WASTE

ett distribuerat och krypterat kompisnätverk

Innehåll

1 Inledning

Eftersom vi nyligen läst en kurs i kryptoteknik så tyckte vi det skulle vara intressant att göra ett arbete om ett krypterat, distribuerat system som WASTE.

2 Bakgrund

WASTE är ett skyddat kompisnätverk bestående av en mängd noder som kommunicerar över krypterade kanaler. Ansluter man till en person i ett nätverk och blir accepterad som medlem så får man också reda på alla andra som är med i nätverket, och kan ansluta till dem.

Protokollet, och dess första implementation, skapades av Justin Frankel på Nullsoft år 2003. Bara några timmar efter att programmet lades upp på Nullsofts hemsida så togs det bort igen av AOL, med ett meddelande om att programmet inte var menat att läckas till allmänheten, och att alla nedladdade kopior måste förstöras.

Vad som skiljer WASTE från andra krypterade nätverk är att det inte har någon central server, utan använder vanliga klienter för att distribuera information om nätets topologi. Avsaknaden av en central server innebär att nätverket inte kan förstöras genom att ta ut en enskild nod.

3 Protokollet

Varje nod i systemet kan vara både server och klient samtidigt. Med server menas en dator med förmågan att ta emot inkommande anslutningar. Trafik som går mellan två noder som ej har direktanslutning till varandra, måste routas genom de noder som arbetar som servrar. Nätverket byggs upp av noder som har en, eller helst flera, anslutningar till andra noder, ju fler anslutningar desto snabbare och stabilare blir nätet.

För att ansluta till ett nätverk räcker det att man ansluter till vilken nod som helst, som redan är med i nätverket och agerar som server. Man kan också ansluta till nätverket genom att någon som redan är ansluten skapar en anslutning till en.

Genom att två noder ur separata nätverk ansluter till varandra skapas ett nytt, större nätverk där alla noder ur de ursprungliga nätverken är med. Vill man förhindra att sånt händer kan man enkelt ställa in ett namn på sitt nätverk, och då tillåts bara klienter med samma nätverksnamn att ansluta.

3.1 Nya anslutningar

WASTE använder sig av ett eget icke-standardprotokoll som utnyttjar RSA och BlowFish.

Alla WASTE-användare har RSA-nycklar som är 1024 bitar eller längre. När två användare försöker ansluta till varandra så utbyter de SHA-1-strängar av sina publika RSA-nycklar. Om någon av parterna ej känner igen hashsträngen sedan tidigare så stängs anslutningen.

Båda parterna slumpar sedan fram varsin delnyckel som krypteras med den andra partens publika nyckel innan de utbyts. Den slutgiltiga sessionsnycklen

är en kombination av dessa två delnycklar. På så vis bidrar båda parter till att skapa en säker sessionsnyckel.

När den slutgiltiga sessionsnyckeln är överförd så krypteras all fortsatt kommunikation med BlowFish/PCBC.

3.2 Ny användare

När en nod, som är frånkopplad från P2P-nätverket, ansluter till en annan nod som är medlem i P2P-nätverket så får den anslutande noden information om alla andra noders existens. Den anslutande noden kan sedan skapa ytterligare anslutningar till andra noder för att stärka dess koppling till P2P-nätverket.

Om man vill kommunicera med en nod som man inte har direktanslutning till kan man antingen försöka skapa en ny anslutning, eller så kan man låta andra noder routa trafiken. Det är upp till varje nod om de tillåter routing eller inte. Noder med låg bandbredd bör ej tillåta routing, för att förhindra att hela bandbredden går åt till att routa andras trafik.

3.3 Tillhandahållna tjänster

WASTE tillhandahåller uppladdning och nerladdning av filer, samt både privat och gemensam chat. WASTE tillhandahåller också sökfunktioner för att göra en global sökning efter en viss fil. På en lägre nivå kan man säga att det finns två huvudtyper av paket. Routade paket och broadcastade paket. Routade paket är vanligvis svar på broadcastade paket.

3.4 Filöverföringar

En användare kan båda söka efter filer i hela nätverket, och lista en enskild användares filer. För att ladda hem en fil så skickas ett “File Request”-paket. Begäran kan antingen vara för en hel fil eller för en del av en fil. Det är sedan upp till implementationen om den stödjer funktioner som tillåter en att pausa och fortsätta ladda på en fil, eller ladda samma fil från flera användare samtidigt (“multisource download”).

För att ladda upp en fil till någon så skickas först ett “Upload”-paket till noden man vill ladda upp något till. Om mottagarnoden accepterar detta så påbörjar den en vanlig nerladdning enligt texten ovan. Enda skillnaden från en vanlig nerladdning är att initiativet har tagits av den nod som filen i fråga ligger på.

3.5 Chat

Chat kan ske i publika eller gömda chatkanaler. Chatfunktionen i WASTE påminner väldigt mycket om den i IRC. Man namnger också kanaler på samma sätt som man gör på IRC, t ex “#chatkanal”. För att skapa en gömd kanal med samma anger man istället namnet “&chatkanal”.

Man kan också kommunicera privat med enskild person precis som man kan göra på IRC.

3.6 Distribution av nycklar

När en ny användare ansluter till nätverket känner han förmodligen bara till ett litet fåtal av de publika nycklar som behöver användas.

För att få reda på den information han vill ha, skickar han ett speciellt "Keydist"-paket – med sin egen publika nyckel och sitt användarnamn – till en person han känner till. Grannen skickar i sin tur vidare paketet till sina grannar och så vidare, tills alla har fått meddeladet. När en nod får ett "Keydist"-paket så svarar det med sin egen publika nyckel, och på så sätt får alla tillgång till krypteringsinformationen.

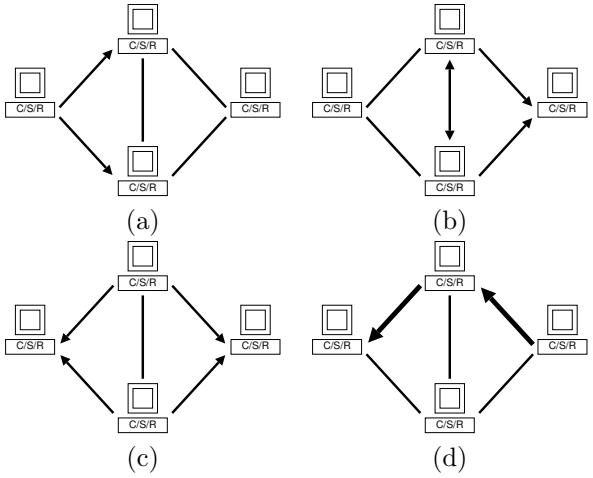
3.7 Routing

Routing är användbart av två anledningar; den uppenbara är att vissa noder inte har möjlighet att lyssna på externa portar och därför måste kommunicera med någon som kan agera server, och den mindre uppenbara är att man kan vilja dölja vem det är man pratar med, från personer utanför nätverket.

Genom att be WASTE att alltid kommunicera genom en routad anslutning så gör man det svårare för eventuella sniffare att upptäcka exakt vem det är man pratar med.

För att kunna routa trafik måste en server först skapa sig en routing-tabell som innehåller information om hur terrängen ser ut, så servern vet vilken av sina grannar den behöver prata med för att komma i kontakt med en specifik klient.

För att hitta en väg mellan två noder måste nod A skicka ett broadcast-paket med nod B som adressat. Den första noden som tar emot paketet från A kommer i sin tur skicka vidare det till alla andra noder som den känner till. Alla dessa gör likadant, men undviker givetvis att skicka tillbaks paketet till noden de först fick det ifrån.



Figur 1: Topologiundersökning

När nod B slutligen får paketet så vet den att alla noder den får det från är en

möjlig routing-väg till nod A. När data ska skickas så kan klienterna godtyckligt välja vilken av de möjliga vägarna de ska utnyttja.

Figur 1 visar hur en dylik transaktion skulle kunna gå till. Nod A, längst till vänster, skickar ett broadcast-paket till nod B, längst till höger (a). Noderna i mitten får paketet från A, och skickar genast vidare till sina grannar. De skickar dels till nod B, men också till varandra (b). Paketen de får från varandra kastas helt enkelt bort som dubletter.

Nod B får paket från båda mittnoderna, och vet då att den kan skicka paket till nod A genom dem (c). När B väl ska skicka data så väljer den valfri av sina möjliga vägar, och börjar skicka (d).

En risk med routing-protokoll är att det uppstår slingor där paket bara skickas runt i all evighet. IP-protokollet löser detta genom ett livslängdsfält (TTL), och WASTE löser det genom att varje server kommer ihåg ett identifikationsfält för varje paket som skickats till det, och jämför med nya paket. Ifall en dublett hittas så skickas det nya paketet helt enkelt inte vidare.

Detta är ett uppenbart skalningsproblem, eftersom det inte går att lagra tillräckligt många identifikationsfält när antalet noder ökar och mängden trafik blir för hög.

4 Användergränssnittet

Användargränssnittet till programmet, som går att ladda hem från sourceforge, är tyvärr inte speciellt användarvändligt. Inställningarna är många, ologiskt sorterade och krångliga att komma åt. Det är uppenbart att programmet inte var menat att släppas till allmän användning.

Gränsnittet består av en mängd enskilda fönster, vilket gör det svårt att hitta rätt del av programmet när man växlar mellan WASTE och andra applikationer. Enligt oss hade det varit enklare med ett gränssnitt liknande det som DC++ använder, där allting samlas i ett enhetligt grundfönster och kan navigeras mellan med hjälp av flikar.

Vi har hittills inte lyckats hitta något grafiskt, praktiskt användbart alternativ till orginalprogrammet, men Mattias Ek, Fredrik Hultin och Jan Lindblom vid Luleå Tekniska Universitet arbetar på en plattformsoberoende Java-version. Denna finns för tillfället bara som mycket begränsad server-version för routning och filutdelning.

5 Hur vi hade gjort

Om vi skulle utveckla ett eget protokoll för ett säkert kompisnätverk så hade vi använt oss av end-to-end-kryptering för chat, filrequester och filer. Som WASTE är konstruerat i dagsläget, kan alla datorer inom nätverket som routar trafik också läsa den routade trafiken. Detta kan tyckas likgiltigt, men om en enda orättfärdig person kommer in i nätverket är det fullständigt öppet för honom.