Implementation av stark autentisering till en webbportal

Daniel Borgstén

Daniel Borgstén

EXAMENSARBETE	
Arcada	
Utbildningsprogram:	Informationsteknik
Identifikationsnummer:	
Författare:	Daniel Borgstén
Arbetets namn:	Implementation av stark autentisering till en webbportal
Handledare (Arcada):	Dennis Biström
Uppdragsgivare:	CSIT Finland Oy

Sammandrag:
Detta examensarbete handlar om olika autentiseringsmetoder på webben som används nuförtiden för att autentisera användaren på olika plattformar och servis. Jag valde detta arbete som mitt examensarbete eftersom vi var i behov av en stark autentiseringsportal på min arbetsplats. Examensarbetet ger läsaren en överblick av hur autentiseringsmetoderna fungerar och i vilka fall de används. Arbetet belyser också läsaren över vilka skillnader stark autentisering och den konventionella autentiseringen har. Syftet med arbetet var att undersöka dessa olika metoder som autentisering kan nuförtiden verkställas och välja det bästa alternativet för CSIT Finland Oys identitetshanteringsprogram. Målet med detta examensarbete var att programmera en stark autentiseringsportal med hjälp av SAML teknologin. Arbetet handlar om hur man konfigurerar en SAML miljö med hjälp av Shibboleth och vilka andra krav autentisering behöver för att kunna kallas stark autentisering.

Nyckelord:	Stark autentisering, SAML, suomi.fi, CSIT Finland Oy
Sidantal:	26
Språk:	Svenska
Datum för godkännande:	

DEGREE THESIS	
Arcada	
Degree Programme:	Information technology
Identification number:	
Author:	Daniel Borgstén
Title:	Implementation of strong authentication to a portal
Supervisor (Arcada):	Dennis Biström
Commissioned by:	CSIT Finland Oy

Abstract:
This thesis compares some of the most used authentication methods used in the internet by modern services and platforms. The thesis is giving a brief look at each one of them, how they work, in which cases are they used and each methods pros and cons. The reason for this thesis was to research different authentication methods that could be used in CSIT Finland Oys identity management software. The goal of this thesis was to implement a strong authentication solution to their existing software. The method chosen was SAML with the help of Suomi.fi. This thesis demonstrates how strong authentication is configured using SAML with the help of Shibboleth and other requirements that are used in the process.

Keywords:	Strong authentication, SAML, Suomi.fi, CSIT Finland Oy
Number of pages:	26
Language:	Swedish
Date of acceptance:	

INNEHÅLL / CONTENTS

1 INLEDNING 7

1.1 Bakgrund 8
1.2 Syfte och mål 8

2 AUTENTISERING 9

2.1 Autentiseringens historia 9
2.2 Autentisering nuförtiden 9
2.3 Autentisering och auktorisering 12
2.4 Stark autentisering 13
2.5 Jämförelse av information 13

3 SUOMI.FI 14

3.1 Suomi.fi tjänstens starkheter 14
3.2 Autentiseringsbegäran 14
3.3 Autentisering via Suomi.fi 15
3.4 Suomi.fi krav 15

4 SAML 16

4.1 SAMLS historia 16

5 IMPLEMENTATION 16

5.1 Shibboleth 17
5.2 Apache HTTP 17
5.3 Shibboleths konfigurering 17
5.4 Apache HTTP servers konfigurering 22
5.5 Implementationens slutsats 22

6 SLUTSATS OCH DISKUSSION 23

7 Källor / References 24

8 Bilagor / Appendices 26

Figurer

Figur 1. ApplicationOverride konfigurering 18
Figur 2. Session konfigurering 18
Figur 3. Logout konfigurering 19
Figur 4. SessionInitiator konfigurering 19
Figur 5. AssertConsumerService konfigurering 20
Figur 6 Handler konfigurering 21
Figur 7. Error konfigurering 21
Figur 8. MetadataProvider konfigurering 21
Figur 9. AttributeExtractor konfigurering 22
Figur 10. CredentialResolver konfigurering 22
Figur 11. Apache httpd.conf konfigurering 23

Terminologi och förkortningar

SSN: Social Security number, USA:s motsvarighet till personnummer
OAUTH: Open Authorization, en standard för användning av inloggnings-information från en internetbaserad tjänst
SAML: Security Assertion Markup Language, en XML-baserad öppen standard för att utbyta autentisering
XML: Extensible Markup Language. ett universellt och utbyggbart märkspråk
ID-FF: Identity Federation Framework
JSON: JavaScript Object Notation ett kompakt textbaserat format som används för att utbyta data
2FA: Two-factor Autentication, en enkel säkerhetsåtgärd vars syfte är att förhindra otillåten tillgång till användarkonton
IP address: Internet Protocol address, ett nummer som används som adress på Internet
SSO: Single Sign-On, en metod inom sammansatta datasystem för att hantera användare med aspekt på användarbehörighet och användarverifiering, så att dessa användare endast behöver logga in en enda gång för att nå olika system
IdP: Identity Provider, möjliggör identifiering av en person
SP: Service Provider, möjliggör servis till en specifik tjänst
VPN: Virtual Private Network, ett säkerhetsprogram som skyddar användares internetanslutning
URL: Uniform Resource Locator, används för att ange adresser på World Wide Web
E-tjänst: Tjänster som produceras och konsumeras i ett elektroniskt medium
HTTP: Hypertext Transfer Protocol, ett kommunikationsprotokoll som används vid överföring av webbsidor på informationsnätverket på Internet
GDPR: General Data Protection Regulation, ett regelverk som EU utarbetat och som börjat gälla 25:e maj 2018

1 INLEDNING

Nuförtiden lagrar företag all information de kan få över användaren på olika tjänster och på social media för att kunna maximera försäljning. Detta har lett till att en lag (GDPR) tagits i bruk inom EU som hjälper skydda användarna och ger användaren rätt att framkalla den information som företag lagrat över dem. I sådana fall måste de olika tjänster och servis förse användaren med rätt användares information. Detta problem har många företag och SP löst med någon slags av stark autentiseringmetod som autentiserar användaren. SAML är en av teknologierna som möjliggör företag att utöka deras tjänster med hjälp av att autentisera användaren utanför sitt interna nätverk.

Idén för examensarbetet fick jag från min arbetsplats där vi var i behov av en autentiseringsportal som skulle använda sig av stark autentisering och mitt intresse för att utveckla en sådan uppstod. Jag skall utforska de olika metoder stark autentisering används på nuförtiden och implementera en stark autentiseringslösning till vår nuvarande portal. För att kunna göra detta kommer jag att forska vilket sätt är det säkraste från SP synvinkel och implementera portalen till CSIT identitetshanterings program.
Arbetet kommer inte att visa steg för steg hur implementationen gjordes och programmerades, utan berättar mera om vilka val gjordes, vilken autentiseringsmetod valdes, vilka skillnader det fanns mellan metoderna och argument varför de val som valdes blev valda. Eftersom de specifikationer jag fått från arbetsgivarens sida är så generella kan det också argumenteras att det inte finns ett rätt sätt att bygga upp autentiseringen till portalen. Därför kommer arbetet mera handla om den teoretiska bakgrunden bakom stark autentisering.

1.1 Bakgrund

Nuförtiden finns det många olika slags metoder som hjälper e-tjänster med att känna igen sina användaren. Oftast finns det metoder för varje e-tjänst beroende på hur tjänsten är uppbyggd och hur mycket pålitlig information tjänsten behöver av användaren. Det finns metoder som endast auktoriserar eller autentiserar användaren. Sedan finns det starkare metoder som gör båda två. I vissa fall kan det också ha en stor skillnad från vilken källa användaren blir autentiserad från.

Idén för projektet fick jag från min gamla arbetsplats där vi skulle komma på ett sätt att autentisera användarna som loggat in till våra sidor. Det var meningen att användarna som inte än hade ett konto till våra sidor, kunde snabbt och smärtfritt autentiseras och skapa ett konto. Efter autentiseringen skulle användaren också auktoriseras inom identitetshandlings programmet enligt de rättigheter användaren fått. Det var också mycket viktigt att auktoriseringsmetoden kunde autentisera användaren från en pålitlig källa. Med hjälp av att använda auktorisering och autentiserings tjänster kan man bekräfta att användaren loggar in till rätt konto.

1.2 Syfte och mål

Syftet med detta arbete är att ge en uppfattning till läsaren över vad skiljer autentisering och auktorisering, vad stark autentisering och användarsäkerhet innebär, hur användarsäkerhet utvecklats inom åren och vilka olika metoder det finns för autentisering och auktorisering.
Målet är att undersöka olika slags metoder som kan användas till att bygga upp en stark autentiseringsportal, belysa läsaren varför stark autentisering är ett säkrare sätt att logga in en användare än det konventionella sättet och hurdana sätt det finns att använda sig av stark autentisering.

2 AUTENTISERING

Med autentisering menar man oftast bevis på personens identitet till exempel vid inloggning till en webbsida. Autentisering kan dock också användas i fall där man vill till exempel verifiera att ett meddelande inte blivit ändrat på sedan den lämnat avsändaren. I detta arbete skall jag implementera stark autentisering till en portal och därför koncentrerar jag mig mera på hur man kan säkerställa användares identitet på, än hur meddelanden blir autentiserade.

2.1 Autentiseringens historia

Redan länge har det funnits många slags metoder att autentisera människor på. Ett av de mest vanligaste sätten att autentisera en person man känner till är via röst eller ansikte. Då en person går in till ett låst rum bekräftar personen egentligen sin rättighet att gå in eftersom personen blivit berättigad en nyckel (Miracl 2019). De vanligaste sätten dock att autentisera en person är via deras identifieringskort, körkort eller pass. Men nuförtiden har folk börjat använda sig mera av digitala tjänster som har också skapat behov av att autentisera personer på digitala plattformar. Detta har lätt till att man nuförtiden kan autentisera sig med t.o.m. sitt eget finger då man loggar in till sitt bankkonto via sin telefon. Bouljoub spekulerar att mycket snart kommer de olika digitala metoderna att möjliggöra resandet utomlands utan ett fysiskt pass eller identitetsbevis eftersom identifieringen kan ske med hjälp av biometrisk autentisering (Bouljoub 2019).

2.2 Autentisering nuförtiden

Det är typiskt att en normal webbsida kräver bara ett användarnamn och lösenord då en användare försöker loggar in. Men detta leder ofta till att användaren har ett mycket simpelt lösenord eller att användaren använder samma lösenord för många olika tjänster (CNN Business 2019).

För att hindra en användares konto falla i fel händer har det skapats många teknologier till företag och den privata sektorn som underlättar användares situation med logga in till tjänster.

Nuförtiden finns det tre olika synsätt att se på autentisering. De tre synsätt förlitar på att användaren 1) vet, 2) har, eller 3) är något. Lösenord är ett exempel på information som användaren vet. Ett identifieringskort är ett fysiskt föremål användaren har. Ett fingeravtryck är någonting användaren själv är, dvs användaren behöver ingen extra information eller fysiska föremål för att kunna autentiseras (Forbes 2017).

En av de mest användbara autentiseringsmetoder som används är 2FA. 2FA ombesörjer en högre nivå av säkerhet eftersom metoden kräver användares lösenord och ett annat skilt fysiskt föremål innan användaren kan logga in. Det kan tänkas att då någon lyfter ut pengar ur en bankautomat behöver personen både sitt PIN och det fysiska kortet för att lyckas. Inom datateknik följer 2FA samma principer. Användaren behöver sitt användarnamn och ett lösenord men dessutom behöver de en engångskod. Förritiden kunde användaren få koden från en speciell fysisk maskin. Nuförtiden är det mycket normalt att koden skickas i stället till användares telefon via SMS. Om alla tjänster skulle erbjuda 2FA kunde användaren ha mindre bekymmer över att använda samma lösenord till många tjänster, eftersom det inte skulle vara det enda som behövs för att logga in till kontot. 2FA är bara en av de många olika autentiseringsmetoder för att göra en användares liv tryggare.

Alternativt kan användaren oftast också skapa ett konto till sidor via en tjänst där användaren redan har ett konto. Då blir användaren auktoriserad av tjänsten som till exempel kan vara Facebook eller Google och sida vart användaren skulle skapa konto accepterar användaren. Efter detta behöver användaren i fortsättningen bara logga in via den tjänst som har valts då kontot först skapades. Detta leder till att användaren har mindre lösenord att bekymra sig över. Problemet dock med att ha ett konto som styr över de andra konton är om användaren tappar rättigheterna för kontot. I synnerhet om det kontot som styr de andra inte autentiserar användaren från en extremt pålitlig källa, som exempel folkbokföringen. Medan vem som helst kan göra ett Facebook konto och påstå sig vara någon de inte är kan inte samma göras om tjänsten autentiserar användaren vid registreringstillfället från folkbokföringen.

Många produkter och service har också tagit i bruk biometrisk autentisering. Biometrisk står för att "mäta liv" och förlitar på användares karaktärsfasta identifiering. De vanligaste identifieringsmetoderna är via röst, fingeravtryck och ansiktsigenkänning. Biometriska identifieringen har också fördelen med att användaren inte behöver ha någon specifik maskin, kod eller minnas deras lösenord för att kunna identifieras. Eftersom ingen människa har exakt samma röst, fingeravtryck eller ansikte påstås biometriska systemet vara väldigt pålitligt. Trots detta var exempel Apples populära fingeravtryckidentifiering, Touch ID hackat inom 24 timmar efter teknologins utsläpp (Forbes 2019).

Affärautentisering är en identifieringsmetod som de flesta har i bruk omedvetet. Teknologin används ofta i sammanband med email och bankservice. Affärautentisering använder sig av information som den redan tidigare fått av användaren för att kunna autentisera användare vid inloggnings skedet. SP kan använda sig av information användaren gett vid registreringsskedet eller data över användaren som sparas då användaren loggat in. SP kan jämföra om den nya data och den sparade data har likheter, dvs analysera om användaren är den samma eller någon som poserar den riktiga användares användarnamn och lösenord. Ett mycket vanligt sätt SP utför affärautentisering är med att spara information över från vilket land användaren normaltvis loggat in. Om detta bryter mot det normala använder tjänsten sig ofta av 2FA för att skicka en engångskod till användaren för ett extra lager av säkerhetsskydd. Medan affärautentisering låter bra, har det också sina nackdelar. Metoden har inget sätt att kontrollera har till exempel användaren farit utomlands på resa eller förfalska användaren sin plats med hjälp av en VPN.

Datoridentifierings autentisering är mycket lik affärautentisering med tanke på att metoden också jämför information från användares första inloggning med användares framtida inloggningar. Datoridentifierings autentisering verifierar att användaren använder samma apparat för inloggningen som tidigare använts. Detta kan metoden göra med hjälp av att installera en mjukvara då användaren loggar in första gången med en ny apparat. Mjukvaran innehåller en krypteringsmodul som systemet kollar att ännu finns kvar på apparaten varje gång användaren försöker loggar in. Lik affärautentisering är datoridentifierings autentiseringen också ett mycket osynligt sätt att autentisera användaren på från använ-

dares perspektiv. Nackdelar med datoridentifierings autentisering är då användaren försöker logga in till sitt konto från en ny apparat och då måste användaren autentiseras på ett annat sätt före en ny installation av krypteringsmodulen kan ske.

I företag och den industriella världen är SSO en mycket vanlig autentiseringsmetod. Med hjälp av SSO kan användaren få tillgång till många olika program med samma konto efter att användaren loggat in till en av de programvarorna som är anknutet till samma SSO paket. Detta är en mycket smärtfri metod från arbetsgivarens perspektiv att ge nya anställda rättigheter för olika program i ett större företag. Med metodens lätthet att logga in användaren till många konton med bara ett användarnamn och lösenord ingår det också större risk för missbruk angående metoden. Om någon obehörig åtkommer användares konto inom SSO paketet, kommer den åt alla program som hörs till samma SSO paket.

2.3 Autentisering och auktorisering

Fastän autentisering och auktorisering låter lika och ofta tas upp i likadana sammanhang är de två olika saker. Auktorisering handlar om vilka rättigheter en användare har och vad användaren får göra i ett system. Autentisering handlar om systemet kan identifiera användare. Det finns en stor skillnad mellan teknologierna och hur användaren kan antigen bli auktoriserad eller autentiserad vid inloggningen till en sida.

Saml och Oauth är nuförtiden några av de allra vanligaste teknologierna som webbsidor använder för att logga in användaren. Båda metoderna har egna användningsfall beroende på vad tjänsten beslutat göra med användaren. Oauth teknologin används ofta då användaren skall auktoriseras. Detta är mycket vanligt att Oauth teknologin används då användaren snabbt vill skapa ett konto till en ny webbsida. Då kan användaren ge tillstånd åt sidan för auktorisering från en annan sida där användaren redan har ett konto och som den nya sidan där användaren försöker skapa konto till litar på. Men protokollet i sig har ingen aning över vem användaren egentligen är. Många tror att Oauth är en autentiseringsteknologi eftersom den mycket ofta också är implementerad in i olika autentiseringsprotokoll, men det är den inte (OAuth 2019). Saml teknologin används för att autentisera

användaren med hjälp av att kontakta en IdP som redan har autentiserat användaren tidigare. Exempel på en mycket vanlig Idp är en bank där användaren har ett existerande konto.

2.4 Stark autentisering

Stark autentisering är en mycket allmän term utan egentligen någon slags ordentlig definiering. Därför blir stark autentisering mycket ofta blandat ihop med att vara 2FA. Däremot är stark autentisering inte nödvändigt 2FA. Enligt den europeiska banken skall stark autentisering kombinera åtminstone två eller fler självständiga faktorer som inte är beroende av varandra. Dessutom skall denna autentiseringsmetod inkludera ett icke återanvändbart element, som kan ej bli stulet eller duplicerat på internetet (Whatis 2015).

2.5 Jämförelse av information

Alla de olika autentiserings och auktoriserings metoderna jämför information användaren gett med information som tjänsten tidigare fått av användaren. Det kan tänkas att informationen användaren ger tjänsten kan vara i olika starkhetsgrader.

Det vanligaste exemplet på inloggning är då användaren loggar in till en tjänst med ett användarnamn och lösenord. Då användaren försöker logga in letar tjänsten om användares användarnamn finns i databasen som är kopplad till tjänsten. Om användarnamnet hittas kollar systemet att lösenordet också stämmer överens och att den är kopplad till användarnamnet. Lösenorden brukas oftast sparas i en krypterad form inne i databasen.
Ett kort lösenord med inga stora bokstäver, numror eller specialtecken tolkas ofta som ett svagt lösenord. Då kan lösenordet tolkas som låg eller svag information och vara på en låg nivå på starkhetsgraden. Medan några av de mera avancerade inloggningsmetoder kräver samma information över användaren som banker dvs användarnamn, lösenord och en engångskod som banken tidigare givit användaren. Detta kan tolkas som ett mycket starkt sätt att logga in användaren eftersom varje inloggning är unikt eftersom engångskoden används. Detta betyder att metoden för inloggningen är endast så stark som den lägsta möjliga acceptabla informationen som tjänsten kräver av användaren.

3 SUOMI.FI

Suomi.fi-autentisering är en autentiseringstjänst för den offentliga sektorn som kan användas för att identifiera slutanvändaren i digitala tjänster. Suomi.fi försörjer organisationer som vill autentisera användaren med information som folkbokföringen har över användaren. Identifieringssystemet jämför information som slutanvändaren gett, med information som redan finns av slutanvändaren i folkbokföringen och autentiserar användaren om informationen stämmer överens. Detta betyder att Suomi.fi är väldigt nyttigt för organisationer och tjänster som vill autentisera användaren vid registreringstillfället. Om slutanvändaren autentiserats via Suomi.fi kan missinformation i tjänstens system också kompletteras med hjälp av den information som redan finns i folkbokföringen. För att en organisation eller tjänst skall kunna ta i bruk Suomi.fi måste de först bli registrerade till Suomi.fi, ha en webbserver konfigurerat enligt SAML standarden och ha en SAML lösning som är enligt Suomi.fi standarden. Suomi.fi erbjuder också en testmiljö utan behov av Suomi.fi registrering (Palveluhallinta Suomi.fi 2019).

3.1 Suomi.fi tjänstens starkheter

Eftersom Suomi.fi knyter ihop information över användares identitet vid registreringstillfället med information som lagrats över användaren i folkbokföringen valdes det som teknologin. Det var ypperligt viktigt för CSIT identitetshanteringssystem att användarna blir autentiserade vid registringstillfället och att de bara kan ha ett konto i systemet. För att få implementeringen gjord skulle en webbserver enligt SAML standarden byggas upp. Webbservern skulle också ha stöd för SAML. För webbservern valdes det Apache HTTP och som SAML lösningen Shibboleth. Valen över metoderna gjordes på basis av att tjänsternas användarvänlighet, säkerhet och kostnad.

3.2 Autentiseringsbegäran

Då tjänsten inte har en aktiv session över en användare som använder tjänsten, skickas en autentiseringsbegäran åt autentiseringstjänsten som SP valt. I mitt exempelfall kommer autentiseringstjänsten vara Suomi.fi. I Autentiseringsbegäran definieras en URL dit SP valt skicka användares information, autentiseringstjänstens URL, en identifikation över

autentiseringsberäran, tidsstämpel på begäran och information över den protokoll som använts.

3.3 Autentisering via Suomi.fi

En användares autentisering anländer då användaren skicka en förfrågan över de angående resurser till tjänsten. Då undersöker SP om användaren redan har rättigheter för handlingen inom sessionen. Om SP inte hittar information över användares rättigheter för handlingen, leds användaren till autentiseringstjänsten dvs Suomi.fi. Den granskar om användaren redan har en aktiv session. Om användaren har en aktiv session, returneras användaren direkt till tjänsten. Medan om användaren inte än är autentiserad och ingen session data hittas leds användaren till för att autentiseras. Efter att användaren autentiserats, leds användaren tillbaka till autentiseringstjänster (Suomi.fi) som skapar sessionen för den autentiserade användaren. Sedan leds användaren tillbaka till tjänsten med sessionsdata. SAML 2.0 teknologin används för hopknytning då sessions data överförs från autentiseringstjänsten till tjänsten.

3.4 Suomi.fi krav

För att SP skall kunna använda Suomi.fi autentisering måste tjänsten först ansöka efter nödvändiga uppgiftstillstånd och skicka deras tjänsts metadata åt IdP. Metadata skall skickas som en standardiserad SAML 2.0 XML fil. En del av XML filen skall fyllas upp med information och attribut över tjänsten. Medan resten av fälten i XML filen skall fyllas upp med information som IdP kräver att finns med. Suomi.fi försörjer information över hur fälten och attribut skall fyllas i XML filen. Efter att organisationen blivit accepterad att använda Suomi.fi autentisering, blir XML filen uppladdad till IdP tjänst (Esuomi 2018).

Tjänstens uppgiftstillstånd kan variera mellan stark, medel och svag. SP får information över användaren från IdP beroende på hur omfattande rättigheter tjänsten fått angående uppgiftstillstånden. De svaga uppgiftstillstånden ger SP tillstånd för användares namn, SSN och användares digitala identifiering. De medelmåttiga uppgiftstillstånden ger ytterligare tillstånd för SP att se användares adress och email adress. De starkaste rättigheterna ger därtill SP tillstånd att se om användaren är en finsk medborgare eller ej.

4 SAML

SAML 2.0 är et öppet SSO protokollstandard som skickar information över användaren, loggningar och olika attribut mellan IdP och SP. SAML transaktionen använder sig av XML standarden för att kommunicera mellan IdP och SP. SAML hopknyter autentisering över användares identitet med auktorisering till att använda en tjänst.

4.1 SAMLS historia

SAML V1.0 var ett projekt som blev utvecklat av OASIS. Syftet med projektet var att ”definiera ett XML system för att utbyta autentisering och autentiserad information”. SAML V1.0 blev färdigutvecklat år 2002 och blev klassificerad som en OASIS standard. Under den tid då första versionen av SAML blivit utvecklat fanns det också ett annat projekt lik SAML som blev utvecklat av Liberty som kallades för ID-FF. SAML och ID-FF hade båda många liknande egenskaper, men den viktigaste egenskapen som ID-FF hade som skilde protokollen från varandra kallades för ”circle of trust”. Circle of trust fungerade som ett identifieringssystem där de som blivit medlemmar måste dokumentera processen hur de identifierat en användare och deras autentiseringsmetod. Sedan måste de andra medlemmarna som var medlemmar i circle of trust undersöka dessa dokument och bestämma om de litar på informationen eller ej. Ett år senare släpptes den nya förbättrade versionen av SAML, som kallades till SAML V1.1. Då bidrog Liberty deras ID-FF project till OASIS för att kunna förbättra SAML även ytterligare. År 2005 släpptes SAML V2.0 ut. SAML V2.0 var en produkt av både SAML V1.1 och ID-FF som slåtts ihop. Sen år 2008 har SAML V2.0 teknologin implementeras på många olika sätt för att autentisera en användare.

5 IMPLEMENTATION

Implementationen utfördes med att konfigurera ett Shibboleth projekt med Apache HTTP servern. Shibboleth konfigurerades via XML filen. Apache servern krävde en mod_shib modul för att konfigureringen mellan dessa två skulle fungera. Eftersom data skickas enligt AJP protokoll standarden, antas det att den mottagande tjänsten kan också hantera ett AJP protokoll.

5.1 Shibboleth

Shibboleth är en mjukvara med öppen källkod. Det betyder att mjukvarans källkod är tillgänglig för modifiering enligt användares behov. Shibboleth består av en SAML lösning som i sig består av tre funktionella komponenter. IdP komponenten som överför autentisering sessioner och anknyter information över användaren till SP för autentisering. SP komponenten vilken är bunden till servern. Webbläsare komponenten fungerar som en klient. (University of Toronto 2012)

Shibboleth konfigureras via en XML fil som IdP och SP använder sig för att utbyta skyddad information. Till denna XML filen tilläggs alla de beskyddade tjänster och enskilda konfigurationer som används.

5.2 Apache HTTP

Apache HTTP server är ett mjukvarsprojekt med hög kvalitets HTTP serverimplementering och hostning för webbsidor. Projektet upprätthålls och uppdateras av en grupp valda programmeraren runt världen som frivilligt är med att utveckla programmet. Dessutom lämnar hundratals användaren och programmeraren in idéer och föreslag på förbättringar angående projektet (Apache 2020).

5.3 Shibboleths konfigurering

Implementationen började med att konfigurera Shibboleths XML fil. I XML filen definieras det information som krävs för att konfigurationen skall lyckas.

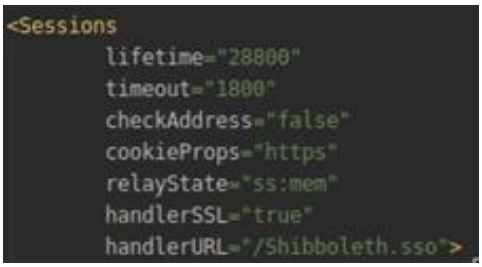
ApplicationOverride beskriver tjänstens information och är ett baselement för alla de kommande fälten.

```
<ApplicationOverride
        id="exempel"
        entityID="https://exempel.fi/shibboleth"
        signing="true"
        encryption="false" attributePrefix="AJP_">
```

Figur 1. ApplicationOverride konfigurering

Id fältet har i detta exempel ingen skillnad, därför har den blivit lämnad tom. Men som exempel skulle tjänstens namn kunna användas som ett id. EntityId fältet beskriver ett unikt Shibboleth URL för tjänsten. Varje tjänst som är uppkopplat till samma Shibboleth projekt måste ha ett unikt EntityiD. Signing fältet kontrollerar avgående meddelandens signatur. Encryption fältet kontrollerar avgående meddelandens kryptering. Krypteringen gömmer informationen under tiden den tid informationen skickas. Till AttributePrefix fältet definieras det hurdant prefix Shibboleth tillsätter information som skickas vidare av användaren till tjänsten. Eftersom AJP protokollet är kompatibelt med Apache servern, valdes det som protokoll. Då Apache servern skickar information vidare skall det i processen avlägsna det definierade prefixet.

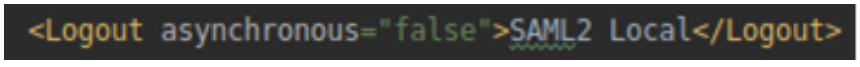
Under Sessions fältet beskrivs identifieringssessionen som syns i bilden nedanför.



Figur 2. Session konfigurering

Lifetime fältet beskriver den maximala tiden på sessionen och tiden mellan begäran. Tiden definieras i sekunder. Fastän sessionen kan vara upp ända till 8 timmar (28 800 sekunder), avbryts sessionen om inga begäran görs på 30 minuter på grund av timeout fältet. I checkAddress fältet användes värdet false eftersom det inte behövs information över användares adress. HandlerURL fältet definierar under vilken stig Shibboleth tjänsten skapas.

I Sessions fältets första barnelement, Logout definieras det hur man loggar ut ur tjänsten.

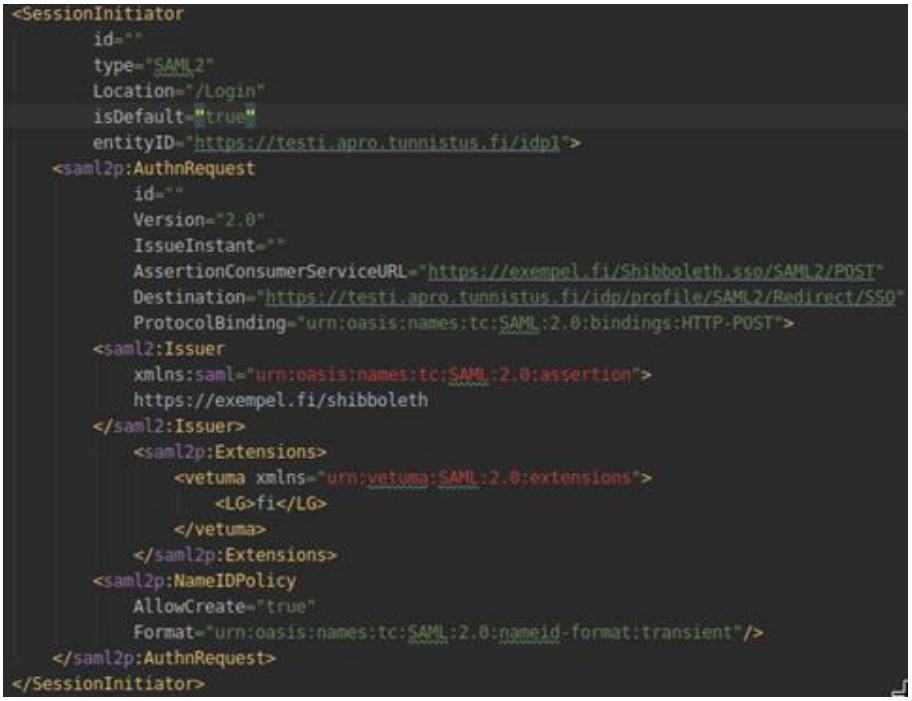


Figur 3. Logout konfigurering

Logout fältet installerar en utloggningstjänst under den definierade stigen. I exemplet ovanför görs en lokal SAML2 utloggning. SP kan börja utloggningsprocessen för användaren med att dirigera om till webbsidans utloggningssida (Shibboleth 2010).

I Session fältets andra barnelement, SessionInitiator definieras det hanterarens information.

Inom SessionInitator definieras ett inre fält, AuthRequest. I AuthRequest beskrivs hurdan den avgående identifieringsbegäran är.



Figur 4. SessionInitiator konfigurering

Id fältet inom SessionInitator är ett valfritt fält där man kan definiera hanterarens namn. Type fältet beskriver till vad protokollet används. Location fältet berättar inom vilken stig inloggningshanteraren existerar. Med att manuellt ta kontakt till adressen kan inloggningen ske. Eftersom tjänsten kan ha många hanteraren måste det definieras ett förinställt värde som används. Det görs med hjälp av att ge isDefault värdet true för tjänsten man vill definiera till det förinställda värdet. EntityId fältet hanterar identifieringstjänstens URL. I exemplet används Suomi.fi testmiljöns URL (Shibboleth 2016).

ID och IssueInstant fylls båda med information då begäran för identifieringen tar för sig. Information som fylls i dessa fält är endast för validering på filen. AssertionConsumerServiceUrl fältet berättar till vilken adress information skickats. Destination fältet berättar till vilket IdP användaren dirigeras till. I exemplet dirigeras användaren till Suomi.fi testmiljöns URL (Esuomi 2018).

Inom Sessions definieras det också AssertionConsumerService fältet. Detta fält definierar hurdana tjänster Shibboleth erbjuder angående identifiering.

```
<md:AssertionConsumerService
        Location="/SAML2/POST"
        index="1"
        Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"/>
<md:AssertionConsumerService
        Location="/SAML2/POST-SimpleSign"
        index="2"
        Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST-SimpleSign"/>
<md:AssertionConsumerService
        Location="/SAML2/Artifact"
        index="3"
        Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Artifact"/>
<md:AssertionConsumerService
        Location="/SAML2/ECP"
        index="4"
        Binding="urn:oasis:names:tc:SAML:2.0:bindings:PAOS"/>
<md:AssertionConsumerService
        Location="/SAML/POST"
        index="5"
        Binding="urn:oasis:names:tc:SAML:1.0:profiles:browser-post"/>
<md:AssertionConsumerService
        Location="/SAML/Artifact"
        index="6"
        Binding="urn:oasis:names:tc:SAML:1.0:profiles:artifact-01"/>
```

Figur 5. AssertConsumerService konfigurering

Det är till dessa förbindelser identifieringstjänsten skickar begäran. Deras stig definierades tidigare i HandlerURL fältet. De URL metadata som definierats för SP är i förbindelse med denna fil (Shibboleth 2018).

Inom Handler fältet beskrivs Shibboleths egna tjänster som möjliggör olikslags funktionalitet.

```
<Handler type="MetadataGenerator"
        Location="/Metadata"
        signing="false"/>
<Handler type="Status"
        Location="/Status"
        acl="127.0.0.1 ::1"/>
<Handler type="Session"
        Location="/Session"
        showAttributeValues="false"/>
```

Figur 6. Handler konfigurering

MetadataGenerator fältet skapar exempeldata som är enligt Shibboleths standarden. Statusfältet rapporterar över tjänsten via en XML fil. Statusfältet kräver en lokal förbindelse som definieras med acl fältet. Den andra Session fältet visar information över användares Shibboleth session. Om ShowAttributeValues fältet definieras till true, visas den överförda informationen över användaren.

Ifall det uppstår en bugg används Errors fältet.

```
<Errors supportContact="exempel@exempel.fi"
        redirectErrors="https://error.fi"/>
```

Figur 7. Error konfigurering

Information på buggen skickas till email adressen som definierats i supportContacts fält. I redirectErrors beskrivs till vilken sida användaren dirigeras då en bugg uppstått.

MetadataProvider beskriver den IdP information som används för tjänsten.

```
<MetadataProvider
        type="XML"
        url="https://testi.apro.tunnistus.fi/static/metadata/idp-metadata.xml"
        backingFilePath="/var/cache/shibboleth/idp-metadata.xml"
        reloadInterval="3000"/>
```

Figur 8. MetadataProvider konfigurering

Type fältet beskriver vilken slags fil som används. Url fältet beskriver destinationen där XML filen är. I BackingFilePath fältet definieras en stig där filen sparas lokalt. I en Linux miljö måste man också ha gett Shibboleth rättigheter för att editera filen. I reloadInterval fältet definieras det i sekunder hur ofta filen skall kollas ifall för ändringar.

AttributeExtractor fältet avsikt är att beskriva var identifieringstjänstens sessionsinformation som skickats blivit lagrat.

```
<AttributeExtractor
        type="XML"
        validate="true"
        path="exempel/attribute-map.xml"/>
```

Figur 9. AttributeExtractor konfigurering

Type fältet beskriver filformaten filen sparats i. Path fältet beskriver destinationen.

CredentialResolver fältet beskriver de certifikat och nyckel som tjänsten använder. Shibboleth inkluderar ett keygen program som möjliggör skapandet av certifikat och nyckel för testmiljön. Certifikatet kan skapas från Linux terminalen med keygen.sh kommandot.

```
<CredentialResolver type="File" key="exempel/sp-key.pem" certificate="exempel/sp-cert.pem"/>
```

Figur 10. CredentialResolver konfigurering

Värdet för File beskriver att tillstånden är i filformat. Både key och certificate fälten består av information över var nyckeln och certifikatet sparats.
För att möjlighetgöra information över användaren som blivit autentiserade, konverteras datan till format som Apache servern kan hantera. Detta definieras i en attributkarta som innehåller information över hur prefixet som tidigare definierats i Sessions fältet blir konverterat till ett AJP prefix.

5.4 Apache HTTP servers konfigurering

Apache HTTP installeras via Linux terminalen varefter konfigureringen kan påbörjas. Apache servern fungerar som en proxyserver för applikationsservern.

```
<Location /Shibboleth.sso>
     ShibRequestSetting applicationId exempel
     SetHandler shib
</Location>

<Location /secure>
     AuthType shibboleth
     ShibRequestSetting applicationId exempel
     ShibRequestSetting requireSession true
     Require valid-user
</Location>
```

Figur 11. Apache httpd.conf konfigurering

Här definieras det hur proxyservern skall vidarebefordra begäran med AJP protokollen till port 8009. Den stig som valts för stark autentiseringen skall definieras till Location fältet och innanför den definieras vilken typ av autentisering används, dvs Shibboleth. Därefter definieras Shibboleth specifika begäran (University of Michigan 2020).

5.5 Implementationens slutsats

Sessionens överförda information syns som nyckelvärdespar i attributkartan. Informationen som befinner sig i nyckelvärdespar och attributkartans id innehåller information som IDP, dvs i detta exempel Suomi.fi skickat av användaren. Hur informationen används hädanefter är beroende på tjänstens uppbyggnad, de varierande plattformar och programmeringsspråk. Enligt de instruktionerna som Suomi.fi skapat, skall användares Suomi.fi session avslutas innan begäran på utloggningen sker (Esuomi 2018).

6 SLUTSATS OCH DISKUSSION

Meningen med detta examensarbete var att forska och förstå olika metoder autentisering implementeras i produkter och tjänster. Sedan skulle det väljas en metod som kunde implementeras till en existerande portal. Med samma forskade jag skillnader mellan auktorisering, autentisering och stark autentisering. Det var mycket viktigt att användarna blev autentiserade från en pålitlig källa. Eftersom Suomi.fi autentiserar användaren från en extremt pålitlig källa blev teknologin valt. Suomi.fi testmiljös uppbyggandet krävde en webbserver och en SAML lösning. Apache HTTP valdes som server och Shibboleth som SAML lösningen. Med detta arbete har jag lärt mig mycket nytt om autentisering, auktorisering, stark autentisering och datasäkerhet.

Metoderna som valts tycker jag var rätta för arbetet då arbetet gjordes. Såklart finns det alltid olika metoder att skapa lösningar, men de som skilde de metoder jag valt från de andra var den starka källan varifrån användarna autentiserades och hur mycket information det redan fanns forskat över metoderna. Utan all den information som redan blivit forskat över de metoderna som jag valt, skulle implementationen varit väldigt svår att genomföra till slut.

Själv är jag rätt nöjd med resultaten och tycker att ha lyckas uppfylla de krav som fåtts från arbetsgivaren. Men det betyder dock inte att lösningen är perfekt. Ett problem uppstår då en användare som inte finns med i den finska folkregistret eller har sina uppgifter i någon av de populära bankservice som används i Finland försöker bli autentiserad. Då kan min lösning inte autentisera användaren, eftersom det inte finns en tillräckligt pålitlig källa varifrån användaren skulle kunna autentiseras. Men det var ett val som jag gjorde med tanke på att höja säkerheten på källan varifrån användaren autentiserades.

7 KÄLLOR / REFERENCES

CNN Business (2019) How hackable is your password? Hämtad 10.12 från:
https://edition.cnn.com/2019/04/22/uk/most-common-passwords-scli-gbr-intl/index.html

Miracl (2019) A Brief History of Authentication. Hämtad 19.12 från:
https://miracl.com/blog/a-brief-history-of-authentication/

Whatis (2015) Strong Authentication. Hämtad 5.2 från:
https://whatis.techtarget.com/definition/strong-authentication

OAuth (2019) User Autentication with Oauth 2.0. Hämtad 9.2 från:
https://oauth.net/articles/authentication/

Forbes (2017) No, Apple's Face ID Is Not A 'Secure Password'. Hämtad 11.2 från:
https://www.forbes.com/sites/jvchamary/2017/09/18/security-apple-face-id iphone-x/#5698e1364c83

Forbes (2019) You probably don't need to worry about this Face ID hack. Hämtad 13.2 från:
https://www.theverge.com/2019/8/9/20798569/face-id-hack-black-hat-conference-2018-glasses-tape

Bouljoub (2019) Electronic signature for authentication, bachelor thesis. Hämtad 20.2 från:
https://www.theseus.fi/bitstream/handle/10024/185630/Abdelwakil-Bouljoub-ELECTRONIC-SIGNATURE-FOR-AUTHENTICATION.pdf?sequence=2&isAllowed=y

YLE (2019) Koodareita on koko ajan enemmän, mutta uusia tarvitaisiin tuhansia vuositain. Hämtad 1.3 från:
https://yle.fi/uutiset/3-11094086

University of Toronto (2012) Single Sign-On and Shibboleth. Hämtad 3.3 från:
http://sites.utoronto.ca/security/projects/shibboleth.htm

Palveluhallinta Suomi.fi (2019) Palvelun käyttöönotto. Hämtad 3.17 från:
https://palveluhallinta.suomi.fi/fi/sivut/tunnistus/palvelukuvaus/palvelun-kuvaus

Apache (2020) What is an Apache HHTP Server Project. Hämtad 3.3 från:
https://httpd.apache.org/ABOUT_APACHE.html

Shibboleth (2010) NativeSPServiceLogout. Hämtad 3.14 från:
https://wiki.shibboleth.net/confluence/display/SHIB2/NativeSPServiceLogout

Shibboleth (2018) NativeSPAssertionConsumerService. Hämtad 3.17 från: https://wiki.shibboleth.net/confluence/display/SHIB2/NativeSPAssertionConsumerService

Shibboleth (2016) NativeSPSessionInitiator. Hämtad 3.17 från: https://wiki.shibboleth.net/confluence/display/SHIB2/NativeSPSessionInitiator

Esuomi (2018) Tekninen rajapintakuvaus. Hämtad 3.17 från: https://esuomi.fi/palveluntarjoajille/tunnistus/tekninen-aineisto/tekni-nen-rajapintakuvaus/

Shibboleth (2018) NativeSPApacheConfig. Hämtad 4.9 från: https://wiki.shibboleth.net/confluence/display/SHIB2/NativeSPApacheConfig

University of Michigan (2020) Information and technology services documentation. Hämtad 11.9 från: https://documentation.its.umich.edu/node/343

8 BILAGOR / APPENDICES