Nasjonal sikkerhetsmyndighet

Rapport
Sist oppdatert: 2016-05-12

Prosjektrapport – HTTPS for offentlige webtjenester

1 Innledning

Cybersikkerhetsavdelingen i Nasjonal sikkerhetsmyndighet (NSM) har utarbeidet anbefaling om bruk av *Hypertext Transfer Protocol Secure* (HTTPS) for å sikre offentlige webtjenester i Norge. NSM kommer også med anbefalinger på hvordan HTTPS bør implementeres for å gi tilstrekkelig sikring av de offentlige tjenestene i Norge. Rapporten vurderer også mulige gevinster og kostnader ved implementering av HTTPS.

2 Bakgrunn

Justisdepartementet (JD) har gitt NSM i oppdrag å vurdere hensiktsmessigheten ved å innføre krav om HTTPS som standard i offentlig forvaltning. Tilsvarende pålegg om sikring av offentlige webtjenester er gitt av andre myndigheter, eksempelvis amerikanske[1] og tyske[2].

3 NSMs anbefaling

NSM mener at alle offentlige tjenester på web alltid skal benytte HTTPS. Dette vil gi både autentisering av tjenesten og integritets- og konfidensialitetsbeskyttelse av informasjonen som overføres.

Autentisering av en tjeneste er avgjørende for å bekrefte at man utveksler data med korrekt tjeneste. Ved å ivareta integritetsbeskyttelse vet man at den samme informasjonen som ble sendt av avsender også når mottaker, det vil si at det ikke er gjort endringer i informasjonen under overføringen. Konfidensialitetsbeskyttelse gjør at uvedkommende ikke får innsyn i informasjonen som sendes.

For at HTTPS skal sikre kommunikasjonen må HTTPS benyttes på en sikker måte. NSM anbefaler at tiltrodde sertifikater benyttes sammen med moderne kryptografiske protokoller og mekanismer, og at systemet konfigureres korrekt. NSMs anbefalte implementering er beskrevet i kapittel 5.

NSM anbefaler også at private og kommersielle aktører benytter HTTPS for deres tjenester.

[1] White House HTTPS-Only Standard directive, https://www.whitehouse.gov/sites/default/files/omb/memoranda/2015/m-15-13.pdf

[2] Tysklands Bundesamt für Sicherheit in der Informationstechnik (BSI), https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/BSI/BSI_Act_BSIG.pdf?__blob=publicationFile&v=1

4 Beskrivelse av teknologien

4.1 Hva HTTPS er

Webtrafikk overføres ukryptert mellom en webtjener og en klient ved bruk av HTTP. Dette gjør at man verken er sikker på hvem man snakker med, om noen avleser innholdet eller om informasjonen er korrekt. HTTPS er overføring av webtrafikk over en sikker forbindelse. Ved å benytte krypteringsprotokollen *Transport Layer Security* (TLS) kan klienten verifisere identiteten til tjenesteleverandøren (autentisere). Webtrafikken overføres kryptert, og er dermed uleselig for uvedkommende (konfidensialitetsbeskyttelse). I tillegg kan ikke dataene som overføres manipuleres under overføring (integritetsbeskyttelse).

Krypteringsprotokollen TLS er den vanligste løsningen for autentisering, integritets- og konfidensialitetsbeskyttelse av kommunikasjon mellom ulike systemer over usikre kanaler, som Internett. TLS befinner seg på applikasjonslaget og de mest kjente anvendelsene er sikring av web trafikk (HTTPS), sikring av overføring av e-post[3] (SMTP over TLS) og opportunistisk TLS (STARTTLS).

TLS er en videreutvikling av *Secure Sockets Layer* (SSL) som ble publisert i 1995. SSL 3.0 var siste versjon av SSL og kom i 1996. I 2014 ble protokollen knekt og deretter forkastet i 2015 i RFC 7568[4].

4.2 Hva HTTPS gjør

HTTPS verifiserer identiteten til en webside eller tjeneste og krypterer nesten all informasjon mellom klient og tjener.

Bruken av TLS-protokollen deles opp i to faser [5]:

1) Etablering av sikker forbindelse mellom klient og tjener
   a. Valg av protokoll og kryptografiske algoritmer
   b. Autentisering av server (og eventuelt klient) basert på sertifikat
   c. Utveksling av kryptografiske parametere
   d. Etablering av krypteringsnøkkel for sesjonen
2) Utveksling av applikasjonsdata over den sikre forbindelsen

Når TLS er etablert for en tjeneste vil applikasjonen(e) overføre informasjon uten at andre som har tilgang til linjene trafikken passerer gjennom, kan se innholdet. TLS gir dermed samme beskyttelse mot avlytting på lokale nett (for eksempel nettkaféer, gjestenett i bedrifter og på flyplasser) som mot avlytting i stamnett (som Internett-tjenesteleverandører og statlige aktører).

Det er allikevel avgjørende at protokollen implementeres korrekt for å gi den ønskede beskyttelsen. Dette er nærmere beskrevet i kapittel 5.

[3] NSMs veileder U-02 Grunnleggende tiltak for sikring av e-post, https://www.nsm.stat.no/globalassets/dokumenter/veiledninger/systemteknisk-sikkerhet/u-02-grunnleggende-tiltak-for-sikring-av-e-post.pdf

[4] Internet Engineering Task Force (IETF), https://tools.ietf.org/html/rfc7568

[5] Basert på NSMs veileder U-03 Sikring av Windows TLS, kapittel 2.2, https://www.nsm.stat.no/globalassets/dokumenter/veiledninger/systemteknisk-sikkerhet/u-03_sikring_av_windows_tls.pdf

4.3 Hva HTTPS ikke gjør

HTTPS beskytter ikke mot trafikkanalyse, som hvem som kommuniserer og hvor mye. Videre er ikke tilstøtende protokoller som DNS omfattet som vil kunne avsløre hvilke tjenester som benyttes.

HTTPS beskytter ikke klienten eller tjeneren fra angrep, slik at man fortsatt har behov for herding av klient og tjener, eksempelvis som beskrevet i NSMs sjekkliste S-02[6].

Tilgjengelighet til tjenester blir ikke påvirket ved bruk av HTTPS.

HTTPS har begrenset mulighet til å beskytte mot falske sertifikater, men kompenserende tiltak kan iverksettes, som beskrevet i kapittel 5.

5 Anbefalt implementering

For at HTTPS skal sikre nettstedet og informasjonen må HTTPS, og dermed TLS, benyttes på en sikker måte. På bakgrunn av dette ønsker NSM å komme med anbefaling om bruk av TLS basert på følgende premisser:

Nyeste versjoner av TLS-protokollen har bedre sikkerhet. NSM anbefaler derfor å benytte TLS versjon 1.2. Dette er siste tilgjengelige versjon, og NSM vil anbefale at versjon 1.3 tas i bruk så snart den er klar. Dersom man benytter oppdatert programvare og installerer tilhørende sikkerhetsoppdateringer, vil disse protokollene være tilgjengelige, jamfør tiltak en og to i S-02. Merk at TLS kan befinne seg i en rekke ulike implementasjoner på en datamaskin. Foruten operativsystem, kan programmer som nettlesere, epost-lesere og Java-kjøremiljøet ha egne TLS-implementasjoner. Skal HTTPS benyttes som en del av HTTP/2 er det i standarden definert at TLS 1.2 skal benyttes[7]. HTTP/2 er beskrevet i kapittel 6.8. Tyskland anbefaler også at TLS 1.2 benyttes[8].

Sterke kryptografiske mekanismer som moderne algoritmer med tilfredsstillende nøkkellengder bør benyttes. NSMs kryptografiske krav[9] lister opp NSMs anbefalte og aksepterte kryptomekanismer. For bruk i TLS, samt i andre kryptoprotokoller, benyttes såkalte *cipher suites*. Dette er pakker med ulike kryptomekanismer, som spesifiserer hvilke mekanismer som benyttes for autentisering, integritetsbeskyttelse, nøkkeletablering og konfidensialitetsbeskyttelse. Det er sterkt anbefalt at disse pakkene tilbyr *(Perfect) forward secrecy*. *Forward secrecy* hindrer at kompromittering av langtidsnøkler (private nøkler) medfører kompromittering av sesjonsnøkler; altså dersom den private nøkkelen til sertifikatet i et system blir kompromittert, vil ikke tidligere kommunikasjon mot systemet være kompromittert. I tillegg bør *autentisert kryptering* benyttes. Autentisert kryptering gir, i tillegg til konfidensialitetsbeskyttelse av overføringen, også autentisering og integritetsbeskyttelse av dataoverføringen. På denne måten vet man at mottatte krypterte data kommer fra motparten.

[6] NSMs sjekkliste Ti viktige tiltak mot dataangrep, https://www.nsm.stat.no/globalassets/dokumenter/veiledninger/systemteknisk-sikkerhet/s-02-ti-viktige-tiltak-mot-dataangrep.pdf

[7] Internet Engineering Task Force (IETF), https://tools.ietf.org/html/rfc7540#section-9.2

[8] Tysklands Bundesamt für Sicherheit in der Informationstechnik (BSI), https://www.bsi.bund.de/DE/Themen/StandardsKriterien/Mindeststandards/SSL-TLS-Protokoll/SSL-TLS-Protokoll_node.html

[9] NSM Cryptographic Requirements, https://www.nsm.stat.no/globalassets/dokumenter/veiledninger/systemteknisk-sikkerhet/ncr3.0.pdf

Sertifiserte implementasjoner bør benyttes. Foruten at TLS-mekanismene, inkludert kryptoalgoritmer, er funksjonelt riktige, bør de evalueres for å oppnå tillit. Eksempler på anerkjente sertifiseringer er amerikanske Federal Information Prosessing Standard (FIPS) 140-2[10] og Common Criteria (CC).

Tiltrodde sertifikatutstedere bør benyttes for å utstede sertifikater til kommunikasjons-partene. Dette er sertifikater som er eksplisitt tiltrodd i nettlesere ved at rot-sertifikatet til sertifikatutstederen er forhåndsinstallert. Det er viktig at man har kontroll med hvilke sertifikatutstedere man har tiltro til, slik at kun sertifikater man ønsker skal benyttes, aksepteres. NSM anbefaler bruk av sertifikatutsteder underlagt norsk lovgivning. I tillegg bør sertifikatutstedere som støtter *certificate transparency* foretrekkes. *Certificate transparency* er en mekanisme som sjekker hvorvidt sertifikatet et nettsted presenterer er det samme sertifikatet som andre har fått servert, og på den måten hindre at noen lager forfalskede nettsteder med falske sertifikater.

HTTP Strict Transport Policy (HSTS) bør implementeres. HSTS instruerer nettlesere til å benytte HTTPS ved fremtidige besøk. Dette vil redusere behovet for å videreformidle HTTP trafikk til en sikker kobling, samt forhindre angrep som prøver å tvinge kommunikasjonen over på en usikker forbindelse.

Certificate Pinning bør benyttes. Dette styrker tilliten ytterligere til sertifikatene som benyttes ved at man kan binde bestemte sertifikatutstedere eller sertifikater mot bestemte tjenester. På denne måten kan man sørge for at kun virksomhetens autoriserte sertifikater og sertifikatutstedere benyttes mot virksomheten, noe som gjør det vanskeligere å benytte forfalskede sertifikater. For HTTPS anvendes *HTTP Public Key Pinning* (HPKP) for *Certificate Pinning*.

Maskinvarebaserte nøkler gir sterke nøkler ved hjelp av sterk nøkkelgenerering, samt bedre beskyttelse og bruk av nøklene. For systemer med høy belastning benyttes gjerne sikkerhetsmoduler for nøkkeloperasjoner også av ytelsesgrunner da dette er ressurskrevende i generelle prosessorer, men også systemer med lav belastning vil få økt sikkerhet.

6 Mulige gevinster og konsekvenser ved implementering

6.1 Økt tillit til tjenesten

Det norske samfunnet generelt og offentlig sektor spesielt, er avhengig av høy tillit hos norske borgere. Alvorlige hendelser av ulike slag kan potensielt ha både omdømmemessige, økonomiske, og strafferettslige konsekvenser, og i ytterste fall innebære fare for liv og helse. Offentlige tjenester har behov for konfidensialitetsbeskyttelse av de data som overføres til sluttbruker, som vil løses ved HTTPS. Samtidig vil HTTPS bidra til autentisitets- og integritetsbeskyttelse, som er avgjørende for offentlige webtjenester. Tap av dette vil potensielt ha et skadepotensiale for både tjenesteleverandør og bruker. Ved kontinuerlig å jobbe med å forbedre sikkerhet og brukervennlighet muliggjøres målsetningene om økt digitalisering av det norske samfunnet som beskrevet i digitaliserings-rundskrivet[11] og i handlingsplan for informasjonssikkerhet i statsforvaltningen[12].

6.2 Bruk av utdatert maskin- og programvare

NSM anbefaler at oppdatert program- og maskinvare benyttes blant annet i nevnte sjekkliste S-02. Maskin- og maskinvare som er støttet av leverandør og oppdatert vil i stor grad kunne implementere

[10] National Institute of Standards and Technology, http://csrc.nist.gov/publications/fips/fips1402.pdf

[11] Kommunal og moderniseringsdepartementet – digitaliseringsrundskrivet, https://www.regjeringen.no/no/dokumenter/digitaliseringsrundskrivet/id2462793/

[12] Kommunal og moderniseringsdepartementet – Handlingsplan for informasjonssikkerhet i staten, https://www.regjeringen.no/contentassets/b7d0918e555b418abda2993a71969cdc/handlingsplan_informasjonssikkerhet_staten.pdf

de anbefalingene beskrevet i kapittel 5, eksempelvis bruk av TLS 1.2. Sluttbrukerutstyr levert siden 2013 som Windows 7, Internet Explorer 11, iOS 5.1.1 og Android 4.4.2 er eksempler på utstyr som støtter de anbefalte protokollene. I de tilfeller der enten tjenesteleverandør eller sluttbrukere benytter utdatert maskin- eller programvare, som ikke støtter de anbefalte standarder, vil man risikere at tjenesten ikke kan leveres sluttbruker. Sluttbruker inkluderer her også andre tjenester som er avhengige av andre tjenester, (eksempelvis brukere av et programmeringsgrensesnitt (API) over HTTPS). I de tilfeller der utdatert maskin- og programvare benyttes, enten hos sluttbruker eller hos tjenesteleverandøren, vil det potensielt medføre økte kostnader til oppdatering eller at anbefalt sikkerhet ikke oppnås.

6.3 Krav til HTTPS fra internasjonale aktører

I tillegg til nasjonale føringer fra myndigheter, eksempelvis amerikanske[13] og tyske[14], viser også toneangivende internasjonale aktører stort fokus på sikre tilkoblinger. Eksempelvis belønner Google sikre sider i deres søkeresultater.[15] For statlige sider som har behov for eksponering av eget innhold vil dermed HTTPS gi økt synlighet for aktuelle sluttbrukere.

Tilsvarende planlegger nettleserleverandører å markere HTTP sider som usikre. Dette gjelder også der HTTPS er implementert, men med utdaterte og usikre sertifikater.

Payment Card Industry Data Security Standard (PCI DSS) er en internasjonal standard for organisasjoner som behandler betalingstransaksjoner. Kravene omfatter de aktører som oppbevarer og behandler kredittkortinformasjon. PCI DSS v3.1-standarden krever at alle eksisterende systemer må tilby minst TLS 1.1 innen juni 2016 og gå totalt bort fra eldre versjoner av SSL og TLS innen juni 2018. Nye systemer må minimum benytte TLS 1.1, men TLS 1.2 anbefales[16].

6.4 Potensielle implementerings- og driftskostnader

Kostnaden ved å implementere HTTPS vil variere. Anskaffelse- og vedlikeholdskostnader av sertifikater ansees ikke å være kostnadsdrivende. For eksisterende tjenester vil det påløpe kostnader relatert til implementering av og vedlikehold av selve tjenesten. Med dette menes for eksempel å konfigurere nettverksutstyr, endre tjenestekonfigurasjon, tilpasse innhold og eventuell anskaffelse av maskinvarebasert nøkkelmodul.

Hvis oppdatert maskin- og programvare benyttes, som beskrevet i tiltak en og to i NSMs sjekkliste for ti viktige tiltak mot dataangrep, S-02, vil det trolig være lite behov for å anskaffe ny maskin- og programvare for å implementere HTTPS. Dessverre er det NSMs erfaring at mange synder mot tiltak en og to i S-02, slik at den faktiske kostnaden ved å implementere kan bli betydelig.

6.5 Forvalting av innhold fra flere tjenestetilbydere (mixed content)

Mange webtjenester henter innhold fra eksterne kilder. Eksempler på slikt innhold er script, bilder, skrifttyper og videoklipp. Når en tjeneste skal oppgraderes til HTTPS bør også disse kildene lastes sikkert for å kunne bli vist sluttbruker. På eksisterende tjenester med mange sider og innhold fra ulike

[13] M-15-13 HTTPS-Only Standard directive,
https://www.whitehouse.gov/sites/default/files/omb/memoranda/2015/m-15-13.pdf

[14] Tysklands Bundesamt für Sicherheit in der Informationstechnik (BSI),
https://www.bsi.bund.de/DE/Themen/StandardsKriterien/Mindeststandards/SSL-TLS-Protokoll/SSL-TLS-Protokoll_node.html

[15] Google webmaster Central Blog, https://webmasters.googleblog.com/2014/08/https-as-ranking-signal.html

[16] PCI Security Standards Council Guidance document,
https://cdn2.hubspot.net/hubfs/281302/Resources/Migrating_from_SSL_and_Early_TLS_-v12.pdf

leverandører kan det å håndtere *mixed content* representere et stort prosjekt for å tilpasses HTTPS. Det finnes enkle tekniske tiltak tilgjengelige som kan løse majoriteten av disse utfordringene med lite ressurser. Det må også nevnes at mange tjenesteleverandører allerede støtter HTTPS, og i de tilfellene er det svært enkelt å implementere HTTPS.

6.6 Støtte til gjennomføring

Implementasjon av HTTPS vil kreve kompetanse og ressurser hos tjenesteeier. Det bør vurderes å etablere veiledningsmateriale og konfigurasjonseksempler for å lette implementasjonen av HTTPS. I tillegg bør man etablere bestillerkompetanse for etablering av nye tjenester..

6.7 Påvirkning på ytelse

Det eksisterer flere vurderinger og ytelsesmålinger om hvorvidt implementering av HTTPS påvirker ytelsen til webtjenester negativt. Konklusjonene er at ytelsen påvirkes marginalt, og i enkelte tilfeller forbedres (som i kombinasjon med HTTP/2). W3C[17], Google[18] og Akamai[19] er eksempler på internasjonale aktører som har undersøkt ytelsespåvirkningen.

I de tilfeller der maskinvare er en begrensning for ytelsen til en tjeneste, kan man vurdere å avlaste asymmetriske kryptooperasjoner i en maskinvarebasert nøkkelmodul. Nøkkelhåndtering i maskinvare vil i tillegg til økt sikkerhet, gi økt ytelse.

6.8 Påvirkning på andre relevante teknologier

Vi gir her en kort oversikt over relevante teknologier som bidrar både til ytelse og sikkerhet og som bør sees i sammenheng med HTTPS. Imidlertid danner dette skrivet seg om HTTPS, og denne paragrafen gir dermed kun kortfattet informasjon om disse tilleggsteknologiene.

HTTP/2

I dag benytter web-tjenester HTTP/1.1. Denne standarden er gammel, fra 1999, og HTTP/2 ble for kort tid siden endelig vedtatt[20]. HTTP/2 er mer effektiv for dataoverføring og reduserer båndbreddebehov, noe som gjør at websider vises raskere. Både Google Chrome, Microsoft Internet Explorer 11 og Mozilla Firefox[21] støtter HTTP/2. Det er viktig å påpeke at selv om HTTP/2 ikke krever kryptering, så har Google og Mozilla sagt at de vil kun benytte HTTP/2 med kryptering. Tester har vist at bruk av HTTP/2 med riktig konfigurert kryptering kan gi inntil 20% raskere sidelasting. Dette gir økt brukervennlighet og en potensiell økonomisk gevinst ved å støtte HTTP/2.

Security Headers

NSM har i anbefalingene adressert flere *security headers* i HTTP, nemlig HPKP og HSTS. Selv om ikke alle er adressert, vil de allikevel kunne bedre sikkerheten hos web-tjenester. Enkelte vil også kunne forenkle implementasjonen av HTTPS. Eksempelvis instruerer `content-security-policy: upgrade-insecure-requests` brukerens nettleser til å laste usikre elementer via en sikker forbindelse. Dette vil i stor grad løse utfordringene med mixed content, da mange usikre elementer fra

[17] World Wide Web Consortiim (W3C) - Securing the web, https://www.w3.org/2001/tag/doc/web-https

[18] Webside av Ilya Grigorik, Web performance engineer hos Google, Inc ,https://istlsfastyet.com/

[19] Akamai Blog, https://blogs.akamai.com/2015/08/is-http2-worth-the-performance-price-of-tls.html

[20] Internet Engineering Task Force (IETF), https://tools.ietf.org/html/rfc7540

[21] Oversikt over HTTP/2 støtte I nettlesere utviklet av Alexis Deveria, , http://caniuse.com/#search=HTTP%2F2

3. part har mulighet til å lastes via sikker forbindelse. Det finnes security headers som ikke er adressert her. Det er anbefalt at samtlige security headers vurderes i oppsett av web-tjenester.

DNSSEC

DNSSEC vil gi ytterligere autentisering av webtjenesten, i tillegg til andre tjenester som for eksempel e-post. DNSSEC er en utvidelse av DNS. DNS er den grunnleggende teknologien som benyttes for å oversette et domenenavn (for eksempel www.nsm.stat.no) til en IP-adresse. Tradisjonelt er problemet med DNS at man ikke vet hvorvidt svaret man mottar er korrekt slik at man potensielt kan ende opp med å benytte en falsk tjeneste. DNSSEC ble standardisert i mars 2005[22] og satt i produksjon for norske domener 9. desember 2014[23]. DNSSEC tar i bruk kryptografiske mekanismer for å autentisere og gi integritetsbeskyttelse for DNS oppslag. NSM er kjent med at flere nasjoner har anbefalt, og enkelte pålagt, å benytte DNSSEC. DNSSEC blir normalt satt opp og administrert av driftsleverandør, det vil si at virksomheter ikke må implementere, men stille krav om at teknologien skal benyttes. Anbefaling rundt HTTPS adressert i dette skrivet er ikke i konflikt med eventuelle anbefalinger for bruk av DNSSEC.

7 Andre vurderinger

7.1 Utvidet bruk av TLS

NSM anbefaler størst mulig bruk av TLS og vil utgi en generell TLS-veiledning, samt mulige andre applikasjonsspesifikke veiledninger som benytter TLS, eksempelvis IMAP og SMTP. NSM anbefaler at TLS 1.2 benyttes i de tjenester der det er tilgjengelig.

7.2 Forslag til innføringshastighet

NSM anbefaler på lik linje med amerikanske[24] og tyske[25] myndigheter at nye webtjenester settes opp som standard med kun HTTPS basert på NSMs anbefalinger i dette skriv. For eksisterende tjenester bør det defineres en sluttdato der alle skal være over på HTTPS, men NSM har som en del av dette prosjektet valgt å ikke komme med et spesifikt forslag. Eksempelvis bør kompetanse hos utøvende aktører og kompleksitet på de ulike tjenester vurderes samt at det bør foreligge tilstrekkelig veiledningsmateriale før tidsplan spesifiseres.

[22] Internet Engineering Task Force (IETF), https://www.ietf.org/rfc/rfc4033.txt

[23] Uninett Norid om DNSSEC, https://www.norid.no/no/registrar/system/videreutvikling/dnssec/omdnssec/

[24] White House HTTPS-Only Standard directive, https://www.whitehouse.gov/sites/default/files/omb/memoranda/2015/m-15-13.pdf

[25] Tysklands Bundesamt für Sicherheit in der Informationstechnik (BSI), https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/BSI/BSI_Act_BSIG.pdf?__blob=publicationFile&v=1