HANTERING AV KLIENTER MED MJUKVARA SOM GÅTT EOL V2.0

Historik

Datum	Version	Av	Förändring
2014-05-27	0.1	Petrus Andersson	Skapat dokumentstruktur
2014-06-05	0.11	Petrus Andersson	Uppdaterat dokument
2014-08-20	0.12	Petrus Andersson	Uppdaterat dokument
2014-08-22	013	Petrus Andersson	Uppdaterat dokument
2014-08-26	0.14	Petrus Andersson	Uppdaterat dokument
2014-08-29	0.15	Petrus Andersson	Uppdaterat dokument
2014-09-02	0.16	Petrus Andersson	Uppdaterat dokument
2014-09-23	0.17	Petrus Andersson	Uppdaterat dokument
2014-09-25	0.9	Olof Mårtensson	Satt till version 0.9
2014-12-03	0.91	Olof Mårtensson	Språkliga och strukturella justeringar efter granskning
2014-12-20	1.0	Olof Mårtensson	Version 1.0
2019-02-04	2.0	André Arvidsson	Fullständig omarbetning och uppdatering av dokumentet samt namnbyte av dokumentet för att bättre återspegla innehållet.

Innehåll

1. Dokumentinformation

1.1 Syfte

Detta dokument agerar referensdokument och tillhör dokumenttypen standarder inom eKlient. En standard definierar ett gemensamt språk mellan aktörer och underlättar det dagliga livet för intressenter.
Dokumentet definierar egenskaper, krav och instruktioner som stöd i den dialog som förs mellan eKlients medlemmar och deras driftspartners/driftorganisationer och applikationsleverantörer.

Syftet med detta dokument är att beskriva hur man kan arbeta med livscykelhantering av mjukvara, varför det är viktigt och hur man kan agera när man tvingas ha kvar icke supporterad mjukvara i sin vardagliga driftsmiljö.

Dokumentets avsikt är att vara till stöd för den lokala/regionala förvaltningen av en centraliserad klientplattform för att leverera en stabil och säker leverans till verksamheten.

1.2 Målgrupp

- Lokal förvaltningsorganisation för klientplattform
- Intressenter och medlemsorganisationer i eKlient
- Aktörer på marknaden med intresse att följa eKlient
- Partners till medlemsorganisationer i eKlient

Dokumentet publiceras publikt på Ineras hemsida under www.inera.se/eKlient för att säkerställa att det är tillgängligt för en bredare publik.

1.3 Revision

Målet är att revidera detta dokument årligen.
Senaste revisionen skedde 2019-02-04.

1.4 Styrande principer

Styrande principer för denna standard är:

- Använd vedertagna standardlösningar som är globalt använda
- Behörighetstilldelning efter principen ej mer än vad du behöver
- Information ska skyddas från förvanskning och obehörig åtkomst eller spridning
- Kontrollerad livscykelhantering av ingående komponenter

1.5 Referenser

Exempelvis närliggande standards dokument i eKlient
referenser till direkta fakta eller ställningstaganden i standarden

Dokumentnamn	Sökväg
eKlient - Kravbibliotek	Inera.se/eKlient
eKlient riktlinjer för hantering mobila enheter	eKlient SharePoint för medlemmar

1.6 Termer och begrepp

T nr	*begrepp/term*	*Beskrivning*
T1	*EoL*	*End Of Life, att den uttalade livscykeln är slut.*
T2	*Skadlig kod*	*är ett samlingsbegrepp för datorprogram som installeras på en dator eller ett datornätverk utan administratörens samtycke*
T3	*Trojan*	*En trojansk häst eller trojan är ett datorprogram som utger sig för att vara till nytta eller nöje, men som gör något annat när det lurat en användare att installera eller köra det. Programmet kan till exempel spionera på användaren, göra betalningar i användarens namn, skicka skräppost eller attackera andra datorer.*
T4	*DoS/DDoS*	*Denial of Service, en attack mot ett datasystem i syfte att hindra normal användning av systemet. De vanligaste attacktyperna är överbelastningsangrepp.*
T5	*Ransomware*	En typ av skadlig mjukvara som är skapad för att hindra tillgång till ett datorsystem eller information till dess att en summa pengar betalats till angriparen.
T6		

1.7 Avgränsningar

Standarden kommer vara avgränsad till att beskriva risker och åtgärder för enheter som av olika skäl tvingas nyttja EoL operativsystem eller EoL mjukvara utan att nämna några specifika versionsnummer.

2. Livscykelhantering av mjukvara

I dagens vardag där tekniken utvecklas i en snabbare takt än någonsin förr och där information snabbt kan förflyttas från den ena endan av världen till den andra är det av yttersta vikt att säkerställa att alla åtgärder som kan vidtas för att säkra informationen har vidtagits. En del av det arbetet är att arbeta med livscykelhantering av den mjukvara som används och det underliggande operativsystemet så att man uppnår en så säker IT-arbetsplats som möjligt.

Det är därför väldigt viktigt att skapa sig en tydlig bild över hur mjukvarorna kommer att förändras över tid, kanske inte nödvändigtvis innehållsmässigt men med vilka cykler det kommer att krävas uppdateringar så att dessa aktiviteter går att planera in i det kontinuerliga arbetet med IT-arbetsplatsen.

Med de förändrade utvecklingsmetoder som idag är rådande i branschen är det allt mer vanligt med ”små” snabba uppdateringar förhållandevis regelbundet som kompletteras med några större som händer en eller ett par gånger om året. Det är en förhållandevis stor skillnad mot hur det såg ut för bara något år sedan där det kunde dröja flera år innan en mjukvara eller ett operativsystem uppgraderades.

Om man slår samman den ökade frekvensen på uppgraderingar av mjukvara samt den ökade komplexitet som finns på lokalt installerad mjukvara och samtidigt ser till de digitala hot som finns i omlopp så inser man ganska snabbt att det är en enorm utmaning att arbeta med att skydda den hanterade informationen från att nås eller förvanskas av obehöriga eller till och med låsas in så att inte ens de rättmätiga ägarna kan få tillgång till den.

Det är bland annat av dessa skäl som det är ytterst viktigt att arbeta med en aktiv livscykelhantering av sin mjukvara.

Det går att bryta ner livscykelhanteringen i olika områden:

- Operativsystem, den underliggande plattformen som krävs för att mjukvaran ska fungera.
- Drivrutiner och applikationer som direkt stödjer hårdvaran
- Tredjepartsmjukvara som krävs för att den huvudsakliga mjukvaran ska fungera
- Mjukvaran användarna arbetar med

Det som är generellt är att ovan uppdelning gäller oavsett om det handlar om traditionella IT-arbetsplatser som baseras på PC eller om det är mobila enheter som använder iOS eller

Android. Det är i slutändan informationen som finns och hanteras på dessa enheter som ska skyddas.

2.1 Operativsystem

2.1.1 Microsoft Windows

Microsoft eftersträvar hela tiden att vara tydliga med hur livscykeln för deras olika Operativsystem ser ut över tid, när supporten upphör och produkten går EoL[1]. Att förlänga supporten efter denna tidpunkt är både extremt kostsamt och tidsödande men kan ibland vara nödvändigt. Man bör i god tid innan ett operativsystem är EoL säkerställt att man kan uppgradera klienter till en supporterad version av operativsystem utan negativ påverkan av vare sig verksamhetsfunktionalitet eller säkerhet.

Även om Microsoft har som ambition att vara tydliga med hur hanteringen av livscykeln för operativsystemet Windows hanteras så finns det fortfarande många applikationer som idag saknar en egen livscykel och i de fall en sådan plan finns är det tyvärr inte ovanligt att den inte följer Microsofts.

Att sätta en egen tidsgräns för när man anser ett operativsystem har gått EoL är en bra lösning och redan i god tid påbörja en avvecklingsplan och även en hård gräns då klienter med EoL operativsystem inte får förekomma i miljön. Om båda dessa två tidsgränser sker innan Microsofts EOL datum uppnår man en större säkerhet och minskar administration och kostnader.

Microsoft gör i dagsläget (2019) två releaser om året av Windows 10, en release på våren och en på hösten. Sedan hösten -2018 har Microsoft i ett uttalande sagt att vår releaserna framöver kommer att ha 18 månaders support och höstreleaserna kommer att ha 30.

2.1.2 Apple macOS

Har man klienter som använder Apple macOS så är det tyvärr inte lika lätt att få en tydlig bild över operativsystemets livscykel. Apple har i dagsläget inget skriftligt uttalat som garanterar hur länge en viss version av macOS stöds med säkerhetsuppdateringar. Apple saknar även publikt tillgänglig information om när en specifik version av macOS inte längre stöds (EoL).

[1] https://support.microsoft.com/sv-se/help/13853/windows-lifecycle-fact-sheet

För information om vilka version som stöds i dagsläget kan man nå den informationen på Apples hemsida, https://support.apple.com/en-us/HT201222.

2.2 Hårdvara

Vid uppgraderingar av operativsystem är det även viktigt att säkerställa att hårdvaran och dess drivrutiner samt tillhörande mjukvara har det stöd för det nya operativsystemet.

2.3 Mobila enheter

På senare år har det blivit allt vanligare att man hanterar känsligt information även på mobila enheter med operativsystem såsom iOS och Android och det är därför även viktigt att man säkerställer att de mobila enheter som används har supporterade operativsystem under hela sin beräknade livslängd. Arbetet med mobila enheter är dock ännu mer fragmenterat och komplext. Mer information om mobila enheter hittas i dokumentet ”eKlient riktlinjer för hantering mobila enheter” som finns publicerat på eKlients SharePoint sida för medlemmar.

2.4 Verksamhetssystem

Från ett användarperspektiv är ofta frågan om vilket operativsystem som används irrelevant då det primärt är verksamhetssystemet och den information som hanteras av det som är det som det dagliga arbetet påverkas av. Det är tillgängligheten till det systemet och den informationen som är det som måste säkerställas och genom planering går det att minimera den tid som systemet behöver vara otillgängligt för användaren.

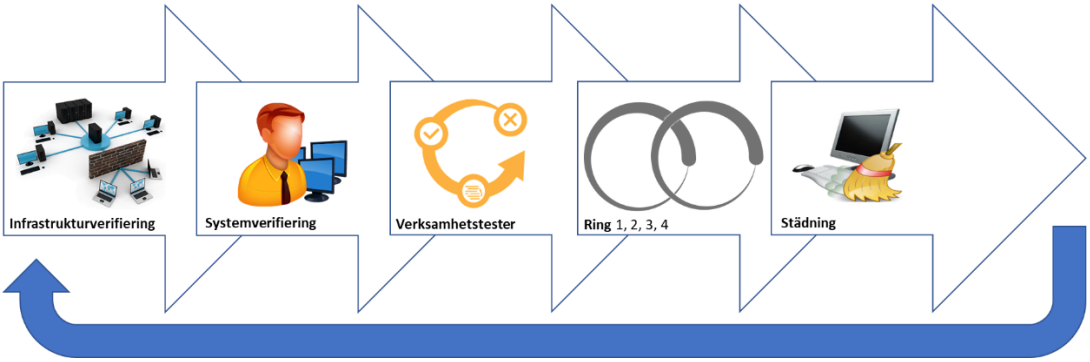
Det är av yttersta vikt att verksamhetssystem har en tydlig livscykelplan och att den planen passar ihop med det underliggande operativsystemet. Det hjälper inte att verksamhetssystem får stöd för en specifik version av ett operativsystem om samma operativsystem endast har tre månader kvar innan det når EoL. Det är därför viktigt med ett bra samarbete med tillverkaren av verksamhetssystemet och att ha en bra kravställning när nya verksamhetssystem införskaffas. Som stöd för den typen av kravställning har eKlient tillsammans med sina medlemmar arbetat fram ett dokument som finns tillgängligt på inera.se/eKlient.

2.5 Tredjepartsapplikationer

I många fall kräver verksamhetssystem olika tredjepartsapplikationer för att fungera och då är det även viktigt att säkerställa att dessa tredjepartsapplikationer passar in i den stora bilden med livscykelhantering av mjukvaran. Det bästa är att säkerställa att verksamhetssystemet inte har några krav på tredjepartsmjukvaror för att fundera, detta för att minska komplexiteten vid planering av livscykelhantering.

2.6 Process för livscykelhantering av operativsystem

Det är således en gedigen process som krävs för att få det löpande arbetet med livscykelhantering att fungera med många inblandade parter innan en uppgradering av slutanvändarens enhet kan ske.



Genom att arbeta proaktivt med kravställning och planering går det att underlätta det löpande arbetet.

2.7 Summering

Att livscykelhantera en IT arbetsplats är en komplex matris med många beroenden där samtliga komponenter behöver ha stöd för uppgraderingen för att den ska kunna genomföras smidigt.

Det räcker med att en komponent i kedjan inte fungera eller har stöd hos tillverkaren för att uppgraderingen inte ska kunna genomföras.

Det gäller därför att ha väldigt god kunskap om samtliga installerade komponenter och deras livscykler för att inte drabbas av oförutsedda förseningar vilket i många fall leder till både verksamhetsstörningar och ökade kostnader.

3. Risker

Att använda operativsystem och mjukvara som är EoL och inte längre får säkerhetsuppdateringar är en stor risk då dessa många gånger innehåller både kända och okända sårbarheter som kan ge obehöriga tillgång till systemet.

Dagens hackers är väldigt sofistikerade och använder sig av många avancerade tekniker för att få tillgång till information och det räcker med en sårbarhet i ett av systemen för att de ska ta sig in. Man kan likställa att ha IT-arbetsplatser med EoL operativsystem/mjukvara som att ha en säkerhetsdörr som ytterdörr men på baksidan av huset har man endast ett nätfönster. I ett

sådant scenario så inser man snabbt att de kriminella elementen väljer nätfönstret som ingång och inte säkerhetsdörren.

Det finns naturligtvis åtgärder man kan ta till för att minska riskerna och de diskuteras senare i dokumentet.

En av de större riskerna som blivit mer och mer uppmärksammad de senaste åren är risken att information förloras, sprids okontrollerat eller förvanskas. Just hanteringen av information är något som setts som ytterst allvarligt vilket har resulterat i att flertalet lagar och förordningar skapats i syfte att minimera riskerna för att det ska ske och i de fall det sker säkerställa att det blir konsekvenser för den part som inte vidtaget de nödvändiga åtgärderna för att säkra informationen. Den av dessa förordningar som blivit mest uppmärksammad inom Sverige är Dataskyddsförordningen[2] men det finns flera andra både inom Sverige och inom EU.

Utöver riskerna som nämns ovan så finns det även riskerna som EoL mjukvara kan skapa i den redan uppgraderade delen av miljön om man låter dessa samexistera. Då måste man även ta med i beräkningarna att de klienterna som har EoL mjukvaran kan användas för att exempelvis:

- Sprida virus, skadlig kod och trojaner
- Användas till attacker, som tex. DoS/DDoS
- Stjäla, förvanska eller göra information otillgänglig
- Botnät för exempelvis utvinnande av kryptovaluta

Det är alltså viktigt med en kontinuerlig riskbedömning av helheten när man av olika skäl tvingas ha kvar osupporterad mjukvara i sin verksamhet. I den riskbedömningen är det viktigt att man inser att det inte enbart är enheten som kör den osupporterade mjukvara som är utsatt för risker utan även allt som kan nås ifrån den och all information som finns lagrad på enheten, det inkluderar även tidigare användare och eventuella lösenord som kan finnas lagrade på enheten. För att minimera dessa risker och dess konsekvenser finns det åtgärder man kan vidta och några av dessa gås igenom senare i dokumentet.

4. Åtgärder

Att med hjälp av information skapa en medvetenhet hos användare om de risker som finns är alltid en bra början för med kunskap kommer även möjligheten att förändra betanden hos

[2] https://www.datainspektionen.se/lagar--regler/dataskyddsforordningen/

användarna. Information kan även öka förståelsen för varför det är så viktigt att uppgradera till en supporterad version av mjukvaran. Utrustade med den kunskapen så är det även lättare för användare att förstå varför IT organisationen ibland måste använda tekniska begränsningar för att minimera antalet möjliga attackvektorer som skadlig kod kan använda sig av. Exempel på åtgärder som kan vidtas listas nedan.

4.1 Inventering

Första steget av åtgärder är att säkerställa omfattningen av klienter med EoL mjukvara som finns i miljön eller ännu hellre proaktivt kartlägga vilka enheter som proaktivt behöver hanteras innan operativsystemen blir EoL. System och applikation som är ett krav för verksamheten måste inventeras och kartläggas, bland dessa kan även specialhårdvara finnas. Det kan vara nödvändigt att utöka inventering för att kartlägga omfattningen.

Det finns även stora vinster med att skaffa sig kunskap om vilka kända sårbarheter som den installerade mjukvaran har och vilken typ av information som hanteras på IT-arbetsplatserna.

4.2 Whitelisting

Whitelisting eller vitlistning på svenska innebär att man mjukvarumässigt styr vad som får exekveras på datorn. På detta sätt säkerställer man att ingen okänd kod kan exekveras. Vitlistning är en väldigt bra lösning om IT-arbetsplatsen och det som är installerat på den är någorlunda statiskt. Vid förändringar måste nämligen regelverken kring vitlistning gås igenom och sker det många förändringar blir detta en väldigt tung administrativ uppgift.

Det ska dock inte glömmas bort att även om vitlistning är väldigt effektivt så skyddar det inte mot alla typer av buggar i operativsystemet.

4.3 Tredjeparts Antivirus

När operativsystemet inte längre får säkerhetsuppdateringar så är det lämpligt att se över sitt skydd mot skadlig kod för att säkerställa att det framöver tillhandhåller signaturfiler och detektering av skadlig kod. Om så inte är fallet bör man snarast möjligt införskaffa ett nytt skydd som även fortsättningsvis tillhandahåller regelbundna uppdateringar.

4.4 Använd en tredjeparts webbläsare

När operativsystemet inte längre får säkerhetsuppdateringar så innebär även det att integrerade webbläsare inte heller längre får säkerhetsuppdateringar. Det är därför rekommenderat att man inte längre tillåter att dessa används utan istället installerar en tredjeparts webbläsare, exempelvis Google Chrome. Google Chrome bör även den

konfigureras ganska strängt så att den inte exekverar Javascript/flash eftersom underliggande OS kan ha sårbarheter. Om den inbyggda webbläsaren krävs för tillgång tillverksamhetsystem är det rekommenderat att man bara tillåter den webbläsaren att nå utpekade webbsiter (vitlistning av webbsidor) för att på så sätt minska risken att skadlig kod exekveras via webbsidor på internet/intranät.

4.5 Avinstallera all mjukvara som inte behövs

En annan åtgärd man kan vidta för att minska antalet möjligheter för skadlig kod att exekveras är att avinstallera all mjukvara som inte är absolut nödvändig för att de system som kräver att man behåller det inte längre supporterade operativsystemet ska fungera. Exempel på sådan mjukvara kan vara Oracle Java, Adobe Flash, Adobe Reader och Microsoft Office.

Ofta är det en eller flera applikationer som är anledningen till att man inte kan uppgradera klienter. För att minska användandet av klienter med EoL operativsystem ska ENDAST problemapplikationerna finnas på dessa klienter, alla applikationer som fungerar på uppgraderad klient SKA användas på dessa och inte på en EOL klient. Motiveringen till detta är att klienten kan låsas ner ytterligare och på så sätt minska de risker som finns med att använda klienten. Detta innebär att användare behöver använda sig av två klienter under denna period.

4.6 Avlägsna enheten helt från nätverket / segmentering

Ett väldigt effektivt alternativ är naturligtvis att helt avlägsna IT-arbetsplatsen från nätverket och endast använda den som en självständig enhet. Tyvärr är detta sällan en möjlighet då de applikationer man använder ofta har behov av resurser på andra servrar eller enheter. Det är då även fördelaktigt att nätverksmässigt isolera enheten så mycket som möjligt, exempelvis kan man blockera all nätverksåtkomst förutom den som är nödvändig för att verksamhetsapplikationen ska fungera.

Så länge EoL klienter finns i miljön så ökar riskerna att dessa klienter kan skapa hot för andra klienter och system. Att nätverkssegmentera så att EoL klienter blir nätverksmässigt avskärmade från andra klienter och servrar och endast når de system och adresser som nödvändigt är ett effektivt sätt att minimera dessa risker.

4.7 Virtualisering

Desktop virtualisering och applikations virtualisering är exempel på virtualiserings lösningar. Att installera och tilldela en problemapplikation via någon virtualiseringslösning eller annan terminalserver lösning är ingen långsiktig plan, eftersom applikationen förmodligen då måste installeras på samma klient OS eller motsvarande server OS som också är att betraktas som EoL, det kan dock som en interimslösning vara skäligt. Viktigt att tänka på är att även vid

virtualisering så måste åtgärder vidtas för isolering och övrigt skydd då även en virtuell PC är en potentiell attackyta.

4.8 Behörigheter

4.8.1 Lokal administratör

De flesta hot mot klienter blir mycket allvarligare om användare har höga rättigheter, t.ex. att de är lokal administratör. Att bara tillåta standardanvändare och även strypa behörigheter lokalt men även mot andra system från dessa klienter är en nödvändig åtgärd, detta minskar riskerna och även administrationen av klienterna.

Det är även generellt en bra åtgärd att aldrig logga in med ett konto med administrativa behörigheter på en klient då dessa behörigheter även sparas där och gör det i så fall möjligt för en eventuell hackare att vid ett intrång stjäla uppgifterna till det administrativa kontot. I detta fall är det säkrare att använda det lokala administratörs lösenordet och sedan byta det med hjälp av LAPS regelbundet. Skulle man komma över lösenordet har man endast access till denna datorn.

4.8.2 Lokala inställningar och behörigheter

Endast de mest nödvändiga lokala behörigheter och inställningar för användaren samt klienten ska vara aktiva. Tjänster som är aktiva på klienten men inte används bör stängas av för att förhindra att de nyttjas negativt och blir ett säkerhetshot.
Tillgång till internet eller intranät ska strypas om möjligt och bara vara tillgängligt om det är absolut nödvändigt, t.ex. för datorer som är anslutna till eller kör administrativa verktyg för exempelvis medicinteknisk eller annan utrustning.

4.9 Säkerhetsuppdateringar och uppdateringar

Även om Säkerhetsuppdateringar slutar att levereras som standard när ett operativsystem går EoL så är det viktigt att säkerställa att dessa klienter har alla säkerhetsuppdateringar som finns att tillgå. Alla installerade applikationer ska ses över så de håller den senaste uppdateringsnivån. Att hålla en klient uppdaterad är alltid viktigt och ännu viktigare efter den går EoL (i den mån det finns uppdateringar att tillgå).

4.10 Lokal brandvägg

Lokal brandvägg ska vara aktiv på klienten för på inkommande och utgående trafik. På en EoL klient ska endast den trafik som är nödvändig tillåtas och det kan vara nödvändigt att nyttja olika regelverk för olika EoL klienter som endast öppnar för den nödvändiga trafiken. Regelverket ska styras central för att lättare kunna administreras.

4.11 Internet Access

Tillgång till internet från en enhet med OS eller mjukvara som är EoL är något som innebär stora potentiella risker. Det är dels ett sätt där användaren av enheten av misstag kan råka utsätta enheten för skadlig kod men det är också ett sätt för en obehöriga utomstående att via sårbarheter i mjukvaran som är EoL få tillgång till enheten. Därför bör det övervägas att blockera tillgången till Internet från enheter som har EoL mjukvara.

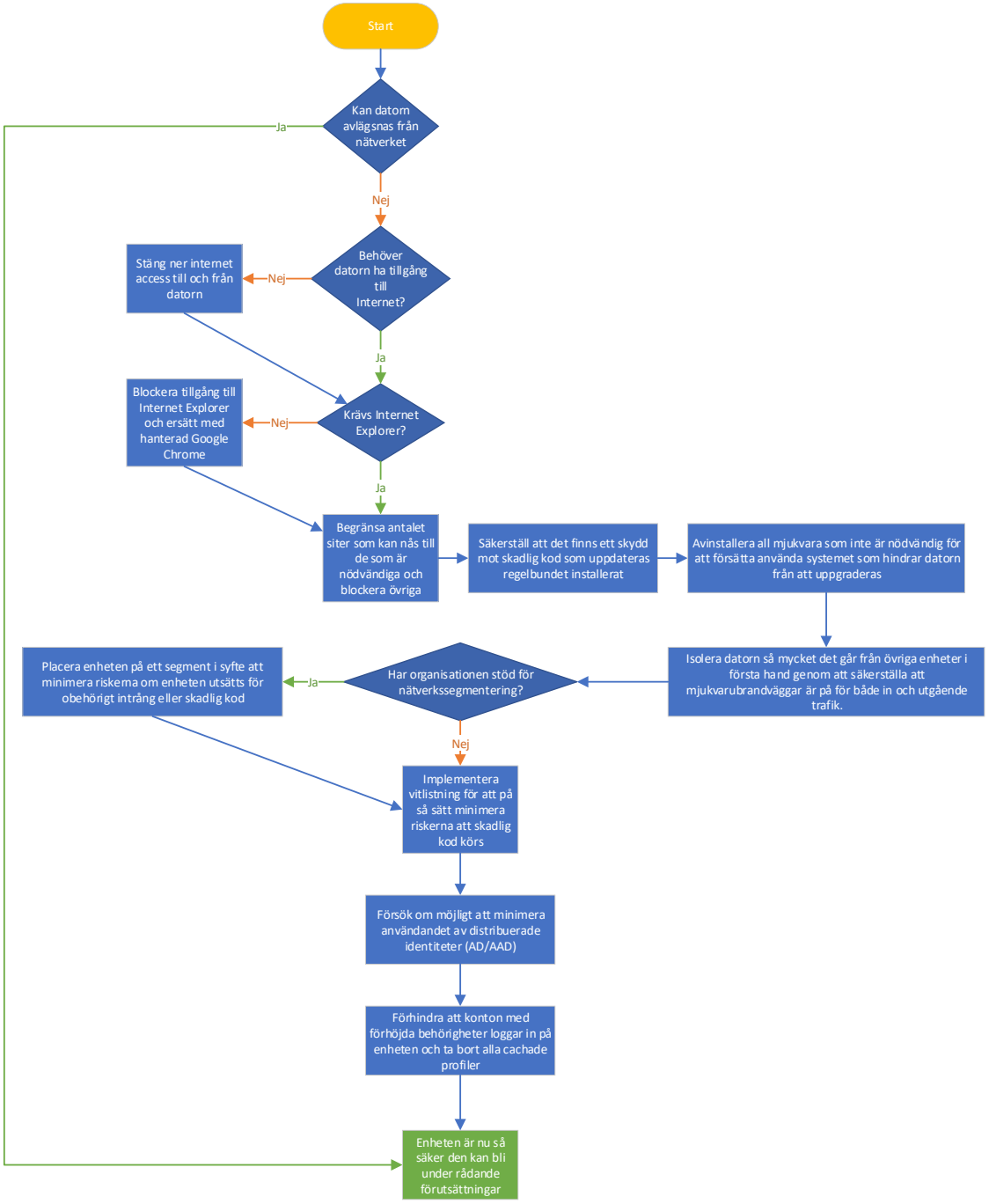
4.12 Hårdvara

Utöver alla övriga risker med att köra ett EoL operativsystem så innebär det oftast även att hårdvarutillverkarna slutar uppdatera drivrutiner och det i sin tur kan resultera i att obehöriga får tillgång till systemen. Det finns konkreta exempel där hårdvarutillverkare i exempelvis tangentbordsdrivrutiner har glömt kvar kod som i praktiken fungerar som en keylogger [3]och sparar allt som skrivs på tangentbordet. Ett sådant säkerhetshål skulle i så fall med största sannolikhet aldrig säkras upp av tillverkaren.

[3] Keylogger är en typ av datorprogramvara eller hårdvara med vars hjälp man kan registrera (logga) tangenttryckningar på en dator. Keyloggern registrerar alla knappar som trycks ned och kan på så vis till exempel få reda på lösenord.

5. Flödesschema

Nedan finns ett flödesschema som kan ses som ett exempel på hur en process för att säkra upp en enhet med EoL mjukvara kan se ut. Denna måste naturligtvis anpassas efter lokala behov.



6. Om det värsta inträffar

Om det trots att det vidtagits åtgärder för att säkra en enhet eller ett system slutar med att skadlig kod introduceras eller ett obehöriga individer får tillgång till systemen är det viktigt att han en handlingsplan även för detta.

Det är därför viktigt att ha dessa planer klara i god tid innan något händer och även om risken är mindre med supporterade system så är det trots det inte ovanligt att hackare eller andra illasinnade får tillgång till mjukvara trots att den är supporterad.

Dessa planer bör således finnas för alla verksamhetskritiska system och ännu hellre för alla system som finns i drift.

Det ska på förhand vara känt hur man ska agera för att avskilja de kompromitterade systemen från den övriga verksamheten och vilken påverkan det har så att nödvändiga reservrutiner kan tas fram.

7. Summering

Att använda mjukvara som är EoL är inte att föredra men tyvärr finns det ibland inga andra alternativ. Självklart är det effektivare att proaktivt jobba med att säkerställa att det aldrig uppstår en situation som gör att man tvingas ha kvar gammal mjukvara men i de fall det inte lyckas är det viktigt att göra en riskbedömning kring konsekvenserna av att ha kvar mjukvaran och att man agerar på de risker som identifieras.

I första hand bör man avskilja enheten helt från nätverket men om det inte går bör man bara tillåta det som är ett absolut måste för att enheten ska fungera.

Allt handlar om att minimera risker och på förhand planera för hur man ska agera om det värsta inträffar. Att ha en tydlig handlingsplan och vem som ansvarar för vad på förhand dokumenterat gör att det går att minimera konsekvenserna vid ett eventuellt säkerhetshot.

Det ska finnas planer för hur man säkerställer att det endast finns supporterad mjukvara i drift men det ska även finnas planer för hur man ska agera om man tvingas behålla mjukvaran efter att den inte längre supporteras.