Medical Records & Secure Connections:
Projektarbete i kursen Datorsäkerhet (EIT060)

EIT, Institutionen för Elektro och
Informationsteknik, Lunds Tekniska Högskola
Martin Hell

Adam Hansson Lyrén, C11 (dic11aha@student.lu.se)
Johan Bäversjö, C11 (dic11jba@student.lu.se)
Mergim Rama, C11 (dic11mra@student.lu.se)
Sven Elfgren, C11 (dic11sel@student.lu.se)

28 februari 2013

LUNDS
UNIVERSITET
Lunds Tekniska Högskola

Innehåll

1 Introduktion

Projektet handlar till stor del om hur man kan upprätta en säker och krypterad anslutning mellan klient och server. För att kunna upprätta denna anslutning kommer vi använda oss av Secure Socket Layer (SSL) protokollet. Djupare teknisk beskrivning av SSL kommer senare i rapporten. Detta projekt kräver att man skapar en produkt som vi kan sälja vidare till myndigheter vilket visar projektets storlek. Koden till programmet kommer inte finnas med i rapporten. När vi använder oss av SSL protokollet kommer vi även använda oss av Java Secure Socket Extension (JSSE). Efter att ha skrivit klart programmet ska det säkerhetsutvärderas. Denna säkerthetsutvärdering täcker alla olika attacker som kan uppstå mot systemet och vilka attacker systemet kan skydda sig mot.

2 Bakgrund

Problemet som projektet grundar sig på är att skapa säkra medicinska journaler för ett sjukhus. Med säkra journaler menas i detta fall journaler som är sekretessbelagda så att inte vem som helst kan ha tillgång till dem, det vill säga att patienternas integritet skyddas. Eftersom systemet är en produkt framtagen för riktig användning måste det begränsas så att olika användargrupper på sjukhuset har olika behörigheter till journalerna. Personer som inte har någon behörighet kommer aldrig att kunna nå journalerna. De olika grupper som ska finnas med är:

- Socialstyrelsen
- Läkare
- Sköterska
- Patienter

2.1 Socialstyrelsen

Socialstyrelsen är det organ som bestämmer, det vill säga de har befogenheter till att kunna ta bort samt lägga till journaler.

2.2 Läkare

Läkaren kan däremot skriva och läsa färdiga journaler, förutsatt att han är den som tar hand om patienten. Han kan också läsa journaler som som finns på hans avdelning. Om patienten inte har någon journal har läkaren befogenheten till att skapa journalen. Kravet är då att läkaren själv är den som behandlar patienten.

2.3 Sköterska

Sköterskans rättigheter är att hon kan läsa och skriva till journalerna som hon är skriven på och att läsa alla journaler från hennes avdelning.

2.4 Patienter

Patienterna är berättigade till att läsa endast sina egna journaler vilket är en självklarhet. För att kunna spara allt som händer på journalerna ska en logg föras som visar vem som gjort vad i journalen samt vid vilken tidpunkt det hände.

3 SSL/ TLS

SSL är ett väldigt väl använt protokoll som man finner på nätet när en säker anslutning krävs. Med SSL förhindrar man tjuvlyssning på nätverket, vilket är en viktig faktor som gör protokollet säkert.

Protokollet befinner sig mellan de två översta lagren i TCP/IP modellen, transportlagret och applikationslagret. Eftersom protkollet används för säkra anslutningar på nätet faller det i sin natur att SSL används mestadels i Hypertext Transfer Protocol (HTTP) [första referensen: http://docs.oracle.com/javase/7/docs/technotes/guides/security

Många andra applikationer kan dra nytta av SSLs säkra anslutningar, några av dem är Telnet och FTP. SSL protokollet har två lager av protokoll vilket har olika funktioner [inludera en bild på det].

3.1 Record Layer

I Record Layer ser protokollet till att meddelandena får integritetsskydd. För att göra detta så använder sig SSL av symmetriska nycklar och digitala signaturer. Det som utförs är att det beräknas en MAC för packetet som ska skickas med hjälp av HMAC funktionen. HMAC funktionen är till för att kunna kontrollera datan om den är säker över en opålitlig länk. Parterna som använder sig av HMAC delar på en nyckel för att kunna utföra samma säkerthetskontroll. HMACen används med en hashfunktion som till exempel MD5, och därav ett h framför MAC. MACen kommer tillsammans med datan krypteras och vara datapacketet av Record protokollet. Utöver detta kommer det finnas en header som innehåller information om hur en 'handshake' går till, vilken version av SSL/ TLS som används samt hur mycket data det finns i paketet.

3.2 Upper Layer Protocols

I Upper Layer Protocols är det flera steg som utförs. Här kommer protokollet tilldela nycklar och autentisera klienter respektive servrar med hjälp av assymetriska nycklar. När protokollet gör detta betyder det att protokollet startar sin 'handshake' metod. Utöver själva autentiseringen av server till klient så tillhandager 'handshake' också:

1. Fastställa vilka algoritmer som ska användas
2. Förhandla om vilka nycklar krypteringen skall använda samt vilken MAC som skall användas
3. Autentisera klienten till servern, vilket inte är nödvändigt

Eftersom denna 'handshake' kan bestå av delar eller hela meddelandecyklar, beskriver vi hur vårt system använder sig av dessa meddelanden i vår 'handshake' metod.

3.3 Handshake

4 Certifikat

Ett certifikat är ett elektroniskt dokument som som änvänds för att kunna bekräfta någons identitet på nätet. Certifikatet har en publik nyckel (publik) som man binder till en användares namn, address m.m. Utöver detta finns det information om utfärdaren av certifikatet. För att kunna använda certifikat behövs det ett "överhuvudsom alla ska kunna lita på. Dessa överhuvuden är CAs, Certificate Authorities.

En CA utfärdar certifikat som andra kan använda för att visa att de är pålitliga på nätet. För att två parter ska kunna lita på varandra måste använda sig av certifikat som är utfärdade av samma CA. Idag finns det enstaka stora organ som tillhandahåller certifikat till nästan alla olika tjänster, dessa är VeriSign Inc och Comodo [referens wiki: `http://en.wikipedia.org/wiki/Certificate_authority`]

För att kunna ge ett bra exempel på detta så tänker vi oss en webbläsare som väldigt många använder. Webbläsaren måste ha ett certifikat utfärdat av samma myndighet som hemsidan har för att man ska kunna lita på att sidan utan att ha besökt den tidigare. Hemsidans certifikat kan bestå av en certifikatkedja, vilket betyder att man skapat en koppling mellan sitt eget certifikat och CA certifikatet [Boken sida 291]. Skulle det inte vara så och sidan saknar certifikatet kommer användaren bli varnad eftersom man inte kan uppge identiteten för hemsidan.

4.1 Autentisering

För att en användare ska kunna ansluta sig till en hemsida måste de autentisera sig, det vill säga kontrollera om hemsidan har ett utfärdat och giltigt certifikat. Hemsidan kommer då skicka sin publika nyckel till användaren som i sin tur skickar en förfrågan till CA:n och frågar om denna nyckel tillhör sidan som skickade den. Skulle den göra det så kommer en uppkoppling ske och informationsbyte blir möjlig. Det händer också att falskasidor tar denna nyckeln och påstår att denna är deras. För att undkomma detta problem krypteras informationen som kräver den rätta privata nyckeln för att dekryptera informationen. För att kunna skapa och hantera certifikaten rätt har man tagit fram standarder för detta. En standard som används väldigt väl och som vi använt till vårt projekt är X.509.

4.2 SSL Certifikat

Eftersom vi använder oss av SSL i vår nätverksanslutning mellan klient och server så kommer det finnas inbakade SSL certifikat som SSL kommer använda vid 'handshake' fasen. Det som händer då är att SSL certifikatet innehar krypteringsnycklarna som ska säkra uppkopplingen, som respektive nod skickar till sin uppkopplingsnod [referens: `http://msdn.microsoft.com/en-us/library/windows/desktop/aa364691%28v=vs.85%29.aspx`. På så sätt delar de krypteringsnycklar i en 'handshake' mellan varanda. Värt att notera är att servern skickar sitt certifikat till klienten men att klienten ska skicka sitt för att autentisera sig är frivilligt.

4.3 Certifikat - Projekt

För att vårt projekt ska fungera måste vi skapa egna certifikat för att kunna autentisera klienterna och på så sätt skapa en två faktor autentisering.

Det vi började med var att skapa ett CA certifikat som är av typen X.509. Detta certifikat använder vi sedan för att skriva på klienteras och serverns egna certifikat. Certifikaten vi skapar är:

- Läkar - Certifikat
- Sköterska - Certifikat
- Patient - Certifikat
- Sjukhus - Certifikat

Varje certifikat kommer ha sin egen keystore där certifikatkedjan mellan användare och CA finns. Keystoren kommer vara lösenordsskyddad. Detta lösenord är personligt och varje användare måste knappa in detta för att kunna autentisera sig mot servern och då kunna koppla upp sig mot servern och nå journalerna.

5 Säkerhetsutvärdering

Säkerthetsutvärdering av systemet.

6 Referenser

Referenser till olika tekniska begrepp som kommer att nämnas.