Nasjonal sikkerhetsmyndighet

IT-veiledning for ugraderte systemer nr. 15 (U-15).
Oppdatert: 2016-10-20

HTTP over TLS

Hypertext Transport Protocol Secure

Hvordan autentisere nettsteder og konfidensialitets- og integritetsbeskytte webtrafikk.

Dette dokumentet beskriver hvordan Hypertext Transport Protocol Secure (HTTPS) virker og hvordan man kan sikre denne protokollen for å oppnå autentisitets-, konfidensialitets- og integritets-beskyttelse av webtrafikk.

Nasjonal sikkerhetsmyndighet

Nasjonal sikkerhetsmyndighet er tverrsektoriell fag- og tilsynsmyndighet innenfor forebyggende sikkerhetstjeneste i Norge og forvalter lov om forebyggende sikkerhet av 20. mars 1998. Hensikten med forebyggende sikkerhet er å motvirke trusler mot rikets selvstendighet og sikkerhet og andre vitale nasjonale sikkerhetsinteresser, primært spionasje, sabotasje og terrorhandlinger. Forebyggende sikkerhetstiltak skal ikke være mer inngripende enn strengt nødvendig, og skal bidra til et robust og sikkert samfunn.

Hensikt med veiledning

NSM sin veiledningsvirksomhet skal bygge kompetanse og øke sikkerhetsnivået i virksomhetene, gjennom økt motivasjon, evne og vilje til å gjennomføre sikkerhetstiltak. NSM gir jevnlig ut veiledninger til hjelp for implementering av de krav sikkerhetsloven stiller. NSM publiserer også veiledninger innen andre fagområder relatert til forebyggende sikkerhetsarbeid.

Postadresse
Postboks 814
1306 Sandvika

Sivil telefon/telefax
+47 67 86 40 00/+47 67 86 40 09

Militær telefon/telefaks
515 40 00/515 40 09

Webadresse
nsm.stat.no

E-postadresse
post@nsm.stat.no

Innhold

1 Innledning

Hypertext Transfer Protocol (HTTP) er kommunikasjonsprotokollen til webtrafikk, og *Hypertext Transfer Protocol Secure* (HTTPS) er HTTP over en sikker forbindelse. HTTPS er altså ingen egen protokoll, men normal HTTP-trafikk over *Transport Layer Security* (TLS). HTTPS tilbyr autentisering av serveren, samt konfidensialitets- og integritetsbeskyttelse av innholdet som overføres. Dersom man i tillegg ønsker autentisering av klienten kan man benytte klient-sertifikat for to-veis-autentisering.

Nettsteder som benytter HTTPS har en URL som begynner med «https://» i stedet for «http://», og som standard benyttes TCP port 443 i stedet for 80.

NSM anbefaler at mest mulig webtrafikk autentiseres, integritets- og konfidensialitetsbeskyttes. Dette vil sikre enorme mengder data over Internett, hvor fortsatt under halvparten av webtrafikken er sikret.

En webserver[1] som skal støtte HTTPS-forbindelser må konfigureres med et sertifikat som blant annet inneholder informasjon om tilbyderen av webtjenesten og hvor lenge sertifikatet er gyldig. Sertifikatet må være signert av en sertifikatutsteder som nettlesere har tiltro til.

Formålet med denne veiledningen er å gi anbefalinger om hvordan man kan etablere sikker overføring av webtrafikk. Ved å implementere anbefalingene vil man øke tilliten til at webtrafikken overføres sikkert. Hvis risikovurderingen tilsier at spesifikke anbefalinger ikke implementeres anbefaler NSM at kompenserende tiltak iverksettes.

Kontaktpunkt for denne veiledningen er post@nsm.stat.no. Kommentarer og innspill mottas med takk.

[1] Dette inkluderer utstyr som TLS-akseleratorer, lastbalanserere, mv.

2 Om HTTPS

Med tradisjonell HTTP overføres webtrafikk ukryptert mellom en webserver og en klient. 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 (autentisering). Deretter overføres webtrafikken kryptert, og er dermed uleselig for uvedkommende (konfidensialitetsbeskyttelse). I tillegg kan ikke dataene som overføres manipuleres under overføring (integritetsbeskyttelse).

Bruken av HTTPS deles opp i to faser:

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 webtrafikk over den sikre forbindelsen

HTTPS gir samme beskyttelse mot avlytting på lokale nett, som mot avlytting på Internett.

3 Anbefalte tiltak

HTTPS må benyttes på en sikker måte for å autentisere nettsteder og konfidensialitets- og integritetsbeskytte webtrafikk. På bakgrunn av dette ønsker NSM å komme med følgende sikkerhetstiltak.

3.1 Benytt en tiltrodd sertifikatutsteder

Velg en leverandør som har åpenhet om utstedte sertifikater (Certificate Transparency). *Certificate transparency* innebærer en offentlig logg over alle utstedte sertifikater fra sertifikatutstederen. En slik offentlig logg gir mulighet for innsyn i, og kontroll med, hvilke sertifikater som er utstedt. *Certificate transparency* er foreslått i RFC6962.

Velg en leverandør som tilbyr Extended Validation (EV)-sertifikater. EV-sertifikater er sertifikater hvor sertifikatutsteder har gjort utvidede sjekker av identiteten til virksomheten, f.eks. at virksomheten har et organisasjonsnummer i Brønnøysundregisterne og har et virkende telefonnummer. EV-sertifikater får typisk grønn adresselinje. Fra 1. januar 2015, har Google Chrome krevd at alle EV-sertifikater er underlagt *Certificate Transparency*.

Velg en sertifikatutsteder underlagt norsk lovgivning. En sertifikatutsteder underlagt norsk lovgivning kan ikke presses av utenlandske aktører til å utstede falske sertifikater. Ved å begrense sertifikatutstedere til norske, vil klienter kunne fjerne flere sertifikatutstedere de ikke behøver å stole på. Reduksjon av tiltrodde sertifikatutstedere øker sikkerheten da enhver tiltrodd sertifikatutsteder kan utstede sertifikater for ethvert nettsted. I tillegg bør man binde sertifikatet og sertifikatutstederen mot nettstedet gjennom bruk av *HPKP* (se kapittel 3.7) slik at falske sertifikatet ikke kan benyttes.

Velg en leverandør som tilbyr Online Certificate Status Protocol (OCSP). OCSP er en måte nettlesere kan verifisere at sertifikatet som websiden tilbyr ikke er revokert. Dette er et alternativ til *Certificate Revocation List* (CRL). CRL skalerer dårlig siden listene kan bli veldig store, og listene oppdateres ikke i sanntid, men typisk på ukesbasis. OCSP fungerer på den måten at nettleseren i stedet for å regelmessig laste ned en stor liste over revokerte sertifikater kontakter sertifikatutstederen hver gang et gitt sertifikatet benyttes for å sjekke om det fremdeles er gyldig. Det er positivt fordi kun relevante sertifikater sjekkes, samt at sertifikater ugyldiggjøres umiddelbart ved revokering. Samtidig har dette konsekvenser for privatliv (sertifikatutsteder får beskjed hver gang en nettleser besøker en webside) og for ytelse (sertifikatutsteder må takle potensiell stor pågang på sertifikatsjekker). Se også kapittel 3.6 om *OCSP stapling*. OCSP er definert i RFC6960.

3.2 Benytt TLS på en sikker måte

NSM utgir råd om kryptografiske mekanismer og algoritmer i *NSM Cryptographic Requirements*[2] og spesifikke råd om TLS i en egen veiledning[3]. For en oppdatert oversikt over hvilke mekanismer, algoritmer og nøkkellengder som anbefales, må denne veiledningen studeres. Anbefalingene i denne veiledningen er i overenstemmelse med overnevnte dokumenter ved publiseringstidspunktet, og kan oppsummeres som følger:

[2] https://www.nsm.stat.no/globalassets/dokumenter/veiledninger/systemteknisk-sikkerhet/ncr3.1.pdf

[3] https://www.nsm.stat.no/globalassets/dokumenter/veiledninger/systemteknisk-sikkerhet/tls.pdf

Benytt nyeste versjon av TLS. For tiden er dette TLS 1.2, men standardiseringen av TLS 1.3 er i ferd med å avsluttes. Når implementasjoner av TLS 1.3 er tilgengelige for webservere og nettlesere bør man ta disse i bruk. TLS 1.2 har for tiden ingen kjente sårbarheter, og så lenge dette stemmer kan man tilby TLS 1.2 og TLS 1.3 i parallell for bakoverkompabilitet med gamle klienter.

Benytt sertifikater med tilstrekkelig nøkkellengde og avtrykk. Benytt elliptisk kurve-baserte sertifikater med minimum 256 bits nøkkellengde eller RSA-baserte sertifikater med minimum 3072 bits nøkkellengde. Benytt SHA-2-baserte avtrykk med minimum 256 bits lengde.

Benytt sikre cipher suites. Sikre cipher suites inkluderer, i prioritert rekkefølge:

TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 (0x00C02C)

TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0x00C030)

TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 (0x00C02B)

TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0x00C02F)

TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 (0x00C024)

TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 (0x00C028)

TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 (0x00C023)

TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 (0x00C027)

Benytt maskinvarebaserte kryptomoduler. *Hardware Security Module* (HSM) er en kryptografisk brikke som tilbyr kryptooperasjoner i maskinvare, fremfor programvare. Dette gjør at nøklene er bedre beskyttet og at maskinens prosessor avlastes for kryptografiske operasjoner. Dette gir derfor bedre sikkerhet og bedre ytelse.

3.3 Videresend HTTP-trafikk til HTTPS

Tilby nettstedet kun over HTTPS. Dersom klienter prøver å koble seg til med HTTP, så videresend forespørselen til HTTPS.

3.4 Benytt HTTP Strict Transport Security (HSTS)

HTTP har et *header*-felt som kan instruere nettlesere til utelukkende å benytte HTTPS for fremtidige tilkoblinger. Dette settes første gangen en nettleser besøker et nettsted og instruksen har en konfigurerbar varighet, for eksempel to år. Videre forlenges instruksen ved hvert etterfølgende besøk til nettstedet. HSTS kan settes på hoveddomenet og underdomenet. I tillegg kan et preload-direktiv spesifiseres som indikerer at nettstedet er villig til å hardkodes til å benytte HTTPS inn i nettleserne, mer om dette i kapittel 3.5. HSTS er spesifisert i RFC6797.

3.5 Benytt Strict Transport Security (STS) Preloading

De ulike nettleser-leverandørene tilbyr en måte å hardkode tvungen HTTPS for et nettsted i nettleseren i stedet for å benytte HSTS-header-feltet til dette. STS har fordelen at også første besøk til et nettsted benytter HTTPS, noe som forhindrer angrep som vil være mulige ved første besøk, altså før nettleseren har sett HSTS-header-feltet. Man kan be om å bli inkludert i flere nettlesere på https://hstspreload.appspot.com/.

3.6 Benytt OCSP stapling

OCSP stapling er en mekanisme hvor nettstedet tilbyr sertifikatstatusinformasjon ved oppkobling til nettstedet. Klienten slipper derfor å kontakte sertifikatutsteder for å verifisere at sertifikatet ikke er revokert. Dette medfører også at sertifikatutstederen ikke vet hvem som besøker nettstedet. I tillegg reduseres lasten på OCSP-tjenesten, samt at ytelsen til klienten forbedres fordi det ikke er nødvendig med ekstra forbindelser til OCSP-tjenesten.

Muligheten til å sende med OCSP-svar i TLS-oppkobling er spesifisert i RFC6066.

For ytterligere sikkerhet, kan man benytte *OCSP Must-Staple*. Dette er et felt i sertifikatet som instruerer nettlesere til å avbryte tilkoblingen dersom det ikke foreligger en OCSP-staple ved oppkobling. Dette gjør revokerte sertifikater ubrukelige. OCSP Must-Staple er foreslått i RFC7633.

3.7 Benytt HTTP Public Key Pinning (HPKP)

HTTP har et header-felt som kan binde nettstedet mot et sertifikat. I RFC-en anbefales det at man alltid skal tilby to bindinger, en binding mot et sertifikat som er i aktiv bruk i sertifikatkjeden, og en binding mot et reserve-sertifikat som ikke aktivt benyttes. NSM anbefaler at man ikke binder mot eget sertifikat, men i stedet binder mot sertifikatutsteders sertifikat(-er). Dette reduserer administrasjonen og potensialet for feilkonfigurasjon som kan stenge brukere ute. HPKP er den eneste måten som forhindrer sertifikatutstedere fra å utstede falske sertifikater for nettstedet. HPKP er spesifisert i RFC7469.

3.8 Benytt også andre sikkerhetsrelaterte header-felt

Det anbefales at følgende header-felt også benyttes, selv om ikke alle er direkte knyttet til HTTPS:

Benytt «HttpOnly;Secure» for å hindre avlytting av informasjonskapsler. Dette hindrer at informasjonskapslene sendes i klartekst og at klientside-script kan lese innholdet i dem.

Benytt «X-Frame-Options: deny» for å hindre clickjacking-angrep. Dersom websiden skal inkluderes av andre websider på samme domene kan "sameorigin" benyttes i stedet for "deny".

Benytt «X-XSS-Protection: 1; mode=block» for å hindre cross site scripting-angrep.

Benytt «Content-Security-Policy: default-src https:» for å laste alle ressurser over HTTPS. CSP tillater også å skille på forskjellige typer ressurser, som for eksempel script og bilder, på en mer finkornet måte.

Hvilke header-felt som er relevante må sees opp mot hva slags tjeneste som tilbys og eventuelt negative konsekvenser, for eksempel at tredjeparts leverandører ikke har støtte for HTTPS og dermed ikke vil kunne benyttes.

3.9 Benytt klientsertifikater for to-veis autentisering

Man kan benytte sertifikater på klienter for to-veis autentisering. Dette forutsetter at man har kjennskap til klientene, slik at man har fått installert sertifikater på disse.

3.10 Iverksett klient-tiltak for sikkerhet i dybden

I hovedsak er webserveren ansvarlig for sikkerheten ved tilkobling til denne. Det er allikevel mulig å implementere sikkerhetstiltak på klientmaskiner som brukerne selv er ansvarlige for. Det følgende er ikke en uttømmende liste over mulige klienttiltak.

- Benytt en oppdatert nettleser som tilbyr sikre protokoller og algoritmer.
- Fjern uønskede rot-sertifikater fra operativsystemets/nettleserens lister over tiltrodde sertifikatutstedere, og slå av automatisk oppdatering av disse listene.
- Fjern øvrige TLS *cipher suites* fra operativsystem/nettleser som ikke står i listen over.
- Benytt nettleseren i «vanlig» modus for å sørge for at HSTS- og HPKP-instruksene skal overholdes. Ved privat/inkognito-modus, vil slike instrukser ikke «huskes» mellom sesjoner og dermed ikke ha effekt.
- Konfigurer nettleseren til at manglende OCSP-svar skal tolkes som at sertifikatet er revokert.
- Benytt en nettleser eller nettleser-utvidelse som alltid forsøker å koble til med HTTPS før HTTP.
- Benytt en nettleser eller nettleser-utvidelse som varsler når nettsteder bytter sertifikater.

Med overnevnte server-tiltak på plass vil klient-tiltakene tilby sikkerhet i dybden, dvs. redundante sikkerhetsmekanismer.

3.11 Verifiser at anbefalingene virker som tiltenkt

For å verifisere at sikkerhetstiltakene på både webserver og klient er effektive, kan man benytte løsninger for å overvåke TLS-oppkoblinger. Slike løsninger kan blant annet verifisere hvilke sertifikater og sertifikatutstedere som utveksles og benyttes i oppkoblingsfasen, og hvilken TLS-versjon og hvilke kryptografiske mekanismer som tilbys og tas i bruk. På denne måten kan man stoppe trafikk som ikke møter organisasjonens retningslinjer.

Det finnes også en rekke nettsteder hvor man kan verifisere sikkerhetskonfigurasjonen til nettsted og klienter. Noen av disse er:

- Benytt Qualys SSL Server Test[4] og securityheaders.io[5] for å verifisere at sikkerhetstiltakene på webserveren er effektive.
- Benytt en tjeneste som badssl.com[6] for å verifisere at sikkerhetstiltakene på klienten er effektive.

Dersom overnevnte punkter er verifisert kan man med rimelig sikkerhet si at anbefalingene virker som tiltenkt.

[4] https://www.ssllabs.com/ssltest/

[5] securityheaders.io

[6] https://badssl.com/

Vedlegg A Dokumenthistorikk

2016-05-27 Dokumentet ble opprettet.

2016-09-06 Intern høring.

2016-10-01 Publisering.

2016-10-20 Lagt til vedlegg for verifikasjon.

Vedlegg B Verifikasjon

Tiltak	Verifikasjonsmetode	Status
Velg en leverandør som har åpenhet om utstedte sertifikater (Certificate Transparency).	Qualys SSL Server Test > Authentication > Server Key and Certificate #1 > Certificate Transparency := Yes	
Velg en leverandør som tilbyr Extended Validation (EV)-sertifikater.	Qualys SSL Server Test > Authentication > Server Key and Certificate #1 > Extended Validation := Yes	
Velg en sertifikatutsteder underlagt norsk lovgivning.	Qualys SSL Server Test > Authentication > Certification Path := Buypass Per 2016-10-20 er kun Buypass AS sertifikatutsteder underlagt norsk lov.	
Velg en leverandør som tilbyr Online Certificate Status Protocol (OCSP).	Qualys SSL Server Test > Authentication > Server Key and Certificate #1 > Revocation information := OCSP (inkludert URL)	
Benytt nyeste versjon av TLS.	Qualys SSL Server Test > Configuration > Protocols > TLS 1.2 := Yes	
Benytt sertifikater med tilstrekkelig nøkkellengde og avtrykk.	Qualys SSL Server Test > Authentication > Server Key and Certificate #1 > Key := RSA 3072/4096 bits eller EC 256/384 bits > Authentication > Server Key and Certificate #1 > Signature algorithm := SHA256withRSA	
Benytt sikre cipher suites.	Qualys SSL Server Test > Configuration > Cipher Suites := [se kapittel 3.2]	
Benytt maskinvarebaserte kryptomoduler.	[Manuell verifikasjon]	
Videresend HTTP-trafikk til HTTPS	Besøk http://example.com/ og verifiser at https://example.com/ besøkes.	
Benytt HTTP Strict Transport Security (HSTS)	securityheaders.io > Raw Headers > Strict-Transport-Security := max-age=63072000; includeSubDomains; preload	
Benytt Strict Transport Security (STS) Preloading	Qualys SSL Server Test > Configuration > Protocol Details > HSTS Preloading := [nettlesere]	
Benytt OCSP stapling	Qualys SSL Server Test > Configuration > Protocol Details > OCSP stapling := Yes	
Benytt HTTP Public Key Pinning (HPKP)	securityheaders.io > Raw Headers > Public-Key-Pins := [liste med minst to ‘pins’]	
Benytt «HttpOnly;Secure» for å hindre avlytting av informasjonskapsler.	securityheaders.io > Raw Headers > Set-Cookie := [må inneholde HttpOnly;Secure]	
Benytt «X-Frame-Options: deny» for å hindre clickjacking-angrep.	securityheaders.io > Raw Headers > X-Frame-Options := DENY / SAMEORIGIN	
Benytt «X-XSS-Protection: 1; mode=block» for å hindre cross site scripting-angrep.	securityheaders.io > Raw Headers > X-Xss-Protection := 1; mode=block	
Benytt «Content-Security-Policy: default-src https:» for å laste alle ressurser over HTTPS.	securityheaders.io > Raw Headers > Content-Security-Policy := default-src https:	
Iverksett klient-tiltak for sikkerhet i dybden	badssl.com > Verifiser at nettleser oppfører seg som forventet	