è catalogato come traffico multimediale tutti i flussi YouTube che presentano la data stringa. La restante parte sono invece classificati come normale traffico di navigazione web.

Capitolo 7 Persistenza Dati e Visualizzazione

7.1 RRDtool

L'utilizzo di un formato standard per il salvataggio dei dati delle attività utente rilevate ha il grande vantaggio di poter utilizzare strumenti già esistenti per l'accesso, l'aggiornamento e la visualizzazione dei dati. Lo strumento che è stato scelto per adempiere a queste funzioni è RRDtool [40].

RRDtool, acronimo di round-robin database tool, definisce un formato particolare opportunamente studiato per il salvataggio di contatori di metriche di rete. Il software si definisce round-robin perché, a differenza dei software di database utilizzati in altri ambiti, utilizza un archivio a dimensione fissa in cui i nuovi dati sostituiscono i vecchi come accade in un buffer circolare. Questa caratteristica, nell'ambito del monitoraggio delle reti, è molto importante, vista la possibilità di grandi quantità di dati in gioco. La dimensione di un archivio RRD viene quindi definita al momento della creazione e rimane sempre invariata.

Il funzionamento del software prevede la comunicazione periodica di dati grezzi. Eventuali dissimmetrie temporali tra l'intervallo imposto, chiamato step, e quello di arrivo dei dati viene gestito in modo automatico per mezzo di un'opportuna interpolazione. Questo dato interpolato viene chiamato PDP, primary data point, e, prima di essere registrato in maniera definitiva, viene aggregato con altri PDP tramite l'utilizzo di una funzione, definita di consolidamento, quale media, massimo, minimo. Il valore così ottenuto viene chiamato CDP, consolidated data point, e viene scritto in uno degli RRA, round robin archive.

Un RRA rappresenta la forma finale di memorizzazione dei dati. Al momento della creazione dell'archivio RRD, devono essere specificati il numero e le configurazioni degli archivi RRA che si intende usare. Per ogni archivio RRA va specificato:

- la funzione di consolidamento da applicare ai PDP
- il numero di PDP che questa deve aggregare in un unico CDP
- il numero di CDP da salvare

Con questi parametri, di fatto vengono specificate diverse risoluzioni temporali per ogni archivio. Normalmente, infatti, è interessante avere un'alta risoluzione per i dati recenti, mentre per quelli passati si può anche accettare una grana meno fine a vantaggio di uno storico di durata più lunga. La scrittura nel corretto RRA, così come la rotazione dello stesso, avviene in modo automatico in RRDtool.

7.2 Archivi Attività

Il software ntopng utilizza spesso RRDtool per il salvataggio di vari contatori e statistiche della rete. Per questo motivo, è stato relativamente semplice estendere il suo utilizzo alla memorizzazione delle attività utente. Lo schema proposto prevede l'utilizzo, per ogni host locale, di un numero di archivi RRD pari al numero delle attività da monitorare. Viene qui discussa la configurazione applicata ad ognuno di questi archivi.

Il parametro da cui partire per la configurazione è il valore di "step". Per sfruttare al meglio lo spazio di archiviazione è stato scelto di utilizzare il valore 60, corrispondente ad un minuto. Questa scelta ha reso anche possibile l'integrazione dell'aggiornamento dei contatori nello script periodico "minute.lua", eseguito da ntopng appunto ogni minuto.

La seconda scelta ha riguardato il numero ed il tipo di metriche da registrare. È stato scelto di riportare su disco tre diverse metriche:

- contatore dei bytes "attivi" inviati
- contatore dei bytes "attivi" ricevuti
- contatore di bytes di "rumore" inviati e ricevuti

Il traffico attivo viene salvato nelle sue due direzioni in forma distinta, mentre per il rumore non viene fatta distinzione.

RRDtool supporta diversi tipo di dato, DS o data source [41], da utilizzare per memorizzare le metriche. Vi sono i contatori semplici, derivati, assoluti, o gauge.

I contatori semplici, a differenza di quelli derivati, non possono essere decrementati e salvano la differenza tra i vecchi ed i nuovi valori. Quelli assoluti ripartono il conteggio, ad ogni dato, da zero. I gauge, invece, salvano i valori in sé piuttosto che la differenza tra questi. Da notare che tutte le DS, ad eccezione dei gauge, salvano il valore delle metriche come frazione di step. Questo, seppur sembri un dettagli implementativo, nella realtà si traduce nella necessità di moltiplicare i valori ottenuti da un interrogazione ad RRDtool per il valore di step per ottenere il valore reale.

La scelta della DS da usare per le tre metriche è ricaduta sui contatori semplici.

Infine è stato necessario stabilire i diversi livelli di risoluzione per gli archivi RRA. La formula

$$tempo_{coperto} = rows\ x\ (PDP\ per\ CDP)\ x\ step$$

indica come calcolare l'intervallo di tempo coperto, in minuti, da un archivio RRA. Il valore di step è fisso mentre i primi due fattori vanno scelti sulla base di copertura e risoluzione temporale desiderate. Il secondo fattore verrà di seguito indicato con il termine di "punti aggregati". Sono stati utilizzati tre archivi RRA con diversa risoluzione. Come funzione di consolidamento è stata usata in tutti e tre i casi la media.

- Il primo RRA ha la funzione di contenere i dati grezzi per minuto. Il numero di punti aggregati è quindi 1, ovvero non vi è aggregazione. La copertura temporale, a causa dell'alta risoluzione, non può essere molto ampia; 3 ore dovrebbe essere sufficienti. Il valore di rows vale (3*3600) / (1 * 60) ovvero 180 righe.
- Il secondo RRA deve garantire il giusto compromesso per una risoluzione accettabile e una copertura temporale media. Si è scelto, in un periodo di una settimana, di aggregare 60 PDA, che corrisponde ad una risoluzione di un'ora. Il parametro rows deve quindi valere (7*24*3600) / (60 * 60) ovvero 168 righe.
- Il terzo ed ultimo RRD deve permettere uno storico più ampio. Si è scelto di considerare un periodo di 90 giorni con una risoluzione di un giorno, 1440 punti aggregati. Il valore di rows deve essere quindi (90*24*3600) / (1440 * 60) ovvero 90 righe.

Gli archivi RRA, complessivamente, contengono quindi 438 record dati. Considerando l'utilizzo di 3 contatori, ognuno di 8 bytes, e la registrazione di 12 attività, la dimensione totale degli archivi RRD per singolo host è di circa 124 KB. La scrittura o meno su disco delle serie temporali relative alle macro-attività può essere controllata dalla pagina di preferenze di ntopng, dove è anche possibile specificare il periodo di tempo per cui mantenere i dati.

7.3 Salvataggio su Redis

Al fine di passare i dati relativi alle attività utente dal programma ntopng a RRDtool è necessaria una rappresentazione intermedia dei contatori, salvata a livello di host. Questi contatori, infatti, verranno continuamente aggiornati all'arrivo di nuovi pacchetti ma, solo una volta al minuto, il loro valore verrà comunicato a RRDtool ed effettivamente memorizzato. Inoltre, poiché è stato scelto come DS il tipo contatore, RRDtool necessita dei valori assoluti delle metriche.

Per far funzionare il tutto è stato quindi necessario salvare da qualche parte il valore attuale dei contatori. In ntopng, una funzione di serializzazione dell'host si occupa di salvare, al momento opportuno, questo tipo di informazioni su Redis [42], un key-value store residente in memoria, in formato JSON. Una funzione di de-serializzazione viene invece chiamata all'inizializzazione dell'oggetto per effettuare il lavoro opposto. Queste due funzioni sono state quindi estese per invocare le rispettive funzioni sull'oggetto UserActivityStats, imputato per il mantenimento e l'incremento dei contatori delle attività. In questo modo, i valori vengono sempre comunicati in modo corretto.

Un'ultima nota riguardo al funzionamento di RRDtool è la seguente. Quando ntopng non è attivo, oppure quando un host viene rimosso dalla memoria, i relativi archivi RRD non vengono aggiornati per un certo periodo di tempo. Al successivo aggiornamento, RRDtool si rende conto dal timestamp che questo è accaduto e inserisce all'interno del proprio database dei valori nan. Non avendo poi un valore precedente di confronto, nella pratica questo si manifesta col fatto che RRDtool salta la scrittura del primo minuto di traffico.

7.4 Visualizzazione

I valori delle attività attività rilevati e salvati negli archivi RRD possono essere visualizzati su un grafico. Per accedere alla visualizzazione è necessario collegarsi all'interfaccia web di ntopng, selezionare un host locale e infine spostarsi nella scheda "attività".

Figura 5: grafico delle attività

Il grafico è stato realizzato utilizzando cubism.js [61], una libreria basata su D3.js. La libreria permette la visualizzazione, in tempo reale, di serie temporali utilizzando un horizon chart [17].

Un horizon chart è un tipo di grafico che utilizza gradazioni di colore d'intensità crescente per visualizzare metriche che variano nel tempo. Grazie a questo espediente, la grandezza verticale di ogni serie temporale può essere ridotta, permettendo una visualizzazione compatta. Nell'implementazione usata, i valori di traffico in entrata sono visualizzati con colore azzurro e si sviluppano dalla parte superiore a quella inferiore del grafico. I valori di traffico in uscita, invece, si sviluppano nella direzione opposta e sono di colore verde.

Il grafico mostra su ogni riga un profilo attività. Dalla combo-box è possibile selezionare tra la visualizzazione di traffico attivo e quella di traffico di rumore di fondo. Muovendo il cursore sul grafico, oppure usando i tasti di controllo cursore, è possibile focalizzarsi su un tempo specifico e leggere i valori valori dei contatori per quel dato tempo. I valori sono negativi in caso di traffico ricevuto dall'host, positivi per il traffico inviato.

Il grafico è in grado di mostrare i dati a diverse risoluzioni temporali. Alcune risoluzioni sono indicate come "real-time" perché permettono di leggere, in tempo reale, le variazioni dei contatori direttamente da ntopng e non dagli archivi RRD. I dati real-time, data la loro natura, non hanno persistenza e vengono scartati alla chiusura della pagina web.

Capitolo 8 Verifica e Test

8.1 File di Cattura

L'elaborazione e l'implementazione delle strategie presentate in questo tirocinio si sono basate sull'analisi di traffico reale, catturato e catalogato in opportuni file di cattura. Per ogni file di cattura sono state indicate le attività manualmente simulate, in modo da poter correlare a queste i flussi generati.

I principali file di cattura utilizzati sono riportati nella tabella che segue. I file non sono disponibili pubblicamente; per un'eventuale richiesta di visione, si faccia riferimento all'indirizzo email emanuele.faranda94@gmail.com

File	Dimensione	Contenuto
facebook_idle0.pcap	144 KB	Facebook avviato ma in idle
facebook_integration0_informateci.pcap	400 KB	Integrazione Facebook informateci
facebook_integration1_stronghold.pcap	4 MB	Integrazione Facebook stronghold
facebook_integration2_cinema.pcap	2 MB	Integrazione Facebook cinema
facebook0.pcap	11 MB	Navigazione web, Facebook login, chat, upload
facebook1.pcap	4 MB	Navigazione Facebook, video playback
facebook2.pcap	51 MB	Navigazione Facebook
https0.pcap	10 MB	Navigazione web
https1_archiveforum	948 KB	Navigazione web su archiveforum
https2_amazon	582 KB	Navigazione web su Amazon
imap_starttls0.pcap	321 KB	IMAP con comando STARTTLS
imaps_gmail3_recv.pcap	28 KB	Sincronizzazione manuale GMail
imaps_gmail4_send.pcap	17 KB	Invio email tramite GMail
imaps0.pcap	577 KB	GMail e Hotmail in background
imaps1_hotmail_first_sync.pcap	9 MB	Prima sincronizzazione account Hotmail
openvpn0.pcap	2 MB	Navigazione web cifrata OpenVPN
openvpn1.pcap	2 MB	Navigazione web cifrata OpenVPN
openvpn2_idle.pcap	404 KB	OpenVPN idle
openvpn3_tcp_443.pcap	444 KB	OpenVPN con trasporto TCP su porta 443
openvpn4_udp_13680.pcap	389 KB	OpenVPN con trasporto UDP su porta 13680

File	Dimensione	Contenuto
ssh0.pcap	1 MB	Traffico SSH misto
ssh1_cat_less.pcap	755 KB	Traffico SSH con utility cat e less
ssh2_nano_cat_less.pcap	6.6 MB	Traffico SSH con utility nano, cat e less
streaming0.pcap	286 MB	Streaming OpenLoad e SendVid
streaming1.pcap	240 MB	Streaming SendVid com MIME type
twitter0.pcap	1 MB	Navigazione twitter
youtube0.pcap	13 MB	Navigazione YouTube e playback video
youtube1.pcap	104 MB	Playback vari video YouTube

8.2 Traffico IMAPS

La verifica del funzionamento del filtro “Sequenza di comandi” relativo al traffico IMAPS è stato effettuata su client Thunderbird.

Thunderbird, quando usato con account GMail, riutilizza spesso le connessioni esistenti sia per lo scaricamento di email che per la sincronizzazione. La caratterizzazione del traffico è avvenuta con successo nella maggior parte dei casi.

Per Hotmail, invece, Thunderbird crea una nuova connessione per ogni azione intrapresa dall’utente. Durante ogni sincronizzazione periodica, il client esegue una completa autenticazione, e viene quindi generato un grande volume di traffico. Le soglie utilizzate per la caratterizzazione, al fine di escludere correttamente il rumore, sono abbastanza elevate e accade a volte che parte del traffico attivo venga invece classificato come rumore. In generale, la classificazione avviene comunque con un buon grado d’accuratezza.

8.3 Traffico Multimediale e Facebook

Su browser Firefox, buona parte del traffico multimediale non cifrato specifica un MIME Type compatibile con quello multimediale e pertanto la sua classificazione avviene correttamente. Servizi come Spotify e Netflix, non specificatamente studiati, utilizzano il filtro SMA in modo generico e pertanto l’accuratezza di caratterizzazione riscontrata è medio-bassa.

Sempre su browser Firefox, il traffico Facebook è caratterizzato correttamente nella maggior parte dei casi. La caratterizzazione, per far fronte ai casi di forte integrazione in siti web di media collegati a Facebook, utilizza delle soglie di riconoscimento elevate e questo ha lo svantaggio di classificare come rumore parte del traffico attivo.

Su browser Chrome, l’ampio utilizzo del protocollo QUIC [13] impedisce il riconoscimento di parte del traffico di rete.

8.4 Android

Per cercare di capire quanto la soluzione proposta possa essere generalizzata, si è scelto di testare il funzionamento del software anche su un ambiente radicalmente diverso da quello d'analisi, usando del traffico proveniente da un dispositivo Android.

In merito alla posta elettronica, Android utilizza, per gli account GMail, delle connessioni HTTPS in alternativa a SMPT e IMAP, una situazione che, in mancanza di uno studio appropriato, impedisce la corretta caratterizzazione del traffico.

Il traffico web viene caratterizzato con un'accuratezza media. Alcuni servizi di push e di analytics che utilizzano le connessioni HTTPS come Pushwush e Mixpanel vengono correttamente classificati come rumore. Altri servizi, tuttavia, vengono talvolta classificati come traffico attivo. In particolare, il CDN di Viber, rilevato durante la fase di background dell'applicazione, viene classificato come traffico di chat attivo.

In definitiva, la metodologia andrebbe adattata a dovere per poter funzionare in modo adeguato anche su dispositivi mobili.

8.5 Performance

Per la misura della performance dell'implementazione fornita, si è utilizzata l'istruzione x86 rdtsc, Read Time-Stamp Counter [43], che accede ad un particolare contatore implementato in hardware all'interno di un registro MSR a 64bit.

Per ognuna delle funzioni filtro implementate, è stato conteggiato il numero di cicli di clock impiegati nell'esecuzione della stessa. La seguente tabella riassume i risultati ottenuti.

Funzione Filtro	Media cicli di clock	Numero rilevamenti
web	135	97616
sequenza comandi	170	10514
wma	235	89475
sma	319	90483
inter-flusso	525	8128

La funzione inter-flusso, utilizzata per la caratterizzazione di traffico Facebook e Twitter, è la più intensiva dal punto di vista computazionale. Questo è dovuto principalmente all'algoritmo di gestione della tabella che mantiene i flussi attivi per host.

Una seconda misurazione è stata fatta per l'invocazione della callback Lua flowProtocolDetected che, come già indicato, si occupa dell'assegnamento di profili e filtri ai flussi. Su 100 rilevamenti, la funzione consuma in media 38582 cicli di clock. Questo costo computazionale è principalmente dovuto all'overhead dell'ambiente Lua, e viene pagato solo una/due volte per flusso.

Sono stati rilevati dei colli di bottiglia in due casi. Quando ci sono tanti host locali attivi, dell'ordine di decine di migliaia, può presentarsi un'eccessiva attività su disco dovuta alla scrittura degli archivi RRD delle attività rilevate. In secondo luogo, quando il numero di pacchetti al secondo è molto elevato, il costo computazionale delle funzioni filtro, modesto se considerato a sé, diventa rilevante e può portare a perdite di pacchetti.

Bisogna comunque considerare che il contesto di monitoraggio in cui un amministratore possa essere interessato all'attività dei singoli host locali è normalmente limitato a reti di piccola o media dimensione e dunque con un numero limitato di host.

8.6 Occupazione della Memoria

Le misurazioni sono state effettuate su architettura x86_64 con compilatore gcc 6.2.1.

Per ogni flusso, vengono mantenute le informazioni sulla categoria del flusso, sul filtro attività applicato, sulla configurazione e stato attuale del filtro, per un totale di 120 bytes.

Per ogni host, vengono mantenuti i tre contatori per ognuna delle 12 attività individuate e le due tabelle utilizzate dal filtro inter-flusso per la caratterizzazione di Facebook e Twitter. In totale l'occupazione è di 608 bytes.

Infine, per stimare l'utilizzo della memoria da parte della macchina virtuale Lua, si è usata la funzione "collectgarbage" fornita da Lua. All'avvio della VM, la memoria allocata è di circa 60 KB. Ad ogni chiamata alla flowProtocolDetected, l'utilizzo di memoria cresce di circa 200 bytes, per poi abbassarsi nuovamente, a seguito dell'esecuzione periodica del garbage collector di Lua.

Complessivamente, l'utilizzo della memoria usata per l'implementazione delle strategie proposte può essere ritenuto soddisfacente.

Capitolo 9 Conclusioni

9.1 Obiettivi Raggiunti

Il traffico di rete è stato sintetizzato in maniera opportuna per mezzo delle macro-attività, creando di fatto un ulteriore livello di astrazione sui flussi di rete.

L'implementazione della soluzione proposta all'interno del software opensource ntopng, accessibile tramite un normale browser web, fornisce uno strumento immediatamente utilizzabile dall'amministratore di rete per visionare il comportamento degli host locali. Il grafico delle attività, visualizzando i dati rilevati su base temporale, permette d'individuare, a colpo d'occhio, anomalie sul traffico di rete.

Sono state elaborate metodologie innovative di caratterizzazione del traffico cifrato, basate sulle metriche di rete, ben diverse dai tradizionali metodi basati su firma del protocollo utilizzati nella stragrande maggioranza dei software di monitoraggio. Grazie all'impiego di queste tecnologie, è stato possibile caratterizzare con un discreto grado d'accuratezza il traffico IMAPS, HTTPS, Facebook e Twitter nei contesti analizzati.

9.2 Lavoro Futuro

9.2.1 Rilevamento Anomalie

I trojan più sofisticati oggi utilizzano connessioni SSL per cifrare la comunicazione con il computer di comando. Servizi come Yahoo e GMail sono spesso utilizzati per nascondere la loro attività [52]. Caratterizzare questo traffico risulta quindi essere essenziale per il loro rilevamento.

Le informazioni sul traffico attivo degli host, accessibili tramite script Lua, potrebbero essere utilizzate per la creazione di regole specifiche che descrivano, su base temporale, le macro-attività ammesse da parte di un host locale. Ad esempio, definendo una politica per cui l'accesso remoto al computer locale possa avvenire solo durante le ore di lavoro, si potrebbero rilevare intrusioni o tentativi d'intrusione in ore notturne. Una regola che stabilisce che le connessioni VPN possono avvenire solo da o verso una determinata rete aziendale potrebbe rilevare invece un'azione non autorizzata da parte di un software o di un dipendente. L'impiego di queste regole è effettivamente possibile proprio grazie alla caratterizzazione del traffico attivo operata durante il tirocinio.

Un altro modo di rilevare le anomalie è tramite il confronto dell'attività di un host con quella attesa, come avviene in software come Darktrace [60]. Darktrace utilizza in modo estensivo l'apprendimento automatico per rilevare lo stato normale della rete. Il software permette di visualizzare lo stato della rete in 3D ed interagire con essa. La possibilità di un attacco viene espressa con una percentuale di confidenza.

Questa metodologia si compone normalmente di due fasi. Per un certo periodo di tempo, si monitora l'host cercando di determinare gli orari e la frequenza delle attività svolte. Una volta stabilito quale sia il suo comportamento normale, si può procedere alla fase operativa. Avendo individuato le attività critiche, si generano degli alert qualora queste si presentino in maniera molto differente dal normale modello.

Questo approccio però dà luogo a diversi problemi. In primo luogo, l'attività di un host varia nel tempo ed è soggetta a periodicità variabile, pertanto la fase d'apprendimento dello stato normale della rete va in qualche modo ripetuta. Ma come si fa ad essere sicuri che, durante l'apprendimento, che un'eventuale minaccia non sia già presente nella rete? Si potrebbe procedere fornendo, per ogni attività critica, un modello generico di base molto pedante, che dia però la possibilità di mascherare manualmente alcune periodicità o livelli d'attività. Il sistema dovrebbe quindi essere abbastanza intelligente da derivare, grazie ai suggerimenti inseriti, il modello specifico della rete.

9.2.2 Caratterizzazione del Traffico SSL

Come testimoniato dai diversi articoli scientifici presentati, il futuro del monitoraggio di rete delle nuove applicazioni, in vista di una sempre maggiore applicazione di metodi di cifratura, passa attraverso lo studio e l'elaborazione delle metriche di rete. Le tecniche proposte per la caratterizzazione del traffico SSL possono essere considerate un punto di partenza per l'elaborazione di metodologie più avanzate.

La sfida, nel contesto del monitoraggio di rete, è sviluppare metodologie al tempo stesso precise ed efficienti, che possano essere applicate nell'elaborazione di grandi quantità di dati.

Software come Weka [29] mettono alla portata di tutti la potenza delle tecniche di apprendimento automatico, che si rivelano particolarmente utili nell'individuazione delle caratteristiche salienti per la classificazione dei flussi. Queste tecniche richiedono uno studio appropriato per essere contestualizzate nell'ambito del monitoraggio di rete e, da sole, non rappresentano una soluzione al problema.

9.3 Offuscare il Traffico

Vista l'attenzione della comunità scientifica alle problematiche di riconoscimento del traffico di rete, è probabile che nel futuro prossimo verranno proposti nuovi metodi innovativi di riconoscimento del traffico. In parallelo, però, si lavora allo sviluppo di nuove tecniche di offuscamento, in particolar modo per quello che riguarda le VPN.

Utilizzare configurazione inusuali, inserire il traffico OpenVPN all'interno di tunnel d'offuscamento [44], creare un nuovo strato di rete ed instradare i pacchetti in modo che appaiano su flussi diversi [32] sono solo alcune delle tecniche proposte.

Tutte le metodologie per l'identificazione del traffico di rete possono, in un modo o nell'altro, essere aggirate. Ad esempio, le tecnologie DPI sono suscettibili a cambiamenti nelle intestazioni dei pacchetti, come pure quelle basate su fingerprint, alla variazione di porta del servizio e, ovviamente, alla cifratura.

D'altronde, le tecnologie basate sull'analisi delle metriche associate ai pacchetti di rete, come quella proposta in questo tirocinio, possono anch'esse essere aggirate, ad esempio variando in modo casuale la dimensione dei pacchetti oppure inserendo ritardi d'invio.

Agire sulle metriche di rete per offuscare il traffico, tuttavia, può avere degli effetti collaterali. Alcune applicazioni real-time come quelle usate per le comunicazioni VoIP o i giochi online, ad esempio, sono molto sensibile ai ritardi di rete, pertanto nuovi metodi d'offuscamento dovranno tenerne conto.

9.4 Competenze Tecniche Acquisite

Il lavoro di tirocinio ha permesso l'acquisizione di varie competenze nel contesto delle reti di calcolatori.

Sono stati utilizzati programmi e tecnologie per la cattura, l'analisi del traffico di rete e la sua relazione con statistiche e metriche comunemente usate nell'ambito del monitoraggio. È stato appreso parte del funzionamento interno di software affermati come ntopng e nDPI, di come implementare dissector per nuovi protocolli, di come estendere le funzionalità di questi programmi per l'integrazione di nuove tecniche analitiche.

Si è appreso come configurare, sia in modo automatico che manuale, ed utilizzare il software OpenVPN, in particolar modo su un server remoto. Sono state approfondite ed estese le conoscenze sull'utilizzo del software di sviluppo collaborativo git relative al fork di progetti, alla creazione di branch di sviluppo, al merge di branch e alla creazione di pull requests.

È stata appresa la programmazione basilare in linguaggio Lua, il suo utilizzo come linguaggio di scripting all'interno di progetti complessi tramite l'integrazione con il codice C. Il lavoro d'analisi dei protocolli e delle tecniche per il riconoscimento del traffico di rete è stato condotto analizzando documenti scientifici, consultando le relative RFC ed effettuandone un'applicazione pratica.

Capitolo 10 Riferimenti Bibliografici

[1] Alberto Dainotti, Walter de Donato, Antonio Pescapè, Pierluigi Salvo Rossi, Classification of Network Traffic via Packet-LevelHidden Markov Models, wpage.unina.it/walter.dedonato/pubs/tc_globecom08.pdf 2008

[2] Andrew W. Moore, Denis Zuev, Internet Traffic Classication Using Bayesian Analysis Techniques, http://www.cl.cam.ac.uk/~awm22/publications/moore2005internet.pdf 2005

[3] Andrew W. Moore, Konstantina Papagiannaki, Toward the Accurate Identication of Network Applications, http://www.cl.cam.ac.uk/~awm22/publications/moore2005toward.pdf 2005

[4] Anthony McGregor, MarkHall, PerryLorier, andJamesBrunskill, Flow Clustering Using Machine Learning Techniques, https://www.cl.cam.ac.uk/research/srg/netos/events/pam2004/papers/166.pdf 2004

[5] Augustin Soule, Kavé Salamatian, Nina Taft, Richard Emilion, Konstantina Papagiannaki, Flow Classification by Histograms, http://www.univ-orleans.fr/mapmo/membres/emilion/publ/Sigm.pdf 2004

[6] Using TLS with IMAP, POP3 and ACAP, https://tools.ietf.org/html/rfc2595

[7] Carl Livadas, Bob Walsh, David Lapsley, Tim Strayer, Using Machine Learning Techniques toIdentify Botnet Traffic, http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.71.887&rep=rep1&type=pdf 2006

[8] Charles V. Wright, Fabian Monrose, Gerald M. Masson, On Inferring Application Protocol Behaviors in Encrypted NetworkTraffic, http://www.jmlr.org/papers/volume7/wright06a/wright06a.pdf 2006

[9] Charles V. Wright, Lucas Ballard, Fabian Monrose, Gerald M. Masson, Language Identification of Encrypted VoIP Traffic, http://static.usenix.org/events/sec07/tech/full_papers/wright/wright_html/ 2007

[10] Cisco, NBAR2 (Next Generation NBAR) Protocol, http://www.cisco.com/c/en/us/products/collateral/ios-nx-os-software/network-based-application-recognition-nbar/qa_C67-723689.html

[11] Cisco, Joy – A package for capturing and analyzing network flow data, https://github.com/davidmcgrew/joy

[12] Félix Hernández-Campos, F. Donelson Smith, Kevin Jeffay, Statistical Clustering of Internet Communication Patterns, http://www.cs.unc.edu/~jeffay/papers/INTERFACE-03.pdf 2003

[13] Google, QUIC, a multiplexed stream transport over UDP, https://www.chromium.org/quic

[14] Guofei Gu, Roberto Perdisci, Junjie Zhang, Wenke Lee, BotMiner: Clustering Analysis of Network Traffic, http://static.usenix.org/events/sec08/tech/full_papers/gu/gu_html/ 2008

[15] Ipoque, PACE, https://ipoque.com/products/pace/

[16] Jeffrey Erman, Martin Arlitt, Anirban Mahanti, Traffic Classification Using Clustering Algorithms, http://www.ce.uniroma2.it/courses/MMI/memopaper2.pdf 2006

[17] Jeffrey Heer, Nicholas Kong, Maneesh Agrawala, Sizing the Horizon: The Effects of Chart Size and Layering on the Graphical Perception of Time Series Visualizations, http://vis.berkeley.edu/papers/horizon/ 2009

[18] Laurent Bernaille, Renata Teixeira, Kavé Salamatian, Early Application Identification, http://conferences.sigcomm.org/co-next/2006/Conext06_Proceedings/papers/f17.pdf 2006

[19] Luca Deri, ntopng – github.com, https://github.com/ntop/ntopng

[20] Luca Deri, nDPI – github.com, https://github.com/ntop/nDPI

[21] Luca Vassio, Idilio Drago and Marco Mellia, Detecting User Actions from HTTP Traces:Toward an Automatic Approach, www.tlc-networks.polito.it/mellia/papers/HTTPTrac2016.pdf 2016

[22] Maciej Korczynski, Andrzej Duda, Markov Chain Fingerprinting to Classify Encrypted Traffic, http://drakkar.imag.fr/IMG/pdf/1569811033.pdf 2014

[23] Manuel Crotti, Maurizio Dusi, Francesco Gringoli, Luca Salgarelli, Traffic Classification through Simple StatisticalFingerprinting, http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.435.1664&rep=rep1&type=pdf 2007

[24] Martino Trevisan, Idilio Drago, Marco Mellia, Maurizio M. Munafò, Towards Web Service Classification using Addresses and DNS, www.tlc-networks.polito.it/mellia/papers/ClueTrac2016.pdf 2016

[25] Michal Zalewski, fl0p - passive L7 flow fingerprinting, http://seclists.org/honeypots/2006/q4/61 2006

[26] Riyad Alshammari, A. Nur Zincir-Heywood, Machine Learning Based Encrypted Traffic Classification: Identifying SSH and Skype, https://www.researchgate.net/profile/Riyad_Alshammari/publication/224092005_Machine_learning_based_encrypted_traffic_classification_Identifying_SSH_and_Skype/links/09e4150fe7046df8bd000000.pdf 2009

[27] Sebastian Zander, Thuy Nguyen, Grenville Armitage, Automated Traffic Classification and Application Identification using Machine Learning, https://www.researchgate.net/profile/Sebastian_Zander2/publication/4198505_Automated_traffic_classification_and_application_identification_using_machine_learning/links/00b49526129b0a51cb000000.pdf 2005

[28] Subhabrata Sen, Oliver Spatscheck, Dongmei Wang, Accurate, Scalable In-Network Identification of P2P Traffic Using Application Signatures, http://ramb.ethz.ch/CDstore/www2004/docs/1p512.pdf 2004

[29] The University of Waikato, Weka 3: Data Mining Software in Java, http://www.cs.waikato.ac.nz/ml/weka/

[30] The University of Waikato, libprotoident, https://github.com/wanduow/libprotoident

[31] Vern Paxson, Lawrence Berkeley Laboratory and EECS Division, Empirically-Derived Analytic Models of Wide-Area TCP Connections, http://www.icir.org/vern/papers/WAN-TCP-models.pdf 1994

[32] Yuzhi Wang, Ping Ji, Borui Ye, Pengjun Wang, Rong Luo, Huazhong Yang, GoHop: Personal VPN to Defend from Censorship, www.icact.org/upload/2014/0096/20140096_finalpaper.pdf 2014

[33] Wireshark, https://www.wireshark.org/

[34] Why TCP Over TCP Is A Bad Idea, http://sites.inka.de/sites/bigred/devel/tcp-tcp.html

[35] Traffic Flow Measurement: Architecture, http://www.ietf.org/rfc/rfc2722.txt

[36] The Transport Layer Security (TLS) Protocol Version 1.3, https://tools.ietf.org/html/draft-ietf-tls-tls13-14

[37] The Transport Layer Security (TLS) Protocol Version 1.2 – Alert Messages, https://tools.ietf.org/html/rfc5246#appendix-A.3

[38] The Tor project, https://www.torproject.org/

[39] tcpdump, http://www.tcpdump.org/

[40] RRDtool, http://oss.oetiker.ch/rrdtool/

[41] rrd-beginners, http://oss.oetiker.ch/rrdtool/tut/rrd-beginners.en.html

[42] Redis, www.redis.io

[43] Read Time-Stamp Counter, http://x86.renejeschke.de/html/file_module_x86_id_278.html

[44] OpenVPN traffic obfuscation, https://community.openvpn.net/openvpn/wiki/TrafficObfuscation

[45] OpenVPN Security Overview, https://openvpn.net/index.php/open-source/documentation/security-overview.html

[46] OpenVPN Protocol, https://wiki.wireshark.org/OpenVPN

[47] Moving average, https://en.wikipedia.org/wiki/Moving_average

[48] Media type, https://en.wikipedia.org/wiki/Media_type

[49] ISPs Removing Their Customers' Email Encryption, https://www.eff.org/deeplinks/2014/11/starttls-downgrade-attacks

[50] Internet Message Access Protocol - Version 4rev1, https://tools.ietf.org/html/rfc3501

[51] IMAP4 IDLE command, https://tools.ietf.org/html/rfc2177

[52] IcoScript rat controlled via email services, including Yahoo and Gmail, https://www.invincea.com/2014/08/icoscript-rat-controlled-via-email-services-including-yahoo-and-gmail/

[53] HTTPS Everywhere, https://www.eff.org/https-everywhere

[54] HTTP 1.1 Header Field Definitions, https://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html

[55] Google, Yahoo SMTP email severs hit in Thailand, http://www.telecomasia.net/content/google-yahoo-smtp-email-severs-hit-thailand

[56] Google Chrome will banish Chinese certificate authority for breach of trust, http://arstechnica.com/security/2015/04/google-chrome-will-banish-chinese-certificate-authority-for-breach-of-trust/

[57] Forward secrecy, https://en.wikipedia.org/wiki/Forward_secrecy

[58] DigitNotar fraud, https://en.wikipedia.org/wiki/DigiNotar

[59] Deep Packet Introspection, https://en.wikipedia.org/wiki/Deep_packet_inspection

[60] Darktrace, https://darktrace.com/

[61] Cubism.js, https://square.github.io/cubism/

[62] Comparison of TCP/IP and OSI layering, https://en.wikipedia.org/wiki/Internet_protocol_suite#Comparison_of_TCP.2FIP_and_OSI_layering

[63] Articolo 617 Quinquies, http://www.diritto24.ilsole24ore.com/guidaAlDiritto/codici/codicePenale/articolo/875/art-617-quinquies-installazione-di-apparecchiature-atte-ad-intercettare-impedire-od-interrompere-comunicazioni-informatiche-o-telematiche.html

[64] Articolo 617 Quater, http://www.diritto24.ilsole24ore.com/guidaAlDiritto/codici/codicePenale/articolo/874/art-617-quater-intercettazione-impedimento-o-interruzione-illecita-di-comunicazioni-informatiche-o-telematiche.html?refresh_ce=1

[65] Application Layer Packet Classifier for Linux, http://l7-filter.sourceforge.net/