

TLS

Sikring av Windows TLS

Valg av protokoll og kryptografiske algoritmer for Windows Schannel TLS

Dette dokumentet inneholder NSMs anbefalinger om hvilke TLS-protokoller og kryptografiske algoritmer som bør brukes av Schannel i Windows 7 SP1, Windows Server 2008 R2 og senere versjoner. Målgruppen er personell som administrerer ugraderte Windows systemer i offentlig forvaltning.



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
E-postadresse
post@nsm.stat.no

Militær telefon/telefaks
515 40 00/515 40 09

Internettadresse
www.nsm.stat.no

Innhold

1 Innledning	5
2 Bakgrunn	6
2.1 Hvordan TLS fungerer	6
2.2 Valg av TLS-protokoll og kryptografiske algoritmer	7
3 Anbefalinger	10
3.1 Anbefalte TLS-protokoller	10
3.2 Anbefalte cipher suiter	10
Vedlegg A Anbefalte cipher suiter (detaljert)	12
A.1 Del 1 Foretrukne <i>Cipher Suiter</i>	12
A.2 Del 2 Akseptable <i>Cipher Suiter</i>	13
Vedlegg B Konfigurasjon av Schannel	14
B.1 Konfigurasjon vha Windows Registry Editor	14
B.2 Konfigurasjon vha PowerShell	14
Vedlegg C Referanser	17
Vedlegg D Dokumenthistorie	18

1 Innledning

Windows 7 SP1 klienter må ofte kommunisere med servere over et åpent nettverk (f eks internett). For å sikre kommunikasjonen kan *Transport Layer Security* (TLS) brukes. Ved bruk av TLS kan vi autentisere servere og klienter og beskytte konfidensialiteten og integriteten til dataene som utveksles.

TLS sin forgjenger er *Secure Socket Layer* (SSL). SSL 2.0 [1] og SSL 3.0 [2] ble utviklet av Netscape på midten av 1990-tallet for å sikre kommunikasjon over internett. Litt senere tok *Internet Engineering Task Force* (IETF) utgangspunkt i SSL 3.0 for å lage en standardisert protokoll. Resultatet ble TLS 1.0 [3], som er en forbedring av SSL 3.0. TLS 1.0 har senere blitt forbedret, hvor dette har resultert i protokollene TLS 1.1 [4] og TLS 1.2 [5].

TLS- og SSL-protokollene kan bruke ulike kryptografiske algoritmer for å sikre kommunikasjonen mellom klienten og serveren. Over tid har det blitt avdekket sårbarheter i flere av de kryptografiske algoritmene og protokollene. Av TLS- og SSL-protokollene er SSL-protokollene de svakeste. Det bør spesielt nevnes at SSL 2.0 ikke anbefales brukt lenger [6].

I dag finnes det ingen grunn til å bruke SSL-protokollene lenger, siden alle moderne operativsystemer har støtte for en eller flere av TLS-protokollene. Fokuset i resten av dokumentet vil derfor være på TLS-protokollene.

Formålet med dokumentet er å gi en anbefaling om hvilke TLS-protokoller og kryptografiske algoritmer som bør brukes for Windows 7 SP1 klienter, Windows Server 2008 R2 og senere versjoner. Sammen med dokumentet er det vedlagt eksempler på konfigurasjonsfiler som velger de anbefalte TLS-protokollene og kryptografiske algoritmene for nevnte Windows-versjoner.

Målgruppen for dokumentet er personell som administrerer ugraderte Windows systemer i offentlig forvaltning.

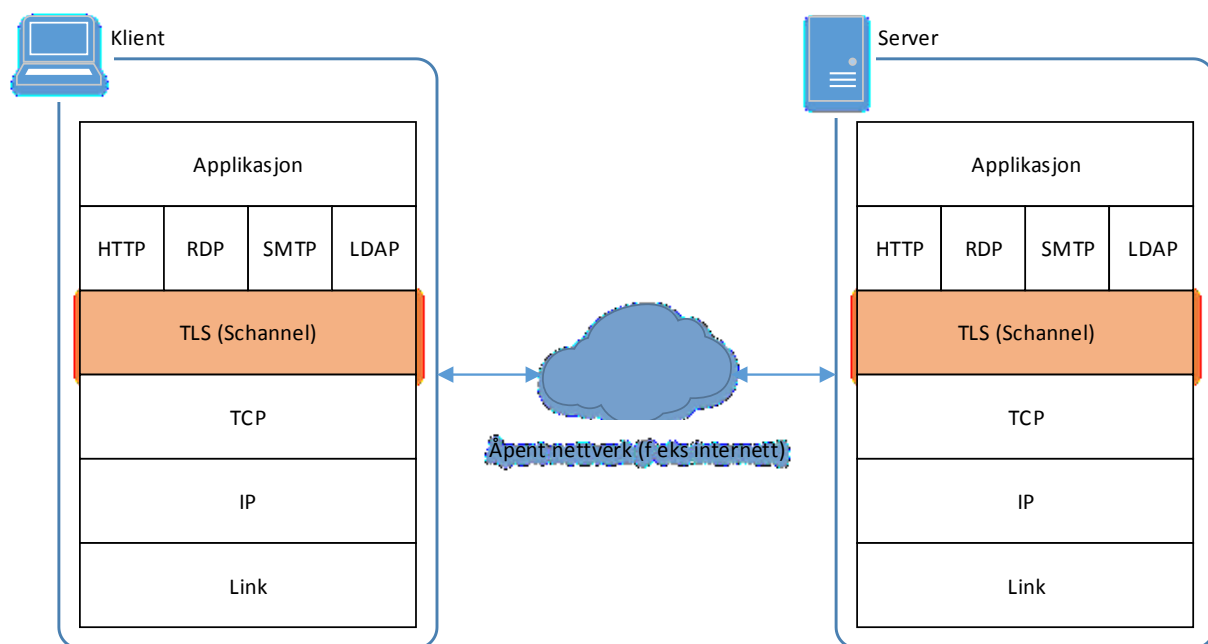
Kontaktpunkt for denne veiledningen er si@nsm.stat.no. Vennligst bruk veiledningens navn som emne. Kommentarer og innspill mottas med takk.

Dersom brukere av denne veiledningen oppdager offentlige nettsteder som er inkompatible med NSMs anbefalte tiltak, rådes brukeren å gjøre disse nettstedene oppmerksomme på denne veiledningen.

2 Bakgrunn

Figur 1 viser plasseringen til TLS i TCP/IP-protokollstakken i et Windows system, hvor implementasjonen av TLS heter *Secure Channel (Schannel)* [7,8]. Foruten om å kunne tilby sikker HTTP (HTTPS) mellom en nettleser og en web-server, kan TLS også brukes av andre applikasjonslagsprotokoller, som f.eks. *Remote Desktop Protocol (RDP)*, *Lightweight Directory Access Protocol (LDAP)* og *Simple Mail Transfer Protocol (SMTP)*. Dette betyr, med andre ord, at TLS kan brukes til å sikre kommunikasjonen for en rekke ulike applikasjoner.

I fortsettelsen av kapittelet gis det først en kort forklaring på hvordan TLS fungerer. Deretter beskriver vi hvordan en klient og en server velger protokoll og kryptografiske algoritmer ved etablering av en TLS-sesjon.



Figur 1 Kommunikasjon mellom klient og server ved bruk av TLS over et åpent nettverk. Figuren viser også TLS sin plassering i TCP/IP-protokollstakken i et Windows system.

2.1 Hvordan TLS fungerer

Denne seksjonen gir en overordnet beskrivelse av hvordan TLS-protokollene fungerer. Se [5,7] for detaljerte beskrivelser.

Bruken av TLS-protokollene deles opp i to hovedfaser:

1. Etablering av en sikker forbindelse mellom klient og server.
2. Utveksling av applikasjonsdata over den sikre forbindelsen.

En rekke aktiviteter gjennomføres ved etablering av en sikker forbindelsen, blant annet:

- 1a. Valg av protokoll og kryptografiske algoritmer.
- 1b. Autentisering av server og eventuelt klient basert på sertifikat og kryptografisk informasjon.
- 1c. Utveksling av kryptografiske parametere for generering av kryptografiske nøkler.

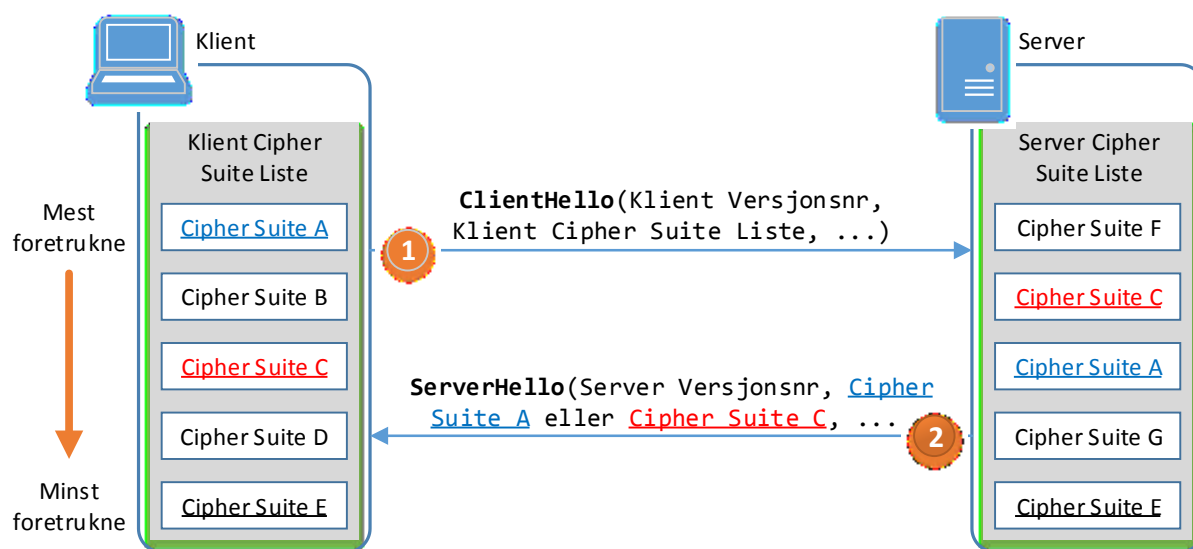
- 1d. Generering av kryptografiske nøkler, hvor nøklene brukes for konfidensialitets- og integritetsbeskyttelse av applikasjonsdataene som utveksles i fase 2.

Etter at en sikker forbindelse har blitt etablert, kan klienten og serveren utveksle applikasjonsdata på en sikker måte.

Siden fokuset i dette dokumentet er på bruken av protokoller og kryptografiske algoritmer i TLS, vil vi i neste seksjon se nærmere på aktiviteten «1a. Valg av protokoll og kryptografiske algoritmer».

2.2 Valg av TLS-protokoll og kryptografiske algoritmer

Figur 2 beskriver hvordan TLS velger protokoll og kryptografiske algoritmer ved etablering av en sikker forbindelse mellom klient og server. I fortsettelsen av seksjonen beskrives først valg av TLS-protokoll, deretter valg av kryptografiske algoritmer.



Figur 2 Detaljert beskrivelse av aktiviteten «1a. Valg av protokoll og kryptografiske algoritmer». Understreket tekst angir cipher suiter (navngitt kombinasjon av kryptografiske algoritmer) som både klienten og serveren støtter. Blå tekst og rød tekst angir henholdsvis klientens og serverens foretrukne cipher suite blant cipher suitene som støttes av begge parter.

2.2.1 Valg av TLS-protokoll

I **Figur 2** ser vi at det første som skjer er at klienten sender en **ClientHello** melding til serveren. Denne meldingen inneholder **Klient Versjonsnr**, som er versjonsnummeret til TLS-protokollen med det høyeste versjonsnummeret som klienten støtter. Versjonsnumrene 3.1, 3.2 og 3.3 brukes for henholdsvis TLS 1.0, 1.1 og 1.2.

Det neste som skjer er at serveren svarer med en **ServerHello** melding. Denne meldingen inneholder **Server Versjonsnr**, som vil ha følgende verdi:

- Hvis serveren støtter protokollen som klienten ønsker å bruke, vil den sende det **samme versjonsnummeret** tilbake til klienten.
- Hvis serveren kun støtter protokoller med **lavere versjonsnummer**, vil den sende det høyeste av de lavere versjonsnumrene tilbake til klienten.

Et eksempel kan være at klienten ønsker å bruke TLS 1.2, mens serveren kun har støtte for protokoller opp til TLS 1.1. Hvis klienten ikke støtter TLS 1.1, vil den sende en feilmelding til serveren og stenge forbindelsen.

Hvis serveren kun støtter protokoller med **høyere versjonsnummer**, for eksempel fordi den ikke har implementert protokoller med lavere versjonsnummer eller fordi den ikke er konfigurert for å bruke de, vil den sende en feilmelding til klienten og stenge forbindelsen.

NSM anbefaler å **sette protokollbegrensninger på både klient- og server-siden**. Ved å sette slike begrensninger, kan vi **hindre bruk av svake protokoller**. Se neste kapittel for NSMs anbefalinger for hvilke TLS-protokoller nyere Windows-versjoner bør støtte.

2.2.2 Valg av kryptografiske algoritmer for bruk med TLS

ClientHello meldingen i **Figur 2** inneholder også en **Klient Cipher Suite Liste**, som angir de *cipher suitene* som støttes av klienten. En *cipher suite* er en navngitt kombinasjon av tre typer kryptografiske algoritmer:

- En algoritme for **nøkkelutveksling** som brukes for beskyttelse av kryptografiske parametere, som igjen brukes ved generering av kryptografiske nøkler. Denne typen algoritme er asymmetrisk og har bra ytelse på små mengder data.
- En algoritme for **kryptering** som brukes for konfidensialitetsbeskyttelse av applikasjonsdata. Denne typen algoritme er symmetrisk og har bra ytelse på større mengder med data.
- En algoritme for **meldingsautentisitet** som brukes for integritetsbeskyttelse av applikasjonsdata. Denne typen algoritme er asymmetrisk og har bra ytelse på små mengder data.

Et eksempel på en *cipher suite* er *TLS_RSA_WITH_AES_256_CBC_SHA256* og navnet på denne *cipher suiten* angir følgende:

- TLS er protokollversjonen.
- RSA er nøkkelutvekslingsalgoritmen.
- AES er algoritmen for kryptering av applikasjonsdata og brukes i CBC-modus med 256-bit nøkkel.
- SHA256 er algoritmen for integritetsbeskyttelse av applikasjonsdata.

Listen som klienten sender til serveren er sortert i prioritert rekkefølge fra den mest til den minst foretrukne *cipher suiten*. I **Figur 2** kan vi se at *Cipher Suite A* er klientens mest foretrukne, mens *Cipher Suite E* er den minst foretrukne.

Basert på klientens liste og serverens egen liste, vil serveren velge en *cipher suite* som støttes av begge parter. Hvis klientens liste ikke inneholder noen *cipher suiter* som serveren støtter, vil serveren sende en feilmelding og stenge forbindelsen.

I **Figur 2** kan vi se at *Cipher Suite A*, *Cipher Suite C* og *Cipher Suite E* støttes av både klient og server. Basert på disse *cipher suitene* kan serveren velge *cipher suite* på én av to måter:

1. Serveren velger den *cipher suiten* som foretrekkes av klienten eller
2. Serveren velger den *cipher suiten* som foretrekkes av den selv.

I **Figur 2** ser vi at servere som implementerer punkt 1 vil velge *Cipher Suite A*, mens servere som implementerer punkt 2 vil velge *Cipher Suite C*.

Schannel implementerer punkt 2 i listen over, dvs Schannel velger den første *cipher suite* i serverens liste som støttes av både klient og server.

I det tilfellet hvor både klient og server kjører Windows operativsystem, vil Schannel sin *cipher suite* liste brukes av begge parter.

Hvis disse listene ikke har blitt endret av administrator, vil både klient og server bruke Schannel sin standardliste. Se [9] for mer informasjon. Denne standardlisten er ikke sortert i prioritert rekkefølge fra de sterkeste til de svakeste *cipher suitene*.

Dessverre kan standardkonfigurasjonen av Schannel i praksis medføre at Schannel velger en *cipher suite* som er relativt langt nede på klientens liste, og da gjerne en *cipher suite* som ikke har støtte for *Perfect Forward Secrecy*¹ [10].

Botemiddelet for dette er å omkonfigurere klientens og serverens *cipher suite* lister ved å fjerne svake *cipher suiter* og ved å endre rekkefølgen slik at sterke *cipher suiter* kommer høyt på listene.

NSM anbefaler at systemeier **omkonfigurerer *cipher suite* listene på både klienter og servere**. Se neste kapittel for NSMs anbefalinger for hvilke *cipher suiter* som bør brukes for Windows 7 SP1, Windows server 2008 R2 og senere versjoner, og hvilke *cipher suiter* som bør lukes ut.

¹ Hovedregelen er at Perfect Forward Secrecy (også kjent som Forward Secrecy) kan oppnås ved bruk av DHE (ephemeral Diffie-Hellman) eller ECDHE (elliptic-curve ephemeral Diffie-Hellman) som nøkkelutvekslingsalgoritme. Det finnes unntak fra denne regelen, men en drøfting av disse faller utenfor omfanget til dokumentet.

3 Anbefalinger

Dette kapittelet inneholder NSMs anbefalinger for valg av TLS-protokoller og *cipher suiter* for Windows klienter og servere i offentlig forvaltning.

Relevante Windows-versjoner er Windows 7, Windows Server 2008 R2 og senere versjoner.

Se [11,12,13] for andre anbefalinger rundt bruken av TLS, utløpsdato for enkelte kryptografiske algoritmer og anbefalinger rundt minimum nøkkellengde for ulike kryptografiske algoritmer.

NSM gjør oppmerksom på at noe tredjepart-programvare **bruker andre TLS-implementasjoner enn Microsoft Schannel**. Dette gjelder f.eks. Java RunTime, Google Chrome, Mozilla Firefox og Opera. Sikkerheten i slik programvare blir derfor ikke endret ved bruk av NSMs anbefalinger.

Seksjon 3.1 inneholder NSMs anbefalinger for hvilke **TLS-protokoller** som bør velges.

Seksjon 3.2 og **Vedlegg A** inneholder NSMs anbefalinger for hvilke ***cipher suiter*** som bør velges.

Vedlegg B beskriver eksempler på konfigurasjonsfiler og script som aktiverer anbefalte TLS-protokoller og *cipher suiter* i Schannel på både klienter og servere.

3.1 Anbefalte TLS-protokoller

Listen under angir NSMs anbefalte TLS-protokoller i prioritert rekkefølge for Windows 7 SP1, Windows Server 2008 R2 og senere versjoner.

Prioritet 1. TLS 1.2

Prioritet 2. TLS 1.1

Prioritet 3. TLS 1.0

NSM fraråder bruk av SSL-protokoller. Se **Kapittel 1** for mer informasjon.

Av *cipher suitene* som anbefales under, er det bare TLS 1.2 som bruker de aller sterkeste *cipher suitene*.

3.2 Anbefalte cipher suiter

De **anbefalte** *cipher suitene* har blitt delt inn i to kategorier, **foretrukne** og **akseptable** *cipher suiter*, mens *cipher suiter* som ikke anbefales, har blitt plassert i kategorien **ikke-anbefalte** *cipher suiter*.

Under er kriteriene som avgjør hvilken kategori en *cipher suite* havner i.

Foretrukne *cipher suiter* (gir best sikkerhet):

1. *Cipher suiter* som bruker AES, SHA-2 og/eller GCM og som har *Perfect Forward Secrecy* (PFS).

Akseptable *cipher suiter* deles inn i fire underkategorier her angitt i prioritert rekkefølge:

1. *Cipher suiter* som bruker AES, SHA-1 og som har *Perfect Forward Secrecy*.
2. *Cipher suiter* som bruker AES og SHA-2.
3. *Cipher suiter* som bruker AES og SHA-1.
4. *Cipher suiter* som bruker 3DES og SHA-1.

Ikke-anbefalte *cipher suiter*:

1. *Cipher suiter* som bruker enten RC4, DES, MD5 eller NULL-kryptering.

Se **Vedlegg A** for en detaljert liste over de anbefalte *cipher suitene*. Denne listen er delt i to:

Del 1. Tabell 2 angir **foretrukne** *cipher suiter*.

Del 2. Tabell 3 angir **akseptable** *cipher suiter*.

Listen over NSMs anbefalte *cipher suiter* er basert på [9], som angir både *cipher suiter* som er støttet og ikke støttet «ut av boksen» av Schannel i Windows 7 og Windows Server 2008 R2.

De to siste *cipher suitene* i **Tabell 3** (underkategorien «3DES og SHA-1») har blitt lagt til for bakoverkompatibilitet. Windows servere trenger ofte disse for å kommunisere med Windows XP klienter, siden Windows XP ikke har «ut av boksen» støtte for AES-baserte ciphre. 3DES algoritmen regnes for å være ganske sikker, selv om den ikke tilbyr så god sikkerhet som nøkkellengdene (112-bit eller 168-bit) skulle tilsi [14]. **Hvis virksomheten ikke har servere som trenger å kommunisere med Windows XP eller hvis man ønsker høyere sikkerhet, kan de to siste *cipher suitene* fjernes fra listen.**

En rekke av *cipher suitene* i [9] har blitt klassifisert som **ikke-anbefalte** fordi de anvender kryptografiske algoritmer som ikke regnes som sikre nok etter dagens standard. **Tabell 1** forklarer hvorfor disse algoritmene har fått en slik klassifisering.

Algoritme	Type algoritme	Hvorfor ikke anbefalt
NULL	Kryptering	NULL betyr ingen kryptering.
RC4	Kryptering	RC4 har en rekke svakheter og regnes ikke som sikker nok etter dagens standard [14,15,16].
DES	Kryptering	DES har en rekke svakheter og regnes ikke som sikker nok etter dagens standard [14].
MD5	Meldingsautentisitet	MD5 er en algoritme som blir mye brukt, men som ikke anses for å være sikker nok [14].

Tabell 1 Kryptografiske algoritmer som har ført til klassifisering av *cipher suiter* som ikke-anbefalte.

Hvis virksomheten anvender Windows Server 2003 eller 2008, anbefaler NSM oppgradering til en nyere versjon av Windows så fort som mulig. Dette gjelder også Windows XP klienter.

Windows Server 2008 støtter kun TLS 1.0 *cipher suitene* i **Tabell 3**. Til sammenligning støtter Windows Server 2003 kun de 3DES-baserte *cipher suitene* i **Tabell 3** «ut av boksen», fordi den ikke har støtte for AES-baserte ciphre.

Hvis det ikke er mulig å bytte ut en Windows Server 2003 med et nyere Windows server-operativsystem, bør alternativt *hotfixen* KB948963 [17] installeres for å få støtte for de to *cipher suitene* *TLS_RSA_WITH_AES_128_CBC_SHA* og *TLS_RSA_WITH_AES_256_CBC_SHA*.

Vedlegg A Anbefalte cipher suiter (detaljert)

Vedlegg A angir en detaljert liste over anbefalte *cipher suiter* for Windows 7 SP1, Windows Server 2008 R2 og senere versjoner.

A.1 DEL 1 FORETRUKNE CIPHER SUITER

Del 1 av listen angir *cipher suiter* i kategorien «foretrukne».

		Cipher suite	Protokoller
Foretrukne cipher suiter	AES, SHA-2 og/eller GCM og Perfect Forward Secrecy	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384_P521	TLS 1.2
		TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384_P384	TLS 1.2
		TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384_P521	TLS 1.2
		TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384_P521	TLS 1.2
		TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384_P384	TLS 1.2
		TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384_P384	TLS 1.2
		TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384_P256	TLS 1.2
		TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256_P521	TLS 1.2
		TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256_P384	TLS 1.2
		TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256_P256	TLS 1.2
		TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256_P521	TLS 1.2
		TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256_P521	TLS 1.2
		TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256_P384	TLS 1.2
		TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256_P384	TLS 1.2
		TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256_P256	TLS 1.2
		TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256_P256	TLS 1.2

Tabell 2 **Foretrukne** cipher suiter.

A.2 DEL 2 AKSEPTABLE CIPHER SUITER

Del 2 av listen angir cipher suiter i kategorien «akseptable».

		Cipher suite	Protokoller
Akseptable cipher suiter	AES, SHA-1 og Perfect Forward Secrecy	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA_P521	TLS 1.0, TLS 1.1, TLS 1.2
		TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA_P521	TLS 1.0, TLS 1.1, TLS 1.2
		TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA_P384	TLS 1.0, TLS 1.1, TLS 1.2
		TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA_P384	TLS 1.0, TLS 1.1, TLS 1.2
		TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA_P256	TLS 1.0, TLS 1.1, TLS 1.2
		TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA_P256	TLS 1.0, TLS 1.1, TLS 1.2
		TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA_P521	TLS 1.0, TLS 1.1, TLS 1.2
		TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA_P521	TLS 1.0, TLS 1.1, TLS 1.2
		TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA_P384	TLS 1.0, TLS 1.1, TLS 1.2
		TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA_P384	TLS 1.0, TLS 1.1, TLS 1.2
		TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA_P256	TLS 1.0, TLS 1.1, TLS 1.2
		TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA_P256	TLS 1.0, TLS 1.1, TLS 1.2
	AES og SHA-2	TLS_DHE_DSS_WITH_AES_256_CBC_SHA256	TLS 1.2
		TLS_RSA_WITH_AES_256_CBC_SHA256	TLS 1.2
		TLS_DHE_DSS_WITH_AES_128_CBC_SHA256	TLS 1.2
		TLS_RSA_WITH_AES_128_CBC_SHA256	TLS 1.2
	AES og SHA-1	TLS_DHE_DSS_WITH_AES_256_CBC_SHA	TLS 1.0, TLS 1.1, TLS 1.2
		TLS_RSA_WITH_AES_256_CBC_SHA	TLS 1.0, TLS 1.1, TLS 1.2
		TLS_DHE_DSS_WITH_AES_128_CBC_SHA	TLS 1.0, TLS 1.1, TLS 1.2
		TLS_RSA_WITH_AES_128_CBC_SHA	TLS 1.0, TLS 1.1, TLS 1.2
	3DES og SHA-1	TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA	TLS 1.0, TLS 1.1, TLS 1.2
		TLS_RSA_WITH_3DES_EDE_CBC_SHA	TLS 1.0, TLS 1.1, TLS 1.2

Tabell 3 Akseptable cipher suiter.

Vedlegg B Konfigurasjon av Schannel

Vedlegg B beskriver eksempler på konfigurasjonsfiler og script for Schannel på Windows 7 SP1, Windows Server 2008 R2 og senere versjoner.

Arkivfilen **Schannel.zip** er tilgjengelig fra [18]. Denne arkivfilen inneholder en rekke filer, som beskrives i fortsettelsen.

B.1 KONFIGURASJON VHA WINDOWS REGISTRY EDITOR

Konfigurasjonsfilen **U03SchannelPolicy.reg** lastes ved hjelp av *Registry Editor*. Denne filen konfigurerer Schannel for bruk av anbefalte TLS-protokoller og *cipher suiter*.

Denne registerfilen er tiltenkt frittstående maskiner som **ikke administreres vha Group Policy**.

B.2 KONFIGURASJON VHA POWERSHELL

Arkivfilen inneholder også en PowerShell-modul **NSM TLS Tools.psm1** og et PowerShell-script **Apply NSM Schannel Policy.ps1** for *enkel Schannel-manipulasjon*.

Modulen lastes av PowerShell vha følgende kommando i et **elevert** konsollvindu:

```
Import-Module '.\NSM TLS Tools.psm1' -Force
```

Ved å laste modulen får man tilgang til følgende funksjoner:

- **Test-SchannelPolicy**
- **New-SchannelUndoPolicy**
- **Set-SchannelPolicy**
- **Undo-SchannelPolicy**
- **New-SchannelXmlPolicy**

Disse funksjonene beskrives i påfølgende avsnitt.

PowerShell-scriptet **Apply NSM Schannel Policy.ps1** tar i bruk funksjoner fra modulen for å utføre følgende oppgaver:

1. Tester om det er samsvar mellom maskinens Schannel konfigurasjon og NSMs Schannel konfigurasjon vha funksjonen **Test-SchannelPolicy**.
 - Hvis det er samsvar, så avsluttes scriptet.
2. Tar backup av maskinens Schannel konfigurasjon vha funksjonen **New-SchannelUndoPolicy**.
 - Funksjonen produserer backup i form av registerfilen **SchannelPolicy.undo.reg**.
 - Denne registerfilen kan brukes til å **tilbakestille maskinen til den opprinnelige Schannel konfigurasjon**, hvis f.eks bruk av NSMs Schannel konfigurasjon ikke gir ønsket resultat.
3. Konfigurer Schannel ved bruk av NSMs Schannel konfigurasjon (**U03SchannelPolicy.reg**) vha funksjonen **Set-SchannelPolicy**.
4. Tester på ny om det er samsvar mellom maskinens Schannel konfigurasjon og NSMs Schannel konfigurasjon vha funksjonen **Test-SchannelPolicy**.

B.2.1 Funksjonen Test-SchannelPolicy

Funksjonen **Test-SchannelPolicy** tester om maskinen bruker svake TLS/SSL-protokoller eller *cipher suiter*. Den kan også teste om maskinens Schannel konfigurasjon samsvarer med NSMs Schannel konfigurasjon.

Kommandoen **Get-Help Test-SchannelPolicy -Full** gir informasjon om hvordan funksjonen brukes.

Funksjonen har blitt testet mot *cipher suiter* som finnes i Schannel og er ikke garantert å fungere for andre *cipher suiter*.

B.2.2 Funksjonen New-SchannelUndoPolicy

NSM anbefaler at det tas backup av Schannel registerverdier før endringer gjøres ved bruk av registerfilen eller AD Group Policy.

Funksjonen **New-SchannelUndoPolicy** kan brukes til dette formål.

Kommandoen **Get-Help New-SchannelUndoPolicy -Full** gir informasjon om hvordan funksjonen brukes.

Funksjonen produserer enten en registerfil (f eks **SchannelPolicy.undo.reg**) eller en **XML**-fil (f eks **SchannelPolicy.undo.xml**).

Registerfilen kan brukes for å tilbakestille Windows registeret til de opprinnelige verdiene, mens **XML**-filen kan brukes for å lage en GPO som tilbakestiller Windows registeret på flere maskiner.

B.2.3 Funksjonen Set-SchannelPolicy

Funksjonen **Set-SchannelPolicy** brukes til å konfigurere Windows registeret på frittstående maskiner med en Schannel policy i **.reg**-format (f eks **U03SchannelPolicy.reg**).

Kommandoen **Get-Help Set-SchannelPolicy -Full** gir informasjon om hvordan funksjonen brukes.

B.2.4 Funksjonen Undo-SchannelPolicy

Funksjonen **Undo-SchannelPolicy** brukes til å tilbakestille Windows registeret på frittstående maskiner til en tidligere Schannel policy. Funksjonen bruker en registerfil (f eks **SchannelPolicy.undo.reg**) til å utføre dette.

Kommandoen **Get-Help Undo-SchannelPolicy -Full** gir informasjon om hvordan funksjonen brukes.

B.2.5 Funksjonen New-SchannelXmlPolicy

Hvis virksomheten skal konfigurere Schannel på flere maskiner, bør Active Directory Group Policy brukes for å sende konfigurasjonen til de aktuelle maskinene i form av et Group Policy Object (GPO).

Funksjonen **New-SchannelXmlPolicy** kan brukes for å lage en GPO for Schannel.

Kommandoen **Get-Help New-SchannelXmlPolicy -Full** gir informasjon om hvordan funksjonen brukes.

Basert på en registerfil (f eks **U03SchannelPolicy.reg**) lager funksjonen en **XML**-fil (f eks ved navn **SchannelPolicy.xml**).

Deretter må systemadministrator lage en tom GPO på en DC og åpne denne for redigering i ADs Group Policy Editor (GPEdit).

Deretter kopieres og limes innholdet i f eks **SchannelPolicy.xml** inn i GPEdit-noden:

Computer Configuration\Preferences\Windows Settings\Registry

Ønskede *cipher suiter* kan **alternativt** settes ved bruk GPO-settingen **SSL Cipher Suite Order** i GPEdit-noden **Computer Configuration\Policies\Administrative Templates\Network\SSL Configuration Settings**.

Det gjøres som følger:

Step 1. Åpne Group Policy Management Editor.

Step 2. Naviger fram til **Computer Configuration\Policies\Administrative Templates\Network\SSL Configuration Settings\SSL Cipher Suite Order**.

Step 3. Høyreklikk på **SSL Cipher Suite Order** og velg **Edit**.

Step 4. Sett innstillingen til **Enabled**.

Step 5. Lim inn strengen med *cipher suiter* i tekstboksen **SSL Cipher Suites**. Denne strengen skal være sortert i prioritert rekkefølge og de enkelte *cipher suite*ne skal være adskilt med komma.

Step 6. Trykk **OK**.

*Hvis strengen som limes inn er **lengere enn 1023 tegn**, vil GPO-innstillingen fortsatt stå som **Not Configured** og ikke som **Enabled** etter at man har trykket **OK**.*

Grunnen er at tekstboksen **SSL Cipher Suites** i det grafiske brukergrensesnittet til *Group Policy Management Editor* har en begrensning på 1023 tegn. **GPEdit vil ikke gi noen feilmeldingen hvis strengen er lengere enn 1023 tegn.**

NSM anbefaler bruk av scriptet **New-SchannelXmlPolicy** dersom Schannel skal konfigureres ved bruk av Group Policy og strengen er mer enn 1023 tegn eller nær 1023 tegn.

Vedlegg C Referanser

- [1] Netscape Communications Corporation, *The SSL Protocol*, 1995.
- [2] Netscape Communications Corporation, *The SSL 3.0 Protocol*, 1996.
- [3] Internet Engineering Task Force, *RFC 2246 – The TLS Protocol Version 1.0*, 1999.
- [4] Internet Engineering Task Force, *RFC 4346 – The Transport Layer Security (TLS) Protocol Version 1.1*, 2006.
- [5] Internet Engineering Task Force, *RFC 5246 – The Transport Layer Security (TLS) Protocol Version 1.2*, 2008.
- [6] Internet Engineering Task Force, *RFC 6176 – Prohibiting Secure Sockets Layer (SSL) Version 2.0*, 2011.
- [7] Microsoft, *TLS/SSL Technical Reference*, [http://technet.microsoft.com/en-us/library/cc784149\(v=ws.10\).aspx](http://technet.microsoft.com/en-us/library/cc784149(v=ws.10).aspx), [Funnet: 2014-02-06].
- [8] Microsoft, *Secure Channel*, [http://msdn.microsoft.com/en-us/library/windows/desktop/aa380123\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/aa380123(v=vs.85).aspx), [Funnet: 2014-02-10].
- [9] Microsoft, *Cipher Suites in Schannel*, [http://msdn.microsoft.com/en-us/library/windows/desktop/aa374757\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/aa374757(v=vs.85).aspx), [Funnet: 2014-02-10].
- [10] V. Bernat, *SSL/TLS & Perfect Forward Secrecy*, <http://vincent.bernat.im/en/blog/2011-ssl-perfect-forward-secrecy.html>, [Funnet: 2014-03-04].
- [11] Bundesamt für Sicherheit in der Informationstechnik, *Technische Richtlinie TR-02102-2 – Kryptographische Verfahren: Empfehlungen und Schlüssellängen – Teil 2 – Verwendung von Transport Layer Security (TLS)*, Version 2014-01, 2014.
- [12] NIST, *Guidelines for the Selection, Configuration, and Use of Transport Layer Security (TLS) Implementations*, NIST Special Publication 800-52, Revision 1, 2014.
- [13] BlueKrypt, *Cryptographic Key Length Recommendation*, <http://www.keylength.com/>, [Funnet: 2014-03-03].
- [14] ENISA, *Algorithms, Key Sizes, and Parameter Report – 2013 Recommendations*, Version 1.0, 2013.
- [15] K. Patterson, *On the Security of RC4 in TLS and WPA*, <http://www.isg.rhul.ac.uk/tls/>, [Funnet: 2014-02-20].
- [16] M. Green, *Attack of the week: RC4 is kind of broken in TLS*, <http://blog.cryptography-engineering.com/2013/03/attack-of-week-rc4-is-kind-of-broken-in.html>, [Funnet: 2014-02-20].
- [17] Microsoft, *An update is available to add support for the TLS_RSA_WITH_AES_128_CBC_SHA AES128-SHA and the TLS_RSA_WITH_AES_256_CBC_SHA AES256-SHA AES cipher suites in Windows Server 2003*, <http://support.microsoft.com/kb/948963>, [Funnet: 2014-02-11].
- [18] NSM, *Eksempler på konfigurasjonsfiler og script for Windows TLS (Schannel.zip)*, <https://nsm.stat.no/publikasjoner/regelverk/veiledninger/veiledning-for-systemteknisk-sikkerhet/>, [Funnet: 2014-12].

Vedlegg D Dokumenthistorie

2014-02-05	Dokumentet ble opprettet.
2014-04-01	Første versjon ferdig. Begrenset sirkulasjon utenfor NSM.
2014-04-03	Sendt ut på høring i Teknologi-avdelingen i NSM.
2014-12-12	Begrenset publisering.
2015-03-16	Publisering på nett.