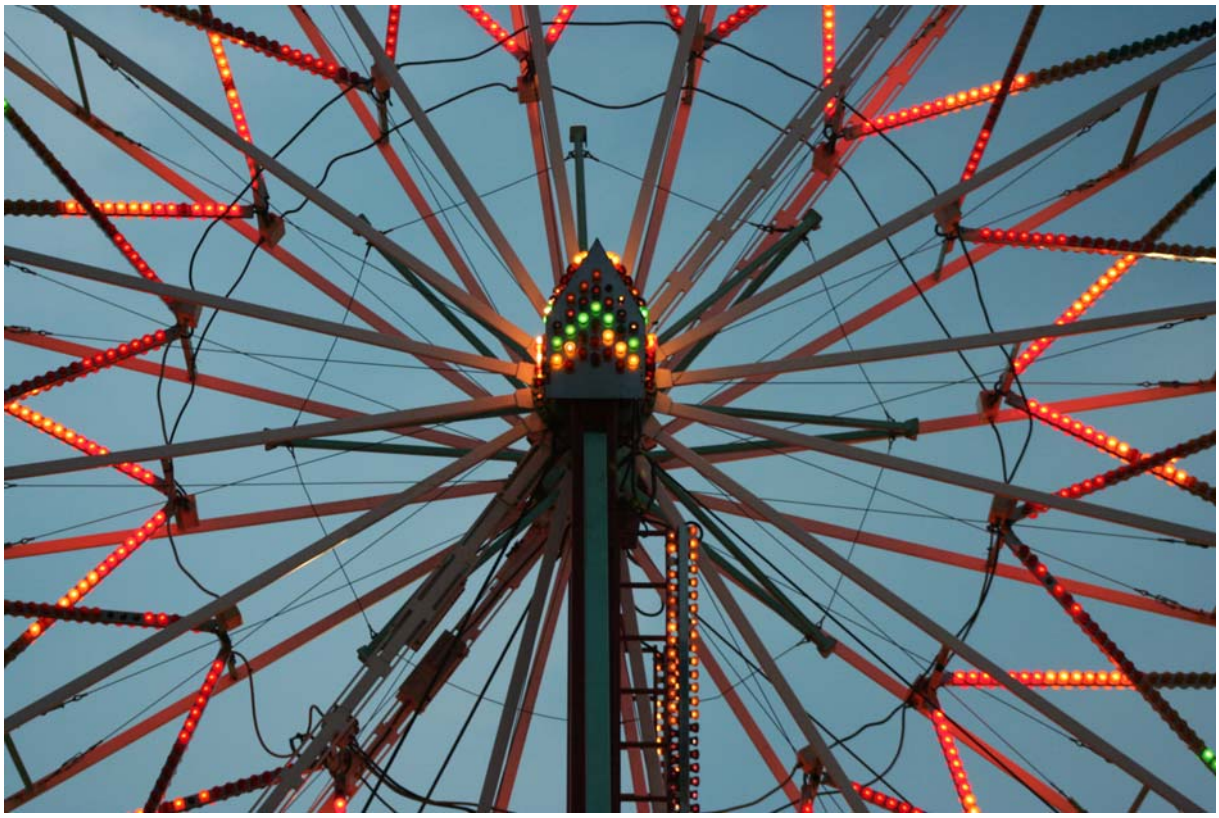
Tivoli

En datasäkerhetsföreläsning i sammandrag



IT-säkerhet, introduktion

IT – vad är det?

IT handlar åtminstone i den här diskussionen om datorer i olika former. Sådana datorer finns det lite överallt nuförtiden, utan att sakerna ser ut att vara datorer. Ta mobiltelefoner t.ex. Vad är det om inte datorer med dåliga tangentbord. Gemensamt för det vi pratar om i den här diskussionen är att det rör sig om maskiner som styrs av mjukvara.

Det var lättare att känna igen datorer förr. Då handhades de av män i vita rockar och den som ville köra program fick lämna in sin hålkortsbunt ena dagen, och fick resultatet nästa dag. Var det fel i programmet så var det förstås inte resultatet man fick, utan felutskriften.

Demo

> Ett telefonnummer visas upp på duken och någon får ringa upp det. Ingenting verkar hända, men det visar sig att uppringaren kan höra föreläsningen såväl i luren som i verkligheten.

Vad hände? Jo, en av telefonerna som fanns i lokalen var specialpreparerad. Det var visserligen en helt vanlig Nokiatelefon, men med utbytt mjukvara. Telefonen uppförde sig precis som vanligt, men som bonus fanns det ett ”spionläge” som aktiverades av en särskild tangentkombination. I spionläget såg telefonen avstängd ut, men om någon ringde till den så kopplades samtalet upp utan någon som helst signal på telefonen, varken synlig eller hörbar. Med den nya mjukvaran fungerade alltså telefonen som en sorts buggningsutrustning, med global räckvidd och gott om batterier.

Just den här telefonen var köpt från en svensk leverantör på nätet. Vid det här tillfället gick det att köpa antingen bara mjukvaran för ca 500 kronor, eller en hel telefon färdigpreparerad med mjukvaran för ca 2000 kronor. Det var alltså inte särskilt dyrt, eftersom det bara handlade om mjukvara.

Det fanns även andra varianter och andra leverantörer. En intressant sort var sådan att den uppförde sig precis som en vanlig telefon förutom när den blev uppringd från ett speciellt telefonnummer. När detta hände svarade telefonen utan att ge någon signal från sig, den fortsatte att se påslagen men inaktiv ut. Ville användaren ringa så kopplades det pågående, avlyssnande, samtalet ner omärkligt för att användaren inte skulle misstänka något. Det var alltså en telefon som det var lämpligt att lura på någon annan, eftersom den gjorde det möjligt att avlyssna omgivningen närhelst det var önskvärt.

Demo

> En installationsfil för ett mycket bra program körs, men tyvärr var något fel så det resultar bara i ett felmeddelande. Därefter kan datorn fjärrstyras av angriparen.

Nu har vi diskuterat vad IT är. För att reda ut vad IT-säkerhet är så är nästa steg att prata om vad säkerhet är.

Säkerhet – vad är det?

Säkerhet handlar, per definition, om skydd från något. Varför sitter vi i allmänhet inomhus när vi går på kurs, och inte utanför? Jo, inomhus är vi, och vår utrustning, skyddade från väder och vind. Men, det finns inte någon absolut säkerhet, utan vi strävar efter att vara säkra nog i förhållande till hotens art och sannolikhet. Vi måste alltså göra en bedömning av vilken sorts skydd vi behöver. T.ex. så är vi inte skyddade från artelleriattacker i en normal inomhusmiljö

eftersom huset är för klent. Trots detta känner vi oss rätt väl till mods, eftersom vår bedömning är att sannolikheten för artelleriattacker är så liten att vi inte behöver skydda oss mot sådana.

Några saker som är speciellt för just IT-säkerhet:

1) Det är svårt och dyrt att säkra system i efterhand. Säkerhet behöver byggas in under hela utvecklingsfasen.

2) Användare förstår inte vad som händer i systemen eftersom systemen är så komplexa.

3) De som bygger systemen vet inte hur de kommer att användas.

Exempel:

> De flesta som läser den här texten har antagligen använt Microsoft Word. Flera har säkert också installerat Word någon gång. Men, de flesta har nog *inte* läst längre än till ”I agree”! Den som gjort det noterade kanske att det i licensavtalet står att Word inte får användas till att styra kärnkraftverk!

4) Ett problem är att när någon beställer ett IT-system så är prioriteringen i första hand kostnad, i andra hand nytta och först därefter säkerhet.

Ett exempel på flera av dessa problem är många av de IP-telefonisystem som säljs just nu.

Demo

> En trådlös IP-telefon används för att ringa upp en fast IP-telefon i rummet. Det visar sig att efter det att samtalet är över så kan det spelas upp av en lyssnande dator.

Demo

> Konfigurationen av den trådlösa IP-telefonen görs via ett webbgränssnitt. Det visar sig att ingen autenticering behövs för att göra detta.

Det som visades var att med hjälp av gratisprogrammet Wireshark, som är ett program för att följa nätkommunikation, så kunde inte bara signaleringen uppfångas, utan även ljudströmmarna kunde avkodas och återuppspelas. Samtalet gick visserligen över en trådlös förbindelse, men demonstrationen hade fungerat lika bra över en trådbunden länk. Det som skulle visas var att IP-telefoni i allmänhet är oskyddad, och därmed lätt att avlyssna. Både uppkopplingen (SIP) och ljudkommunikationen (RTP) går i klartext.

Vad är det då som driver införandet av IP-telefoni, både privat och i företag. I allmänhet är det jakten på minskade kostnader. Tyvärr läggs ofta allt ansvar för säkerheten på användaren, utan hjälp eller stöd från tillverkaren av produkten. För företag som vill införa IP-telefoner på skrivborden hos de anställda går det att göra på ett rimligt säkert sätt, men det kräver att de som designar näten som ska användas gör det på ett bra sätt, så att avlyssning inte blir allt för enkel att genomföra. T.ex. bör telefonerna inte sitta i samma lokala nät som datorerna på företaget.

Inom IT-området används en stor mängd metaforer. Metaforer är praktiska för att förmedla en bild av något, men ska man göra säkerhetsbedömningar så behövs större förståelse än vad metaforerna ger. Det som är lurigt är att metaforer kan få en att *tro* att man förstår något som man egentligen inte förstår. I vissa andra branscher har man istället en helt egen terminologi, t.ex. inom medicin eller sjöfart. När någon seglare ber en att skränsa tampen som ligger om moringen, så förstår man ingenting om man inte själv är seglare. Det är dock också en sak att förstå, nämligen att man inte förstår.

Användare har ofta en viss känsla för vad som är säkert och osäkert att göra. Dock är denna känsla normalt helt skild från den bild som IT-säkerhetsfolk har.

I bästa fall har vi lyckats lära oss att vi inte ska klicka på bilagor som vi får i vår inbox om vi inte vet varifrån de kommer och det verkar rimligt att vi fått den. En relevant fråga är dock hur många som egentligen lärt sig detta, på riktigt. 20–30% ?

USB-pinnar

USB Dumper

Demo

> En USB-pinne stoppas in i datorn. Efter att ha tittat filerna på pinnen tas den ur maskinen. Det visar sig att alla filer på pinnen kopierats över till datorn. Detta inkluderar även en dold katalog, som inte syntes när innehållet visades.

Det som hände var att ett program som heter USB Dumper var aktivt i datorn. Detta program detekterar när USB-minnen ansluts och kopierar minnesinnehållet till en mapp i den katalog programmet startades från. Tekniskt sett är ett program som USB Dumper en enkel sak, men med tanke på hur många USB-minnen som ansluts i olika datorer, även i fall där användarna inte egentligen känner varandra, så kan ett sådant här program antagligen hitta en hel del känslig information, till exempel på en mässa eller konferens där många byter dokument med varandra genom att använda USB-minnen.

JPEG-bild på USB-pinne

Demo

> En USB-pinne stoppas in i datorn. Efter att ha tittat vad filerna på pinnen heter och är för typ (någon pdf, någon bild, några texter mm.) tas pinnen bort, utan att någon fil öppnats. Det visar sig dock att trots detta rätt försiktiga beteende har en bakdörr installerats på datorn.

Vad som hände var att en bug i jpeg-hanteringen i Windows utnyttjades för att installera bakdörren. För, bara för att ingen fil öppnades av användaren så innebär inte det att Windows inte öppnade någonting. När Windows får syn på en bild så öppnas den och Windows ”kollar läget” med bilden.

Felet i jpeg-hanteringen var följande: En jpeg-fil innehåller en del extrainformation i början av filen, som kan innehålla data om kameran och dess inställningar till exempel. Det finns dessutom ett kommentarsfält, som kan användas för fria kommentarer i text. När bilden öppnas kopieras innehållet i kommentarsfältet till en för ändamålet avsedd buffert i bildvisningsprogrammet. I det här fallet var bilden specialpreparerad, och kommentaren var jättelång, längre än vad som får plats i kommentarsbufferten i bildvisningsprogrammet. Själva buggen var att programmet inte korrekt kollade hur lång kommentaren var, utan istället kopierade hela kommentaren trots att den därmed skrev över annan information i datorns minne. Kommentaren var designad så att den innehöll körbar kod som kunde fås att exekveras och därmed öppna bakdörren in i operativsystemet.

USB-pinne med u3-funktionalitet

Demo

> En annan pinne, synbarligen mycket lik den förra pinnen, stoppas in i datorn. Denna gång tittar vi inte ens efter vad som fanns på pinnen. Trots detta blir maskinen mycket uppenbart ”hackad”.

I det här fallet var det en USB-pinne med en nyare teknik som heter u3. En u3-pinne utger sig för att vara både ett minne och en CD-spelare. CD-spelare är den enda sorts enhet som får

automatstarta program i Windows. Problemet är att det är enkelt att byta ut det ordinarie automatstartande programmet på u3-pinnen (som är välartat och fyller en viss funktion) mot vilket program som helst, t.ex. ett som gör något elakt. Lämpligen görs det elaka utan att det syns på utsidan (till skillnad från demot). Denna typ av USB-minnen är obehagliga eftersom det är mycket accepterat att låna och låna ut USB-minnen till varandra.

Demo

> En normal u3-pinne demonstreras. Därefter programmeras den om så att den får funktionaliteten hos den ”hackande” u3-pinnen. Detta visar sig vara mycket enkelt.

Metoden för att programmera om en u3-pinne är att använda sig av det uppdateringsprogram som finns för att uppdatera pinnens systemprogramvaran. Normalt ansluter uppdateringsprogrammet över nätet till en webbadress hos tillverkaren och hämtar den senaste versionen av systemprogramvaran och skriver till u3-pinnen. Genom att lokalt starta en webbserver som utger sig för att vara tillverkarens webbserver så kan uppdateringsprogrammet luras att installera vilken mjukvara som helst, till exempel något elakt (eller något praktiskt).

USB-pinne med biometriskt skydd

Det finns ett stort antal produkter av typen USB-minne som skyddas med biometri. I alla fall vi sett så ska biometri tolkas som fingeravtryck. Produkterna marknadsförs som ”säkra”, i betydelsen att bara behöriga personer kan komma åt informationen som finns på dem.

Argumentationen kring säkerheten hos fingeravtryck gör ofta en logisk kullerbytta. Det brukar se ut ungefär såhär:

> Alla människor har olika fingeravtryck. Ett system som använder fingeravtryck för att verifiera identiteten hos personerna är därför mycket säkert.

Men säkerheten i en tillträdeslösning beror inte direkt på hur många användare som kan råka ha samma ”nyckel” till systemet, utan hur lätt det är för en angripare att presentera en giltig kombination av påstådd identitet och tillhörande ”nyckel”.

Säkerheten begränsas i tillträdeslösningar som bygger på fingeravtryck av att fingeravtrycken från en viss given människa är lätta att få tag på och att det är lätt att tillverka falska fingeravtryck som de flesta fingeravtrycksläsare accepterar som giltiga. Hur lätt det är beskrivs bland annat i ett examensarbete från Linköpings universitet av Marie Sandström (Detektering av Artificiella Fingeravtryck vid Användarautentisering**,** LITH-ISY-EX-3557-2004).

Penetrationstest med USB-pinnar

Ett penetrationstest är när man hyr in någon för att försöka bryta sig in i ens eget IT-system. Antingen gör man det för att man vill testa och se hur svårt det är, och kanske lära sig om något som bör förbättras, eller så kan det vara som i det här fallet, att syftet var att skaka om användarna för att få dem att bli mer säkerhetsmedvetna.

Ett vanligt tillvägagångssätt vid penetrationstest är att försöka prata sig förbi receptionen och ta sig till ett konferensrum eller liknande och där koppla in en bärbar dator i det interna nätet. I det här fallet provade företaget som fått uppdraget, Secure Network Technologies, en annan metod. De tog ett antal gamla USB-minnen, av den klassiska typen, och preparerade med lite lagom intressant information. Dessutom lade de till en körbar fil, kamouflerad som en bild. När programmet kördes tankades datorn av på intressant information som sedan skickades över Internet till en mottagare hos Secure Network Technologies. Tidigt på morgonen, innan någon kommit till jobbet, gick de med USB-minnena till huset där företaget bodde, och

placerade ut minnena lite här och där. Därefter gick de till ett kafé och väntade. Så fort personalen började komma till jobbet började också informationen från deras datorer att trilla in. Av 20 utplacerade pinnar hittades 15. Alla dessa stoppades in i någon dator i det attackerade företaget. Informationen som kom ut kunde sedan användas för att göra en manuell attack.

DCOM-attack

Om vi aldrig stoppat in något alls i maskinen, utan låter den vara precis som den var när den var nyinstallerad. Är den säker då? Nej, naturligtvis är den inte det. Det som ansågs säkert för ett år sedan behöver inte vara det längre. Nya säkerhetsluckor hittas ständigt, varför operativsystem och program behöver uppdateras frekvent.

Demo

> En maskin med Windows XP uppdaterad till och med service pack 1 användes. Maskinen hade inte preparerats och inte misskötts på något sätt. Det visade sig trots detta vara möjligt att via ett nätanrop installera en bakdörr.

Denna gång var det ytterligare en buffer overrun, närmare bestämt just den sort som masken Blaster använde sig av. Tekniken är lik den som användes vid jpeg-demot tidigare. Det som hände var att ett lämpligt utformat IP-paket skickades till en tjänst i Windows som heter DCOM. DCOM är en ganska gammal och inte så flitigt använd, teknik i Windows, men den finns kvar av kompatibilitetsskäl. Ungefär på samma sätt som i jpeg-fallet så är det en reserverad buffert i datorn som inte är stor nog för den data som skrivs dit, och på samma sätt skrivs programmet över med ett nytt program, som står för elakheterna.

Buffer overrun igen! Detta är det vanligaste säkerhetsproblemet idag, och har varit så i över 30 år. Varför är det så? Det borde inte vara svårt att undvika, för det är inte svårt att göra något åt det här. Det handlar bara om att kontrollera att det som kopieras till en buffert inte är större än bufferten.

Är det Windows som är problemet?

Är det inte så enkelt att det är Windows som är problemet? Nej, riktigt så enkelt är det nog inte som att Windows alltid är sämst. Historiskt har det varierat vilket OS som varit mest drabbat av säkerhetsproblem. Åtminstone en förklaring är att ”man angriper det man kan”, i två betydelser. 1) Naturligtvis är det roligare att ge sig på system som det finns många av, och dessutom finns det mer att göra då också. 2) Det finns förstås många som kan mycket om de system det finns många av. Alltså finns det många som har kompetens att angripa systemen.

Vi har hittills bara pratat om attacker mot Windowssystem. Nu är det dags för ett Linuxexempel. Ett problem med Linux är att vid installationen frågas om vilka komponenter av en väldig massa som ska installeras. Många tycker att eftersom de just köpt en jättehårddisk, så ska de naturligtvis installera allt. Sedan använder de inte alla dessa komponenter och följer därför inte med och uppdaterar dem efterhand som det behövs. Ofta kör alltså Linuxsystem en väldig massa program och tjänster som användaren inte är intresserad, eller ens medveten, om.

Attack mot en web-server

Men om vi nu gjort allt rätt, kört alla uppdateringar, har en brandvägg osv. Kan då något gå fel ändå?

Demo

> På datorskärmen syns FOI:s hemsida, i demot hämtad från en lokal maskin som fungerar som webserver. Ett program körs på en attackmaskin, men ingen kommunikation sker med ”användarens” dator. Nästa gång vi laddar FOI:s hemsida ser den annorlunda ut.

Hur kunde det här hända? ”Användarens” maskin var ju inte ens attackerad. Eller?

För att reda ut det här måste vi beskriva hur websurfning går till. När användaren vill titta på en websida kommer maskinen som användaren använder för att surfa (klienten) att be webservern om denna sida. När webservern får förfrågan kommer denna att svara och leverera den efterfrågade sidan och klienten kan visa upp sidan för användaren.

I demot attackerades webservern av en illasinnad maskin som lyckades ersätta websidan hos webservern med en sida med annat utseende. Detta får till följd att alla som efterfrågar denna sida kommer att få den felaktiga sidan istället. Attackeraren lyckas alltså framtvinga att en massa maskiner förses med felaktig information, trots att det bara är webservern som attackerats. Helt säkra är vi alltså inte ens om vår egen maskin är helt säker.

Om vi antar att även webservern är säker för intrång, klarar vi oss då?

Demo

> Ett program startas på den angripande datorn, men ingen attack görs mot någon annan dator. När klienten laddar om websidan har den förändrats igen.

I det här fallet utnyttjade vi säkerhetsbrister hos de underliggande nätverksprotokollen. Vi attackerade alltså inte någon maskin i systemet, utan lurade dem genom att utnyttja egenskaper hos protokollet ARP.

ARP är det protokoll som används för att i lokala nätverk slå upp nätverksinterfacens hårdvaruadresser utifrån de globala IP-adresserna. När klienten vill hämta information från webservern känner den till webserverns globala adress, men för att kunna genomföra kommunikationen behövs (den lokala) hårdvaruadressen. För att hitta den skickas en fråga ut (ARP request) där det efterfrågas vem som innehar den globala adressen xxx.xxx.xxx.xxx. Den maskin som har den angivna globala adressen svarar och meddelar sin hårdvaruadress, varefter den önskade kommunikationen kan genomföras.

För att inte behöva göra sådana ARP-frågor efter hårdvaruadresser allt för ofta (effektivitet) sparar varje dator kopplingen mellan global adress och hårdvaruadress, så att om den används snart igen kan den göra uppslagningen internt. För att ytterligare öka effektiviteten sparas sådana kopplingar även om datorn inte gjort ARP-frågan själv, utan hört en annan ARP-fråga eller ARP-svar.

Eftersom alla enheter lyssnar även efter andras ARP-anrop och ARP-svar så är det möjligt informera omgivningen genom att skicka ARP-svar även om det inte kommit någon fråga. På detta sätt kan en enhet som t.ex. byter hårdvaruadress informera kringliggande enheter om detta. I demot användes tekniken för att (felaktigt) meddela enheterna på nätet att webserverns globala adress numer hörde ihop med den attackerande datorns hårdvaruadress. Genom denna omdirigering gick alla förfrågningar efter websidor hos webservern istället till den attackerande datorn. En möjlighet som inte utnyttjades var för den attackerande datorn att välja vad den ville skicka för websidor, t.ex. beroende på vem som frågade. T.ex. kunde den valt att visa den korrekta websidan för datorer som skulle kunnat avslöja attacken, för att på så sätt undgå upptäckt så länge som möjligt.

Masken Slammer och kärnkraftverket

Det följande är en berättelse från verkligheten. Beskrivningen skiljer en del mellan olika versioner av berättelsen, men huvuddragen är gemensamma.

Slammer var en mask som spreds över Internet mellan Windowsmaskiner under 2002. Från sommaren 2002 fanns det uppdateringar att hämta som stängde den säkerhetslucka som Slammer använde för att sprida sig över Internet. Slammer var en relativt godartad mask. När den infekterat en dator gjorde masken inget annat än att slumpa fram nätadresser som den försökte attackera. Alltså var det värsta som kunde hända att nätet blev överbelastat.

I berättelsen fanns följande aktörer: First Energy – ett elbolag som producerar och distribuerar el i USA, en icke namngiven underleverantör, Davis-Besse – ett First Energy-ägt kärnkraftverk med många dokumenterade incidenter.

Den 25 januari 2003 hände följande: Slammer kom in i underleverantörens nät via en laptop som en anställd haft uppkopplad mot Internet utanför företagets brandvägg. Slammer spred sig snabbt, och via en T1-linje som gick direkt mellan First Energy:s nät och underleverantörens nät hittade masken in i First Energys datorer. Direktkopplingen fanns eftersom bolagen samarbetade och litade på varandra, och naturligtvis var näten sammankopplade innanför brandväggarna.

Kärnkraftverket hade två nät, ett administrativt och ett för kontrollen av själva kraftverket. Av säkerhetsskäl var inget av dem kopplat till Internet, men det administrativa nätet hade en koppling till First Energys nät, för att kunna skicka administrativ information till moderbolaget. Tyvärr är IP-förbindelser dubbelriktade så Slammer letade sig in i kraftverkets administrativa nät. Detta borde inte ha varit någon fara, men det visade sig att kontrollnätet och det administrativa nätet delade kablar, trots att de inte kunde, och inte heller skulle kunna prata med varandra. De var logiskt åtskilda, men fysiskt sammankopplade. Detta fick till följd att när Slammer spridit sig till många datorer i det administrativa nätet (men inte till någon i kontrollnätet) och det blev överbelastat, så blev även kontrollnätet överbelastat. Det blev omöjligt att kommunicera med kraftverket via kontrollnätet, trots att inga av kontrolldatorerna var, eller kunde bli, smittade av masken.

Eftersom kärnkraftverk är en verksamhet med mycket höga säkerhetskrav så fanns det naturligtvis analoga nödsystem som fortfarande fungerade. Emellertid behövdes inte dessa eftersom kraftverket var avstängt för service sedan ett antal veckor, och därmed inte utgjorde någon säkerhetsrisk ens när det inte kunde kontrolleras.

Vems felet var är svårt att svara på. Det var många fel som tillsammans gjorde det möjligt för Slammer att slå ut kontrollen av kraftverket. Framför allt är händelsen en bra illustration av det faktum att det är svårt att förutsäga hur säkerhetsproblem kan uppstå som en följd av kombinationer av svagheter i systemet. Detta motiverar varför även principiella svagheter bör åtgärdas, även om de till synes inte kan utgöra reella hot. Händelser och egenskaper kan samverka på sätt som är oförutsägbara i förväg. Enda sättet att hantera detta är att bygga säkerhet i flera, oberoende lager, som var för sig är tillräckliga.

Sammanfattning så här långt

Det som beskrivits hittills är situationen sådan den är idag. Trenden är att det kommer att gå mot ännu större extremer i framtiden. Detta då datorerna blir allt mer specialiserade. I en vanlig bil finns idag femton till tjugo datorer. I ett normalt hushåll finns närmare hundra om man räknar mikrodatorer som finns i olika sorters elektronikprodukter. Det är lika bra att

vänja sig vid tanken på att datorer kommer att vara en del av säkerhetsproblem i den överskådliga framtiden.

Användarnas bild och verkligheten

En enskild dator består av ett stort antal delar, vilka kan fördelas på ett antal olika nivåer:

- Längst ner finns hårdvaran. Allt annat i en dator är mjukvara.
- Lägsta nivån av mjukvara är I/O-rutinerna, som är specialskrivna för just den hårdvara som finns i den aktuella datorn. Dessa ”gömmer” hårdvaran för resten av mjukvaran, så att ingen annan behöver bekymra sig för exakt vilken hårdvara som är installerad. Istället kommunicerar resten av datorn med I/O-rutinerna, som visar upp ett standardiserat gränssnitt som inte beror på hårdvaran.
- Ovanpå I/O-rutinerna finns operativsystemet (OS:et) som är det/de program som styr datorns verksamhet. OS:et startar andra program, erbjuder tillgängliga tjänster till användaren osv. Exempel på OS är Windows, MacOS, Unix och Linux.
- Ovanpå OS:et finns tillämpningsprogrammen som är de program som gör något som användaren är intresserad av. Exempel är Word, Excel och Firefox.

Bilden som de flesta användare har är att de kommunicerar med tillämpningsprogrammen, t.ex. att de skriver saker i Word. I verkligheten går kommunikationen mellan användaren och Word inte alls direkt. Användaren skriver på ett tangentbord, som kontrolleras av I/O-rutiner, som kommunicerar med Windows (operativsystemet), som i sin tur skickar information om vad användaren gjort till Word. Kommunikation i andra riktningen, t.ex. en förändring på skärmen som användaren inducerat, går från Word till Windows, från Windows till skärmens I/O-rutin, och från I/O-rutinen till hårdvaran. Det är alltså ett stort antal steg inblandade som användaren inte är medveten om, och vart och ett av dessa steg är möjliga att påverka.

På samma sätt är det med epost. Användarens syn är att eposten går från dennes epost-program till mottagarens epost-program. I själva verket pratar användarens epost-program med OS:et, som ser till att mottagarens OS tar emot meddelandet och skickar det vidare till epost-programmet. Naturligtvis skickar OS:et i sin tur först meddelandet ner till I/O-rutinerna, som skickar vidare till hårdvaran, som sköter den verkliga transmissionen av informationen. Fast när det gäller epost så är sällan avsändande dator kopplad direkt till den mottagande datorn. Detta innebär att kedjan med kommunikation mellan olika nivåer i datorn genomförs i ett antal maskiner längs vägen från sändaren till mottagaren.

Funktionsfaran

Ett tilltagande problem är det ständigt växande antalet funktioner som programmen innehåller. Mer specifikt beror problemet på att inte bara varje funktion kan innehålla säkerhetssvagheter, utan att varje *kombination* av funktioner kan ge upphov till oavsiktliga, farliga svagheter. Med dagens program är antalet kombinationer av funktioner gigantiskt, och det är omöjligt att verifiera att inga oönskade effekter uppstår vid någon av dessa kombinationer.

Någon har påstått att en duktig användare använder 20–30 olika funktioner i Word. Någon annan har räknat och kommit fram till att i Office-paketet som helhet finns nära 1500 olika funktioner. Frågan är om alla funktioner verkligen behövs, eller ens används av någon alls.

Sammanfattningsfunktionen

Vi på FOI har en favoritfunktion i Word – ”sammanfattning”! Funktionen ”sammanfattning” är en funktion som automatiskt skapar en sammanfattning av det aktuella Worddokumentet. Tanken är att om det behövs en sammanfattning så kan Word skapa den automatiskt, eller om det är ett långt dokument som användaren behöver få information om så kan Word skapa en kortfattad sammanfattning som användaren kan läsa istället. Det vore en fantastisk funktion om den fungerade, vilket vi anser att den inte gör. För att testa lät vi Word göra en sammanfattning på 20 meningar av 1917 års svenska bibelöversättning.

> 13,1. Upp. 2,9. Upp. 6. Och HERREN, din Gud. 20,12. Upp. 4,24. 14,15. Upp. 6. Du allena är HERREN. >Upp. 2,6. Upp. >Upp. 2,6. Upp. 7,34. 16,9. 33,11. Upp. 51,7. Upp. 51,6, 45. Upp. 50,8. Upp. 46,11. Upp. >Upp. >Upp. 3,24. Upp. 10,9. Upp. 15. Och du.

Vi brukar säga att vissa av kristenhetens finare nyanser har fallit bort.

Funktionen ”snabbspara”

Ett annat exempel på en funktion som har säkerhetsimplikationer är ”snabbspara” i Word 97. Med ”snabbspara” aktiverat så går det något snabbare att spara dokument, vilket gör det lockande att använda funktionen. Den är också aktiverad som standard i Word 97.

Demo

> En offert med harmlöst utseende visade sig innehålla mindre smickrande kommentarer i tidigare versioner. Dessa tidigare versioner gick att hitta genom att öppna dokumentet i t.ex. Notepad.

Normalt i Word är det möjligt att ångra ändringar som görs, vilket tyder på att originalinformationen finns kvar någonstans, det är bara det att den inte syns. I normalfallet när ett dokument sparas går Word igenom dokumentet och ”städar upp” så att bara den information som hör till dokumentets aktuella status finns kvar, varefter denna städade version sparas till disk. Denna uppstädning tar lite tid att genomföra, vilket har motiverat införandet av funktionen ”snabbspara” som är snabbare eftersom den *inte* städar upp i informationen innan den sparas. Detta får dock till följd att även information som är raderad ur dokumentet följer med när det sparas till disk. Denna kan förstås plockas fram igen av någon som har tillgång till lämpliga verktyg, t.ex. Notepad i Windows.

Ett exempel är en rapport som den brittiska regeringen släppte, en rapport som låg till grund för invasionen av Irak. Rapporten släpptes till media i Wordformat och när media granskade Wordfilen fanns där information som regeringen inte avsett sprida. Bland annat framgick det att den delvis var plagierad från offentliga källor, fast med vissa formuleringar tillspetsade för att låta mer spännande. Det syntes också att dessa ändringar inte gjorts av personer med kunskap inom området. Dokumentet kom att bli känt under namnet ”Dodgy Dossier”.

För att undvika att information läcker på detta sätt bör inte ”snabbspara” användas. Det är dessutom så att om den används i dokument som ändras mycket kan filstorleken växa så till den grad att det går långsammare att snabbspara än att spara på vanligt sätt. Det är också värt att notera att Worddokument inte är avsedda att använda för informationsdistribution. Worddokument skickar man lämpligen mellan sig när man skriver på ett gemensamt dokument. När det väl ska distribueras till den eller de som ska läsa dokumentet så bör något annat, lämpligare format användas. Exempel på sådana format är Adobe Portable Document Format (pdf), Hypertext Markup Language (html) och utskrift på papper.

Risken med överstrykningar

Demo

> Ett worddokument visas med delar av texten överstruken. Tanken är att demonstrera hur känsliga delar i ett dokument kan döljas innan dokumentet distribueras. Några enkla handgrepp i Word får dock den överstrukna texten att framträda tydligt.

Det här demot visade ett liknande problem, nämligen faran med att avhemliga ett elektroniskt dokument genom att göra överstrykningar eller andra liknande modifikationer. Det finns alltid en risk att informationen finns kvar, mer eller mindre uppenbart.

Ett exempel är ett dokument angående en Irakincident som Pentagon avhemligade genom att stryka över vissa delar. Dokumentet exporterades sedan till pdf-format och distribuerades till media. Misstaget bestod i att dokumentet, trots att det gjorts om till ett nytt format, innehöll hela texten men med svarta streck över de känsliga delarna. Det var naturligtvis enkelt att leta fram informationen som de svarta strecken dolde. Efter detta har Pentagon meddelat att de ändrat sina rutiner för avhemligande.

Påskägg

Påskägg är dolda, odokumenterade tillägg, ofta av humoristisk karaktär, som finns i t.ex. datorprogram. De lockas fram genom osannolika kombinationer av tangenttryckningar och musklickningar. Påskägg är vanliga trots att de ses som mycket oönskade av företagen som står bakom produkterna. På Internet finns listor med påskägg, t.ex. www.eeggs.com

Demo

> Det visar sig att Excel och Word (i rätt version) innehåller en flygsimulator respektive ett flipperspel.

Påskägg kan vara underhållande och är i allmänhet ofarliga, men, de är också ett tecken på någonting. Om det går att gömma en flygsimulator eller ett flipperspel i en produkt av den dominerande mjukvarutillverkaren i världen, hur kan vi då lita på att inte något annat, mer illasinnat, är gömt i de program vi använder? Vilket program som helst kan innehålla nästan vad som helst!

En orsak är att testning av program huvudsakligen ägnar sig åt att verifiera att allt som ska finnas med verkligen finns med och fungerar. Vad som finns förutom detta studeras inte närmare. Det är också så att det är mycket svårt att verifiera att inget extra finns med i ett program, även för den som tillverkar programmet. För den som är användare är det omöjligt.

Någon lösning på detta problem finns egentligen inte. Som användare av andras system måste vi räkna med möjligheten att de kan innehålla illasinnade funktioner, även i fallet med säkerhetsprogramvara. Vi måste bygga våra system utifrån denna förutsättning, och konstruera säkerhet i flera nivåer och hoppas på att inte alla säkerhetsfunktioner fallerar samtidigt. Om möjligt bör våra IT-system designas så att vi inte är i en sämre situation om systemet går sönder, än om vi aldrig installerat det från början.

Digitala signaturer

Digitala signaturer är namnet på en sorts kryptografiska metoder vars användande påminner om hur vi använder vanliga namnunderskrifter.

Underskrifter på papper

Ett vanligt dokument, tryckt på papper och underskrivet, har en särskild innebörd. Genom att skriva under dokumentet intygar personen att det som står i dokumentet är korrekt. Underskriften fungerar för att visa acceptans eller viljeyttring gentemot det som är skrivet. Tekniskt möjliggörs detta av att det är relativt svårt att skapa en underskrift på ett papper om inte underskriftens ”ägare” hjälper till. Att flytta en underskrift från ett papper till ett annat kräver antingen kopiering eller klipp-och-klistra, och båda metoderna är oerhört svåra att lyckas med utan att en enkel granskning avslöjar förfalskningen. Likaså är det svårt att göra falska underskrifter, dvs. skriva en namnteckning som är tillräckligt lik någon annans. De tekniska svårigheterna stöds ytterligare av att det i lag anses vara ett relativt grovt brott att skapa falska underskrivna dokument.

Digital signering och verifiering

Digitala signaturer påminner om vanliga underskrifter, men eftersom de bygger på kryptografiska metoder så är allt byggt kring kryptonycklar istället för personer och deras namnteckningar. Nycklarna förekommer i par, en hemlig signeringsnyckel och en ohemlig verifieringsnyckel.

Signeringsnyckeln används tillsammans med signeringsalgoritmen och dokumentet för att skapa själva signaturen, som är en till synes slumpmässig bitsträng som bifogas dokumentet som signerats. Eftersom signeringsnyckeln är hemlig så är det bara innehavaren av denna som kan skapa just denna signatur. Det innebär att varje person som vill kunna signera dokument måste ha en egen, hemlig nyckel, som bara just den personen använder vid sina signaturer.

Varje signeringsnyckel har en tillhörande verifieringsnyckel, som passar bara till just den signeringsnyckeln. Tillsammans med verifieringsalgoritmen, dokumentet och den digitala signaturen används verifieringsnyckeln för att kontrollera att alla ingående delar passar ihop och fungerar korrekt tillsammans. Om de *inte* gör det så är det någon av de ingående delarna som är fel. Det kan vara dokumentet som inte är det som ursprungligen signerats, eller verifieringsnyckeln som inte hör till signaturnyckeln. Dessa två händelser går i allmänhet inte att skilja på, utan den verifierande får nöja sig med att dra slutsatsen att detta dokument inte signerats av den person vars verifieringsnyckel den verifierande använder. Eftersom verifieringsnyckeln inte är hemlig så kan vem som helst som har tillgång till den verifiera de signerade dokumenten.

Problem i samband med digitala signaturer

Ett problem i sammanhanget är att på ett säkert sätt kunna knyta personidentiteter till verifieringsnycklar. Vid verifieringen av ett dokument är det väsentliga vilken *person* som signerat, men det enda som egentligen går att verifiera är vilken *nyckel* som använts. Problemet är inte olösbart, men kräver välfungerande administration av nycklar. I princip går lösningarna ut på att användare och/eller administrativa entiteter med hjälp av digitala signaturer intygar kopplingen mellan andra identiteter och nycklar. Ofta används begreppet public key infrastructure (PKI).

Ett annat problem är att personen som innehar en signeringsnyckel inte själv utför signeringen. Istället är det någon typ av dator eller maskin som utför de omfattande beräkningar som är nödvändiga för att skapa den digitala signaturen. Alltså måste denna maskin också innehålla den hemliga signeringsnyckeln. Det innebär att om maskinen effektivt går att angripa så kan en attackerare komma åt signeringsnyckeln och/eller framtvinga signeringar av dokument som den rätte användaren inte ville signera.

Ytterligare en svårighet i sammanhanget är att digitala dokument i grunden bara består av en bitsträng, en sekvens av ettor och nollor. Vad de betyder och hur de ska tolkas är upp till betraktaren. Om avsändaren (som signerar) och mottagaren (som verifierar) inte tolkar bitsträngen på samma sätt så förändras betydelsen av dokumentet trots att det är helt korrekt överfört och den digitala signaturen stämmer.

Demo

> Ett dokument betraktas i programmet WordPad, och signaturen visar sig korrekt. Betraktad i en annan dator, men med samma WordPad så har 500 kronor blivit 2500 kronor, trots att signaturen fortfarande är korrekt.

Skillnaden beror på att det är olika Windowsversioner som används, där den ena, nyare stödjer döljande av ”dold” text, vilket den andra, äldre inte gör (där krävs det stöd i själva visningsprogrammet, vilket WordPad inte har). Bitsträngen som signerats är oförändrad, men den tolkas olika på de olika datorerna. Tvåan i 2500 är i form av dold text men döljs bara i den nya versionen av Windows.

Modifierade virus

Demo

> Källkoden för ett enkelt, men äkta, virus (Homepage) visas upp, och det demonstreras att en dator med antivirusprogram inte tillåter att viruset startas. En mycket måttligt förändrad version av viruset visas upp och det framgår att denna förändrade version inte känns igen av antivirusprogrammet.

Homepage är inget annat än ett VisualBasic-script som innehåller en enkelt krypterad del, och en dekrypteringsfunktion. Den krypterade delen innehåller all logik för viruset. Det Homepage gör är att det epostar sig själv till alla adresser i alla adressböcker, raderar spåren efter sig själv i epost-programmet och riktar webbläsaren till en pornografisk websida.

Antivirusprogrammet känner igen Homepage, men det visade sig att det var lätt att förändra Homepage så att det inte kändes igen, och inte stoppades av antivirusprogrammet. Den enda förändring som krävdes var krypteringen ändrades något.

Det vi kan lära oss av försöket med att förändra ett existerande virus är att det är mycket lätt att kringgå virusskyddet i vanliga datorer. Detta innebär inte att virusskydd är meningslöst, utan bara att det är lätt att skapa virus som initialt inte hindras av antivirusprogrammen. Med snabba uppdateringar av virusdefinitionerna i antivirusprogrammen så kommer dock de drabbade att utgöra en mycket liten del av den totala populationen av användare.

Det finns en rolig historia som berättas i en mängd olika versioner. I en form så ser den ut såhär: ”Det finns ett talesätt bland safariguider på Serengeti. Om ett lejon attackerar behöver man inte springa fortare än lejonet för att komma undan. Det räcker att springa fortare än den långsammaste turisten.” Logiken påminner om den som gäller avseende virus och antivirus: Så länge inte alla står stilla så kommer de flesta att klara sig undan.

Trådlösa tekniker

Introduktion

Det finns ett flertal trådlösa tekniker i samband med dagens IT-system. Trådlösa tekniker är ofta ett enkelt, snabbt och bekvämt sätt att koppla samman enheter som behöver kommunicera. I samband med tillfälliga nät och nät med rörliga noder så är trådlösa tekniker ofta den enda möjligheten. Den som väljer att använda trådlösa tekniker behöver dock vara

medveten om att det medför större säkerhetsrisker än trådbundna tekniker. När kommunikationen sker trådlöst är det plötsligt möjligt för angripare att ansluta sig till kommunikationen på stora avstånd – långt större än vad många tror.

Att rörliga noder behöver kommunicera trådlöst är uppenbart, men att tangentbord ansluts trådlöst till datorn som står bara några decimeter bort är oftast ett utslag av lättja och bekvämlighet hos användaren. Bekvämlighet är inte nödvändigtvis något fel i sig, men vem tar ansvar för säkerhetsproblemen? Är det rimligt att tro att användaren kommer att göra det när själva anledningen till införskaffandet är att denne inte vill anstränga sig?

Det finns en ständigt konflikt mellan säkerhetsproblem och önskan om att saker ska vara lätta att använda, och det verkar som om denna konflikt börjar om från början varje gång en ny teknik dyker upp. Här är u3-tekniken som beskrivits tidigare ett bra exempel.

Demo

> Datorns powerpointpresentation styrs av en radiofjärrkontroll som kommunicerar med datorn via en liten USB-mottagare. Fjärrkontrollen innehåller knappar för framåt, bakåt, helskärmsläge och några ytterligare funktioner. Det visar sig visar dock att möjligheten att styra datorn går utöver funktionerna i fjärrkontrollen. Det är dessutom så att även utan tillgång till fjärrkontrollen kan en angripare kontrollera datorn via radiogränssnittet.

Bluetooth

Bluetooth är en trådlös teknik för att kommunicera över korta avstånd, typiskt några meter. Ofta handlar det om att koppla ihop en mobiltelefon med någon annan enhet, t.ex. ett head-set, men det kan också vara en handdator som kopplas till en persondator för att utbyta information.

Sedan Bluetooth introducerades första gången har ett antal möjliga attacker presenterats. En del av dessa har angripit svagheter i själva Bluetooth-protokollet och en del svagheter i det underliggande operativsystemet på mobiltelefonen. Sådana svagheter kan användas för att tappa telefonen på information eller för att få den att göra dolda uppringningar och därmed fungera som avlyssningsenhet.

Demo

> En telefon visas upp och angriparen visar sig kunna styra den på avstånd.

Ett argument som brukar framföras är att problemet inte är så stort eftersom Bluetooth bara har en räckvidd på ca 10 meter. Det är alldeles sant så länge man använder de små antenner som finns inbyggda i de bärbara enheterna. Med en större antenn ökar dock räckvidden markant. Inom vissa gränser gäller att räckvidden bara begränsas av budgeten.

Det nuvarande rekordet för Bluetooth-kommunikation med en vanlig handburen elektronisk almanacka är nära två kilometer. Det var ett antal studenter vid Caltech i Californien som satte fast en riktantenn på underredet till ett gevär. Syftet med detta var att kunna använda kikarsiktet för att sikta mot den handdator de kopplade upp sig mot. För att få ideala förhållanden ställde de sig på var sin brygga och fick därmed helt fri luft mellan sig och bra speglingseffekter i vattenytan.

Slutsatser

IT-säkerhet är ett besvärligt ämne. Det är oerhört brett och mångfacetterat, vilket gör det svårt även för heltidsarbetande proffs att ha överblick och kunskap om alla delar. För de allra flesta är det alltså nödvändigt att få hjälp med IT-säkerhetsfrågor i vissa fall.

Samma resonemang leder också fram till att i en organisation behövs regler och policies som stöd för de anställda för att de ska kunna bedriva sin verksamhet på ett tillräckligt säkert sätt, och det är viktigt att alla i organisationen är medvetna om dessa och accepterar att följa dem. Dessa regler och policies måste därför vara rimliga, och kunskap om dem och om IT-säkerhet i allmänhet måste spridas genom olika typer av utbildning. Detta gäller även personer i ledande ställning som sitter på ansvaret för verksamheten.

För att lyckas med detta måste det avsättas resurser, både i tid och pengar, för utrustning, utbildning, uppföljning och kontroller. Säkerhet har den tråkiga egenskapen att den kostar, utan att för den skull bidra med något synbart till verksamheten, annat än frånvaron av de problem som kan uppstå vid bristande säkerhet.

Som medarbetare är det väsentligt att acceptera de regler och policies som finns. Det finns antagligen goda skäl till att de ser ut som de gör. Om det skulle visa sig att någon del av dem är väldigt besvärlig så är det mycket bättre att gå till den säkerhetsansvarige och lägga fram sitt problem. I de flesta fall går det att lösa, åtminstone tillfälligt, på ett bra sätt utan att säkerheten för den sakens skulle blir allt för lidande.

På grund av säkerhetsproblematikens natur så finns det ingen enkel totallösning, ingen ”Super Security 2000” som ger perfekt säkerhet utan problem. När det dyker upp produkter med ungefär sådana påståenden finns det skäl att se upp. Teknik kan hjälpa en bit på vägen mot en god säkerhet, men den är bara en del. En god säkerhet är lika mycket resultatet av ett sinnestillstånd hos de anställda i organisationen. En viss grad av misstänksamhet tillsammans med kunskap ger möjlighet att göra bra bedömningar och avvägningar i de fall regler och policies inte ger klar vägledning.

Referenser

Alla berättelser och exempel från verkligheten kommer från nyhetssidor och sajter om säkerhet på Internet. Pekare dit är inte stabila över tiden, varför vi har avstått från explicita sådana. Det bör dock vara lätt att hitta själv med hjälp av valfri sökmotor och lämpliga sökord från denna text.

Vidare läsning

- http://www.schneier.com/
  Schneiers månadsbrev läses av de flesta säkerhetsintresserade. Sajten innehåller alla gamla brev, och man kan prenumerera.
- http://www.pts.se/internetsakerhet/Sidor/startsida.asp
  PTS’ sida med säkerhetsinformation
- http://www.cl.cam.ac.uk/~rja14/book.html
  En bra, men tjock, bok om säkerhet. I elektronisk form.
- http://sakerhet.idg.se/
  Svensk tidskrift inom området.

Kontaktuppgifter

Om du vill kontakta oss angående vår kursverksamhet, eller av någon annan anledning så kan du nå oss via följande personer:

David Lindahl, david.lindahl@foi.se, 013-37 83 62

Jacob Löfvenberg, jacob.lofvenberg@foi.se, 013-37 81 86

Mikael Wedlin, mikael.wedlin@foi.se, 013-37 80 96

Författare: Jacob Löfvenberg
Kompendieversion: 1.14
Tryck: juni 2008

Versioner

1.0	18 januari 2007	Jacob Löfvenberg	Powerpointkommentarerna har arbetats om till första versionen av kompendiet.
1.1	6 februari 2007	Jacob Löfvenberg	Lagt till text om IP-telefoni och styrkor och svagheter med biometri för säkerhet.
1.11	9 februari 2007	Jacob Löfvenberg	Språkliga justeringar
1.12	2 mars 2007	Jacob Löfvenberg	Minskat styckemellanrummet. Mindre korrigeringar. Förminskat författarnamnet. Första versionen som går i tryck.
1.13	13 mars 2007	Jacob Löfvenberg	Lagt till USB Dumper och hacka u3-pinne
1.14	13 april 2007	Jacob Löfvenberg	Några språkliga justeringar. Gjort punktuppräkning av ”datorns nivåer”
1.14	10 juni 2008	Jacob Löfvenberg	Inga ändringar, bara nytryck.
1.15	15 december 2011	Jacob Löfvenberg	Lagt till demona USB-styrpinne, web-interface hos IP-telefon, bluebug, trojan i. installationsprogram..