Università degli Studi di Perugia

DIPARTIMENTO DI MATEMATICA E INFORMATICA
Corso di Laurea Magistrale in Informatica

Network Forensics

Studente:
Fabio Mignogna

Professore:
Stefano Bistarelli

Anno Accademico 2009-2010

Indice

Sommario

In questa relazione illustreremo lo stato dell'arte della **Network Forensics**. Verrà poi presentato una tecnica con la quale i criminali informatici commettono i loro reati: l'**anonimato**. Analizzeremo uno strumento di anonimato, **Tor**, e illustreremo le falle con cui gli addetti alla Network Forensics faranno le loro indagini.

Introduzione

La chiave principale del business di un'azienda è certamente l'apertura verso Internet. Con questa apertura, però, le aziende (e non solo) sono esposte ad attacchi di ogni tipo. La Computer Forensics si è evoluta e si è adattata ai cambiamenti tecnologici e allo studio dei crimini su Internet. Ecco la Network Forensics (NF). Prima di analizzare in dettaglio cosa fa la NF, diamo sua la definizione apparsa per la prima volta nel 1997 in un articolo di Marcus Ranum[1]:

> *...è il prelievo, la memorizzazione e l'analisi degli eventi di rete al fine di identificare la sorgente degli attacchi alla sicurezza o l'origine di altri problemi del sistema di rete...*

Come si può vedere si tratta sempre di analisi “dopo il fatto“ ma orientata ai problemi di sicurezza e non solo a quelli legali. Andremo a sviscerare la definizione analizzando ogni termine. Prima di addentrarci in questa analisi, faremo un cenno sulla differenza fra Network Security (NS) e Network Forensics.

1 Network Security vs Network Forensics

La differenza sostanziale fra NS e NF sta nei tools utilizzati. Gli strumenti della NS non si occupano, in genere, di raccogliere prove digitali, ma di segnalare eventi. Ad oggi sono stati sviluppati diversi Network Forensic Analysis Tools (**NFAT**) che, oltre ad analizzare flussi di dati ed eventi, consentono dopo il fatto di averne evidenze utili in ambito dibattimentale.

1.1 NFAT

Esistono strumenti per monitorare il traffico in tempo reale, ma sono dispendiosi in termini di risorse hardware e umano. Non sono adatti alle reti più grandi di un singolo workgroup. Risulta più pratico, invece, archiviare tutto il traffico e analizzare un sottoinsieme. Gli NFAT possono fornire una visione più ricca dei dati raccolti che consente di ispezionare il traffico da un livello più alto e astratto dallo stack del protocollo.

Di solito hanno incorporato un software di sniffing e un sistema di analisi dei pacchetti raccolti. Il più delle volte usano *tcpdump* come sniffer dei pacchetti, che è un formato *de facto*. Questi strumenti sono stati sviluppati al solo fine forense e di indagine sugli incidenti e non sono destinati agli amministratori di rete. La *Figura 1* ci mostra l’idea che sta dietro agli strumenti della NF.

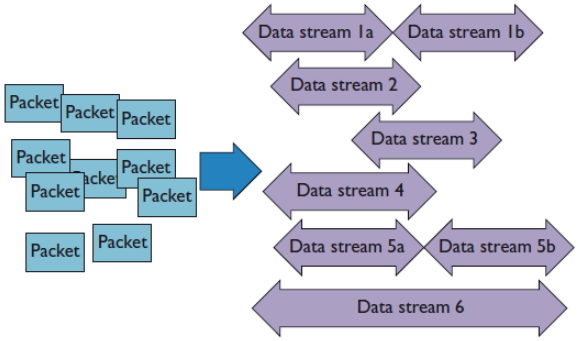

Figura 1: Uno strumento di network forensics organizza in singoli pacchetti di connessioni a livello di trasporto tra le macchine, permettendo così di analizzare il traffico di rete strato dopo strato.

1.1.1 Esempi di strumenti NFAT

Ecco alcuni NFAT:

- Xplico (Open Source)[2]
- E-Detective[3]
- Network Miner (Open Source)[4]
- . . .

2 Identificare i dati utili

La maggior parte delle attività svolte nella Computer Forensics (CF) conduce verso un ambito di rete, ergo verso attività nella NF. Basti pensare alla sola navigazione, e-mail, chat, software P2P, FTP, VoIP, Telnet, Community, VPN, ecc. Molte attività della NF hanno come conclusione il sequestro di una o più memorie di masse e quindi si ricade di nuovo nella CF. Una buona interazione fra NF e CF permette di non allungare i tempi di risoluzione dei casi.

Cosa vuol dire, allora, identificare i dati utili al fine di un indagine su una rete?

1. verificare quali sistemi e supporti sono interessanti (switch, router, ISP, ecc.);
2. trovare gli elementi che testimoniano una determinata attività sulla rete (LOG, report IDS, ecc.);
3. trovare macchine o supporti di memoria da "girare" alla CF per descrivere cosa è accaduto ed a prelevare tali elementi probatori.

2.1 Genuinità dei dati

Qualora non sia possibile far intervenire la CF perché gli elementi che dimostrano un crimine sono un prodotto della rete che non risiede specificamente su un nodo fisico, la copia dei dati rilevanti è un processo di difficile interpretazione e con scarse garanzie. Nel CF le garanzie si esplicano in quanto la copia forense viene certificata mediante un codice di verifica che ne assicura la genuinità prima e dopo l'analisi (hash). Nel NF i dati rilevanti sono spesso frutto dell'elaborazione congiunta di dati provenienti da diversi nodi e non semplicemente di una copia, da cui la certificazione mediante hash non ha molto senso. Pensiamo, ad esempio, ai sistemi distribuiti.

Come si certificano i dati? Si deve certificare il processo di ottenimento assieme allo stato (di natura dinamico) della rete nel momento del prelievo.

2.2 Di cosa fare la copia?

La rete è formata da tanti nodi, che sono dei semplici computer. Se si considera un nodo il problema, la soluzione più semplice sarebbe quella di fare il shutdown della macchina, sequestrarla e agire secondo la CF. Questo tipo di approccio ha, però, un "piccolo" problema: molti dei nodi di una rete non possono ammettere lo shutdown senza compromettere definitivamente informazioni e processi che viaggiano sulla rete stessa da cui si arriverebbe all'assurdo di preservare lo stato della singola macchina alterando quello della rete.

In questo caso l'unica via d'uscita è la *copia a runtime*, ossia durante la normale attività del nodo e della rete. Sono pochi, però, gli strumenti hardware e software in grado di realizzare un'attività di questo tipo (si tratta di estensioni dei normali software utilizzati nella CF). Dal punto di vista legale si aprono problematiche notevoli: la copia a runtime di dati su una memoria di massa operativa è un'operazione irripetibile, perchè ricostruire le condizioni globali entro le quali si svolge è praticamente irrealizzabile.

Le operazioni irripetibili sono problematiche da diversi punti di vista:

1. sono difficili da documentare e certificare;
2. sono difficili da presentare in fase di dibattimento dei risultati ottenuti.

2.3 Quali dati recuperare?

Dopo aver fatto la copia, sorge spontanea la domanda: quali dati ci interessa recuperare? La risposta è: dipende.

È molto difficile e complesso chiedersi *quali* dati recuperare. Entra in gioco un grado di aleatorietà notevole che rendono i dati più indizi che elementi probatori. Ad esempio si può recuperare:

1. le entry nei file di log
2. le password per accedere alle sessioni di lavoro
3. . . .

3 Anonimato in rete

Analizzeremo, ora, un modo tramite il quale vengono commessi reati informatici, cioè l'**anonimato**.

Citando l'esaustivo Wikipedia[5]:

> *L'anonimato (o anche anonimìa) è lo stato di una persona anonima, ossia di una persona di cui l'identità non è conosciuta. Questo può accadere per diversi motivi: una persona è riluttante a farsi conoscere, oppure non lo vuole per motivi di sicurezza come per le vittime di crimini e di guerra, la cui identità non può essere individuata.*

L'anonimato ha, però, vari problemi:

> *Nascondere la propria identità può essere una scelta, per legittime ragioni di privacy e, in alcune occasioni, per sicurezza personale: un esempio ne sono i criminali, i quali, solitamente, preferiscono rimanere anonimi, in particolare nelle lettere ricattatorie.*

4 The Onion Router

Tor[6] (The Onion Router) è un sistema di comunicazione anonima per internet basato sulla seconda generazione del protocollo di *onion routing*. Permette di difendersi dalle tecnologie di sorveglianza della rete, come l'analisi del traffico, deviando le comunicazioni attraverso una rete distribuita di server (*relay*) gestiti da volontari in tutto il mondo.

Tor funziona con molti dei programmi di uso quotidiano, come i browser, i client per la chat, i programmi di login remoto e tante altre applicazioni basate sul protocollo TCP. Tor, infatti, funziona esclusivamente su TCP.

Tor permette anche di creare dei servizi nascosti nella sua rete, denominati *hidden services*.

È un progetto sicuramente nobile, purtroppo usato anche dai criminali informatici.

4.1 Funzionamento di Tor

Tor aiuta a ridurre i rischi derivati dall'analisi del traffico usando un'idea simile ad usare un percorso tortuoso e difficile da seguire, cancellando periodicamente le informazioni. Tor non crea un path diretto dalla sorgente alla destinazione: i pacchetti dati nella rete Tor prendono un percorso casuale attraverso molti

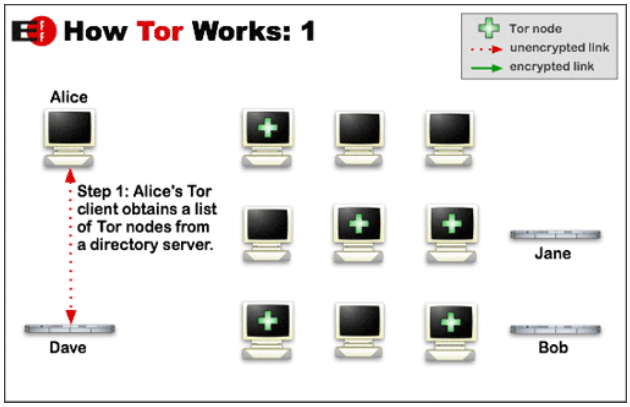

Figura 2

relay, in modo che nessun attaccante situato in un singolo punto possa dire da dove venga o dove sia diretto un certo traffico.

Per creare un path di rete privato con Tor, il software crea incrementalmente un circuito di connessioni cifrate, tramite TLS, attraverso i relay della rete Tor. Il circuito viene esteso un salto alla volta, e ogni relay lungo il percorso conosce solo quale relay gli ha dato le informazioni, e verso che relay inoltrarle. Nessun relay conosce il percorso completo che il pacchetto ha preso. Il software negozia un nuovo insieme di chiavi crittografiche per ogni salto lungo il circuito, per assicurarsi che ciascun nodo non possa tracciare queste connessioni durante il passaggio. La negoziazione delle chiavi segue lo schema Diffie-Hellman.

Una volta che un circuito è stato stabilito, si possono scambiare diversi tipi di dati e usare molti tipi di applicazioni attraverso una rete Tor. Tor funziona solo con i flussi TCP e può essere usato da ogni applicazione che abbia il supporto SOCKS (ad esempio il Privoxy, dove diamo un accenno dopo). La *Figura 3* ci mostra il funzionamento della rete Tor.

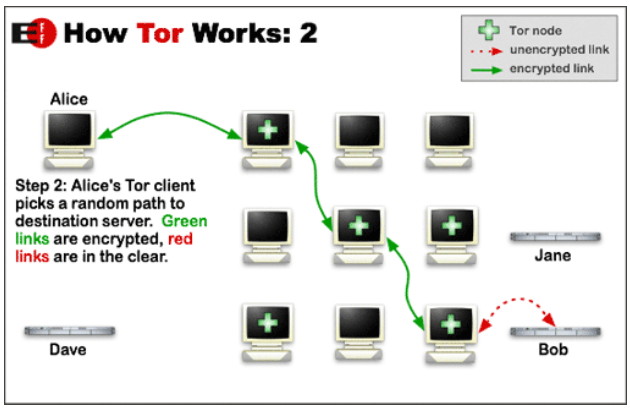

Figura 3

Tor utilizza lo stesso circuito per le connessioni che avvengono nell'arco di dieci minuti, questo per evitare che nessuno possa collegare le azioni precedenti con le successive. La *Figura 4* ci mostra il cambio di path dopo dieci minuti.

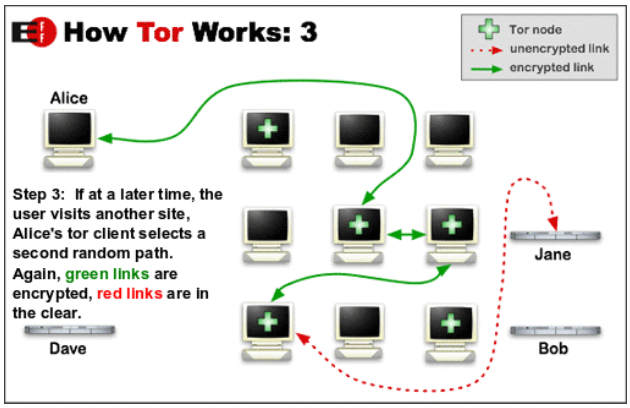

Figura 4

Come si nota, l'ultimo passo, cioè exit node-destinazione, non è cifrato. Tutte le informazioni sono in chiaro.

4.1.1 Accenni su Privoxy

Privoxy[7] è un programma di proxy web, spesso usato in combinazione con Tor. Ha funzionalità di filtro per la protezione della privacy, per la modifica dei dati delle pagine web, per la gestione dei cookies, per il controllo degli accessi, e per la rimozione selettiva di contenuti come annunci, banner e pop-up.

Si tratta di un software estremamente potente che permette una configurazione estremamente accurata delle proprie regole, ma che anche nella sua configurazione più semplice dimostra la sua efficacia. Questa flessibilità lo rende adatto sia ad un uso personale (installandolo ad esempio sul nostro PC) che ad un uso in ambienti di rete più complessi (dal piccolo ufficio alla LAN più estesa).

4.2 Hidden Services

Tor consente ai propri utenti di offrire vari servizi (web server, chat server, ecc.) nascondendo la propria posizione nella rete. Grazie ai *rendezvous point* è possibile far utilizzare questi servizi agli altri utenti Tor senza conoscere la reciproca identità. Vediamo i passi per creare un hidden services.

Un hidden service deve rendere nota la sua esistenza nella rete Tor, altrimenti non può essere acceduto. Per questo il servizio sceglie alcuni relay a caso, stabilisce dei circuiti (e non connessioni dirette) verso di essi e chiede loro di fungere da *introduction point* comunicandogli la sua chiave pubblica. Creando questa connessione è difficile stabilire se questi introduction point sono associati a qualche hidden services. In più gli introduction point non conoscono l'IP del servizio, ma solo la sua chiave pubblica. La *Figura 5* ci mostra il passo.

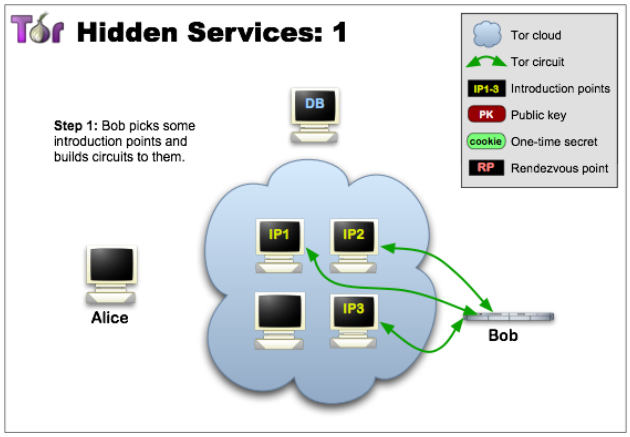

Figura 5

Una volta scelti gli introduction point, l'hidden service costruisce un *hidden service descriptor*, che non è nient'altro che un pacchetto contenente la sua chiave pubblica ed un sommario degli introduction point, e firma questo descriptor con la sua chiave privata. Invia il descriptor a un gruppo di directory server, usando sempre un circuito Tor. Il descriptor verrà trovato dai cient che richiederanno XYZ.onion, dove XYZ è un nome di 16 caratteri derivato in modo unico dalla chiave pubblica dell'hidden service. Dopo questo passo, l'hidden service è attivo. La *Figura 6* ci mostra il passo.

Quando un client desidera contattare un hidden service, deve conoscere prima il suo indirizzo onion. Dopodiché il client può iniziare a stabilire la connessione scaricandone il descrittore dai directory server. Se esiste un descrittore per XYZ.onion, il client ora conosce il gruppo di introduction point e la corretta chiave pubblica. Il client crea, allora, un circuito verso un altro relay scelto a caso e gli chiede di fungere da *rendezvous point* comunicandogli un *one-time secret*. La *Figura 7* ci mostra il passo.

Una volta presente il descriptor e pronto il rendezvous point, il client costruisce un *introduce message* (cifrato con la chiave pubblica dell'hidden service) contenente l'indirizzo del rendezvous point ed il one-time secret. Il client invia questo messaggio a uno degli introduction point, chiedendo che venga

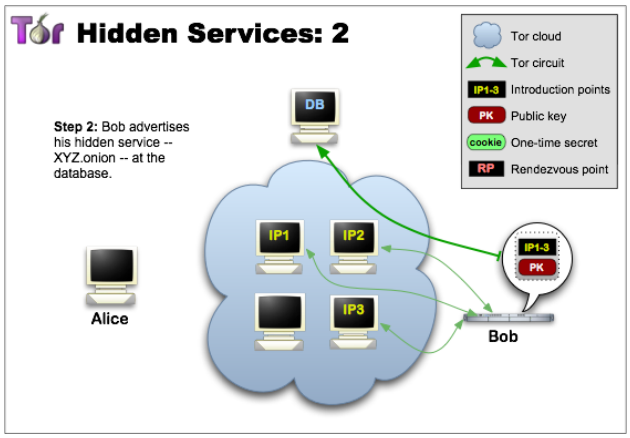

Figura 6

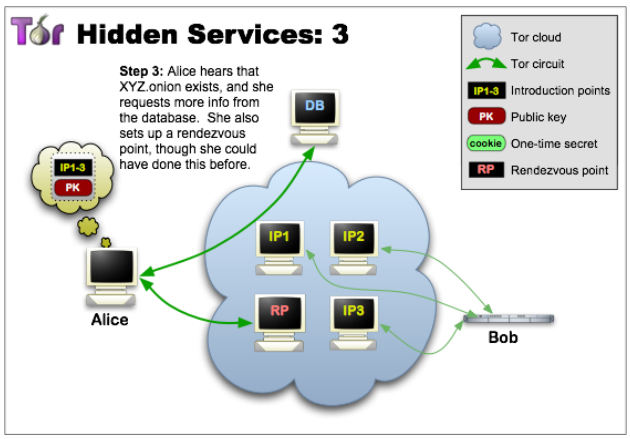

Figura 7

consegnato all'hidden service. È inutile ribadire che ogni scambio di messaggi avviene sempre tramite un circuito Tor. La *Figura 8* ci mostra il passo.

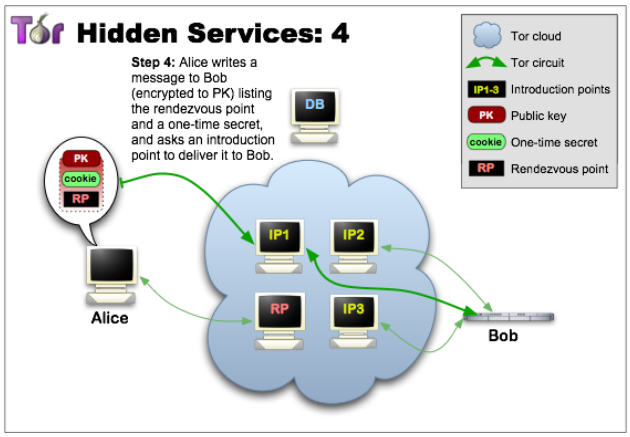

Figura 8

L'hidden service decifra l'introduce message del client e scopre l'indirizzo del rendezvous point ed il one-time secret contenuto. Il service crea un circuito verso il rendezvous point e gli invia il one-time secret in un *rendezvous message*. La *Figura 9* ci mostra il passo.

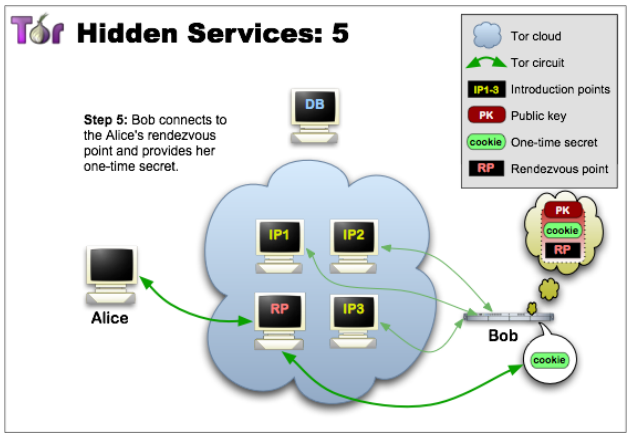

Figura 9

Il rendezvous point notifica al client che la connessione è stata stabilita con successo. Dopodiché il client e l'hidden service possono usare i loro circuiti verso il rendezvous point per comunicare tra di loro. Il rendezvous point inoltra semplicemente i messaggi dal client al service e viceversa. La *Figura 10* ci mostra il passo.

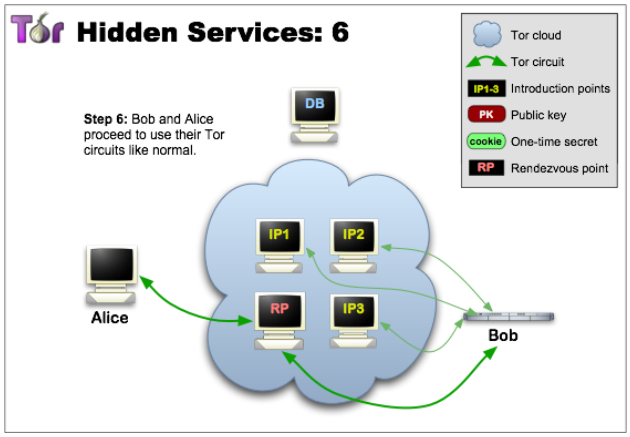

Figura 10

5 Vulnerabilità della rete Tor

Come la NF può effettuare analisi? Il metodo brutale è quello di sequestrare i server Tor, come è avvenuto in Germania nel 2006[8] oppure sfruttare le sue vulnerabilità.

Timing attacks Tor è un sistema a bassa latenza, quindi sarebbe possibile correlare una connessione cifrata di partenza con una connessione in chiaro di destinazione;

DNS Leak Molti software continuano a fare richieste DNS bypassando la rete Tor;

Software malconfigurati Ad esempio, nei web browser si lasciano attivi le opzioni che permettono il caricamento dei file flash, degli script javascript e la memorizzazione dei cookie;

Connessioni dirette dei software di messaggistica istantanea Alcuni client di messaggistica istantanea che non supportano l'interfaccia SOCKS, bypassano la rete Tor. L'unico modo per sfruttare Tor è quello i usare Privoxy;

Intercettazione dell'exit node Essendo l'ultimo collegamento non cifrato, è possibile un attacco man-in-the-middle;

TLS Attack Possono essere rilevati nel traffico TLS diverse deviazioni del tempo di sistema. Un attaccante può modificare il tempo di sistema del destinatario attraverso NTP e rintracciare le connessioni TLS dalla rete anonima. È un attacco molto esoso.

Riferimenti bibliografici

[1] M. Ranum, *Network Forensics and Traffic Monitoring*, Computer Security Journal, Vol. XII, 1997

[2] http://www.xplico.org

[3] http://www.edecision4u.com/E-DETECTIVE.html

[4] http://networkminer.wiki.sourceforge.net/NetworkMiner

[5] http://it.wikipedia.org/wiki/Anonimato

[6] http://www.torproject.org

[7] http://www.privoxy.org

[8] http://punto-informatico.it/1641461/PI/News
/germania-crackdown-contro-rete-tor.aspx