IT 09 037
Examensarbete 15 hp
Augusti 2009

Säkerhetsanalys av Windows Server 2008 i Militära System

Teet Sirotkin

Institutionen för informationsteknologi
Department of Information Technology

UPPSALA
UNIVERSITET

Teknisk- naturvetenskaplig fakultet
UTH-enheten

Besöksadress:
Ångströmlaboratoriet
Lägerhyddsvägen 1
Hus 4, Plan 0

Postadress:
Box 536
751 21 Uppsala

Telefon:
018 – 471 30 03

Telefax:
018 – 471 30 00

Hemsida:
http://www.teknat.uu.se/student

Abstract

A Security Analysis of Windows Server 2008 in Military Systems

Teet Sirotkin

This analysis recommends not to use Windows Server 2008 without proper evaluation in any system containing classified information. The reasons are too low assurance and too weak authentication. If Windows Server 2008 is to be used it should be supplemented with a stronger authentication mechanism. The installation should preferably be of Server Core type and the server should be maximally hardened.

Handledare: Björn Victor
Ämnesgranskare: Ivan Christoff
Examinator: Anders Jansson
IT 09 037
Sponsor: MUST

Tryckt av: Reprocentralen ITC

Sammanfattning: I denna analys rekommenderas att inte utan förbehåll använda Windows Server 2008 i något system som innehåller information som omfattas av sekretess. Skälet till detta är bristande assurans och svag autentisering. Om Windows Server 2008 används bör den kompletteras med en starkare autentiseringslösning. Helst bör installationen vara av Server Core-typ och servern bör härdas maximalt.

Nyckelord: Windows Server 2008, Server Core, militära system, försvarssystem, IT-system, försvaret, Försvarsmakten, MUST, informationssäkerhet, datasäkerhet, autentisering, assurans, GTP, Secure Desktop, SITS, CUAP.

1. Inledning

1.1. Bakgrund

En av militära underrättelse- och säkerhetstjänstens (MUST) viktigaste uppgifter är att verka förebyggande för att skydda hemliga uppgifter från att röjas, obehörigen förändras eller förstöras.

Teknikkontoret inom MUST (SÄKK TEK) ansvarar bland annat för att säkerställa att försvarsmaktens IT-system upprätthåller tillräckligt hög säkerhet. Som ett led i detta har man tagit fram en uppsättning krav, Försvarsmaktens Krav på Säkerhetsfunktioner (KSF). Syftet är att tidigt etablera rätt IT-säkerhetsarkitektur och säkerhetsmekanismer så att skyddsvärda uppgifter och verksamhetskritiska system ges ett tillräckligt skydd.

Kraven för ett system skiljer sig beroende på vilken skyddsnivå systemet placeras i. Skyddsnivåerna är definerade av Försvarsmakten och följer internationell praxis (NATO, EU). De fem olika skyddsnivåerna är (i stigande skala) Öppen, Hemlig/Restricted, Hemlig/Confidential, Hemlig/Secret, samt Hemlig/Top Secret. Vilken skyddsnivå ett system placeras i beror främst på vilken informationssäkerhetsklass informationen som lagras/hanteras i systemet är inplacerad i.

1.2. Problemställning

Kraven på säkerhetsfunktionerna är skrivna för att vara generella och specificerar således inte exakt vilka tekniska lösningar som skall nyttjas och inte heller vilka produkter som bör användas.

Inför driftsättning av ett nytt eller uppdaterat IT-system kan då frågan uppstå huruvida en viss produkt kan uppnå tillräcklig säkerhet för att uppfylla kraven för en given skyddsnivå.

Ett sådant exempel är användningen av Windowsservrar inom försvarssystem. Det är ännu inte klarlagt upp till vilken skyddsnivå dessa kan integreras i försvarssystem med bibehållen säkerhet. Denna studie avser att klargöra denna fråga, eller mer generellt:

Hur säkra är Windowsservrar ur ett försvarsperspektiv?

Studien föreslår också motåtgärder mot hoten och blir därigenom även användbar för den som avser driftsätta system innehållande Windowsservrar.

1.3. Avgränsning

MUST håller på att omarbeta sitt ramverk för införande av säkerhet i försvarssystem. Detta innebär att nuvarande krav på säkerhetsfunktioner kommer att uppdateras till Ver 3.0. De nya KSF:erna kommer inte längre omfatta en viss kravmängd per skyddsnivå, utan hänsyn kommer även tas till hotbild och omgivning. Man tar även i beaktande antalet användare, lokalisering, koppling till externa nät, etc. De nya kraven avses även vara tillämpbara på komponenter, och inte enbart system. Det är mot dessa nya krav som Windows Server egentligen borde mätas. Vid tiden för denna analys var dock inte arbetet med dessa krav avslutat, så tonvikten i detta dokument lades på att analysera säkerhetsaspekter hos Windows i allmänhet, och det som är nytt i Windows Server 2008 i synnerhet.

Det finns redan mycket skrivet om säkerheten hos Windowssystem (se lästips och referenser). Denna studie ämnar främst ta tillvara den kunskap som finns tillgänglig och applicera den på typiska försvarssystem. M a o analyseras de säkerhetsegenskaper som främst berör Försvarsmakten, emedan övriga lämnas därhän.

I takt med att det hittas nya, mer eller mindre exotiska, sårbarheter, så pumpar Microsoft ut nya säkerhetsuppdateringar (patchar) som eliminerar dessa sårbarheter. Således är det inte så intressant

att titta särskilt noga på enstaka incidenter (mer än att försöka lära av historien, eller extrapolera frekvensen av framtida sårbarheter). Mera intressant är då att studera de grundläggande säkerhetsfunktionerna och dra slutsatser utifrån dessa. Följaktligen kommer denna studie endast översiktligt belysa enskilda attackmetoder – det finns gott om litteratur om detta – utan istället analysera de mer universella aspekterna.

Att genomföra en analys av säkerheten hos Windows 2008 Server är en grannlaga uppgift. Produkten kommer i flera olika varianter som var och en kan ha en eller flera roller. Dessutom finns ett flertal tjänster som man kan välja att nyttja eller inte nyttja. Således blir det många permutationer – var och en med sina egna säkerhetsaspekter. En server lever inte heller sitt eget liv, utan är beroende av sin omgivning, både den logiska och den fysiska. Den totala bilden påverkas även av de datorer servern interagerar med och det fysiska skyddet som finns tillgängligt.

För att denna studie skall vara genomförbar och meningsfull har en rad avgränsningar och antaganden gjorts. Dessa bygger på hur Windowsservrar typiskt används i försvarssystem.

- Servern är fysiskt skyddad

En server som är fysiskt tillgänglig för en angripare går strängt taget inte att skydda. Det finns en mängd metoder för att exploatera en sådan server. Att starta servern från en CD och cracka adminlösen med t ex L0phtcrack(1), eller via Firewire-porten skriva direkt i kärnan (p g a en svaghet i Firewirespecifikationen) (2), är några exempel. För att denna studie skall vara meningsfull antar vi att alltså att servern befinner sig i en intrångsskyddad miljö.

- Inga kopplingar finns till externa nät

Om det skulle finnas en kanal in/ut till ett nät som man inte har kontroll över, så uppstår en mängd nya frågeställningar. Detta är i synnerhet sant om kopplingen är mot Internet. För att inte glida bort från huvudspåret antas att data endast överförs medelst sekundära media, vilket också är fallet i de flesta försvarssystem inom högre skyddsnivåer.

- Ingen trådlös kommunikation förekommer

Endast Ethernet används på datalänknivå. Trådlös kommunikation förekommer sällan vid högre skyddsnivåer.

- .NET-ramverket är inte installerat

.NET är ett ramverk som är ett stöd för utvecklare och används bl a när man implementerar egna applikationer. Det installeras automatiskt om man gör en defaultinstallation av en Windows Server, (dock inte vid en Server Core-installation) och används i vissa försvarssystem. Behöver man inte explicit använda .NET kan det vara klokt att låta bli att installera ramverket. Assuransen blir låg när .NET används – man har mycket lite kontroll över vad som egentligen händer vid t ex interprocesskommunikationen. En säkerhetsanalys av .NET tarvar en helt egen studie och faller utanför ramarna för denna analys.

1.4. Tillvägagångssätt

Största delen av arbetet bygger på litteraturstudier och intervjuer. Litteratursökning har skett främst via Internet, och då genom Google. Vissa böcker har också studerats (se lästips och referenser). Intervjuer och diskussioner har ägt rum på Institutionen för Informationsteknologi i Uppsala (IIT), MUST SÄKK, Basesoft Open Systems AB, SD-labs och TDC i Örebro. En del experiment har

utförts hos Basesoft, som har provmiljöer med Windowsservrar uppsatta. Här fanns också möjligheter att utnyttja verktyg för penetrationstester och sniffers. Momenten som ingick i studien var följande:

Litteratursökning
Det finns en uppsjö av mer eller mindre seriös litteratur som beskriver datasäkerhet och hacking. Det första momentet bestod i att söka fram relevant litteratur och sålla bort material som inte höll måttet.

Litteraturstudier
Mycket tid lades ner på att förstå hur Microsoft har byggt in säkerhet i sina Windowsservrar och hur sannolikt det är att dessa kan exploateras i försvarssystem.

Intervjuer
En rad samtal har förts med sakkunniga om vad de upplever som brister i Windowssystem, resp vilka hot som är aktuella mot försvarssystem.

Experiment
För experiment och jämförelser av olika installationer gjordes en Server Core installation, resp en full installation, i en labbmiljö. Servrarna installerades med default-inställningar i en virtuell miljö (VMware-server). Endast nödvändiga grundkonfigurationer gjordes, främst avseende nätverksinställningar.

2. Windows Server 2008 – Översikt

Detta kapitel innehåller en översiktlig beskrivning av Windows Server 2008 och Active Directory. Om man redan är välbekant med dessa kan man hoppas direkt till nästa kapitel, annars får man här en bakgrund som gör det lättare att ta till sig innehållet i resten av dokumentet.

2.1. Inledning

Windows Server 2008 bygger precis som sin föregångare, Windows Server 2003, på NT-kärnan (3).

Kodbasen är samma som i Windows Vista och därmed delar de nyheter som BitLocker, Address Space Layout Randomization (ASLR), en väsentligt uppdaterad nätverksstack (med stöd för IPv6), samt förbättringar i kärnan (främst avseende minne och filsystem). Även Windows trotjänar-protokoll SMB (även kallat CIFS) har fått en ansiktslyftning och finns nu i en ny version, SMB2. Även om Microsoft har satsat mycket på att göra sitt operativsystem säkrare, så är andelen ny kod stor, vilket naturligtvis medför risk för att nya sårbarheter har introducerats.

Windows 2008 Server finns i följande varianter (4):

- Datacenter
- Enterprise
- Standard
- Web
- HPC
- Itanium (Intels 64-bitars ILP-processor)

Microsoft-produkterna har mer och mer gått från kompletta servrar till servrar med olika roller. En server kan ha en, eller flera av följande roller. Tre roller har tillkommit i Windows Server 2008.

- Active Directory Domain Services (ADDS)
- Active Directory Federation Services (ADFS) Ny

- Active Directory Lightweight Directory Services (AD LDS) Ny
- Active Directory Rights Management Services (ADRMS) Ny
- Application Server
- DHCP Server
- DNS Server
- Fax Server
- File Services
- Hyper-V1
- Network Policy and Access Services
- Print Services
- Terminal Services
- Universal Description, Discovery and Integration (UDDI)
- Web Services
- Windows Deployment Services

Inom varje roll finns möjlighet att nyttja en mängd olika tjänster. Här följer några exempel:

- Backup
- BitLocker
- Failover Clustering
- Multipath I/O
- Network Time Protocol (NTP)
- Removable Storage Management
- Simple Network Management Protocol (SNMP)
- Subsystem for Unix-based applications
- Telnet Client
- Windows Internet Naming Service (WINS)

För Windows Server 2008 har Windows dessutom tagit fram en avskalad installation, den s k Server Core-installationen, där man tagit bort onödiga tjänster och de flesta grafiska användargränssnitt (GUI). Säkerhetsinställningarna är också lite hårdare åtskruvade.

En Server Core installation kan anamma följande 8 roller:(5)

- Active Directory Domain Services (ADDS)
- Active Directory Lightweight Directory Services (AD LDS)
- DHCP Server
- DNS Server
- File Services
- Hyper-V1
- Print Services
- Web Services

Utöver detta finns möjligheter att installera ADDS som en Read-Only Domain Controller (RODC) i miljöer med dåligt fysiskt skydd. En sådan server bör även kompletteras med BitLocker och Role Distribution, d v s endast lokala administratörsrättigheter och inte administratörsrättigheter för hela domänen. I en RODC lagras alla AD-objekt utom användarlösenord. Klienter kan inte skriva till en RODC. Databasinformation replikeras från en full domänkontrollant. Data går aldrig åt andra hållet. Man kan välja att filtrera bort attribut som inte skall kopieras.

2.2. Server Core installation

Windows operativsystem har alltid varit ett populärt angreppsmål för hackers. En anledning till detta är naturligtvis den stora spridningen, men det finns andra orsaker. På en typisk Windowsmaskin går ett stort antal tjänster och processer igång vid uppstart. Windows har också en mängd grafiska gränssnitt (GUI) som i många fall är integrerade med kärnan. Windows bygger på en stor kodbas, Windows Server 2003 består av 50 miljoner rader kod jämfört med Open Solaris 9.7 miljoner(6). Allt detta sammantaget gör att angreppsytan blir stor.

Som ett svar på detta har Microsoft tagit fram en s k Server Core installation av Windows Server 2008. Här har man avlägsnat alla onödiga tjänster och i stort sett alla GUI, även filutforskaren och startmenyn, dock inte Notepad, Task Manager och vissa kontrollpanelsapplettar. All interaktion sker via kommandoradstolken, eller via Microsoft Management Console från en annan dator (7).

Detta reducerar attackytorna betydligt, men de problem som ändå kvarstår är de som är intressanta att titta närmare på.

2.3. Active Directory

Active Directory, AD, är en katalog som innehåller information om tjänster, resurser och användare inom ett nätverk. Varje instans av dessa representeras i databasen av ett unikt objekt med tillhörande attribut, som bl a beskriver åtkomsträttigheter och säkerhetsinställningar. Scheman används för att beskriva hur attributen ser ut. Objekten organiseras i en hierarkisk struktur som består av en skog, som består av träd, som i sin tur består av domäner. Ofta består skogen av en enda domän. Objekten inom en domän kan placeras i Organizational Units (OU). Denna struktur är avsedd att återspegla den logiska strukturen i en organisation. Det är på denna nivå man normalt sätter rättigheter och gör säkerhetsinställningar, m h a grupp-policies. OUs kan vara nästade, d v s en OU kan ha en annan som förälder. Grupp-policies är objekt i sig och kallas Group Policy Objects (GPOs). Objekten har åtkomsträttigheter knutna till sig, s k System Access Control List (SACL).

Objekt kan också grupperas i sites, som är oberoende av den hierarkiska strukturen och är mera tänkta att avspegla den fysiska miljön.

Operationer på objekt kan loggas. Om man exempelvis lägger upp filer och skrivare i AD:s databas, så kan både lyckade och nekade åtkomster, m m loggas.

Det bör redan här poängteras att hela skogen skall betraktas som en säkerhetsdomän. Om man vill att subjekt i en domän skall kunna utföra operationer på ett objekt i en annan, så måste man upprätta förtroende mellan dessa bägge domäner, s k trust. Om trust nyttjas mellan domänerna i en skog (vilket är default) så har administratörer rättigheter även över domängränserna. Således kan en angripare som har tagit över en domän få en attackyta även mot övriga domäner i skogen. Situationen blir än värre om man upprättar trust mellan skogar, eller domäner i olika skogar. De flesta försvarssystem tycks dock ha en enda domän i en enda skog.

AD installeras på en eller flera Domain Controllers (DC). Varje AD-instans innehåller alla scheman och konfigurationsdata gällande skogen, men endast objekten inom sin domän. Undantaget är dock Global Catalog Servrar, som innehåller (något nedbantad) information om alla objekt. Replikering mellan servrarna sker enl multi-master-varianten och nyttjar Remote Procedure Calls (RPC). Master-slave används dock vad gäller information om scheman, domäner, m m. AD kommunicerar över TCP/IP och nyttjar DNS.

Varje objekt har ett Distinguished Name (DN), en Globally Unique Identifier (GUID), samt i vissa fall ett User Principal Name.

En av AD:s huvuduppgifter är att hantera användare och tilldela dem rättigheter.

2.4. Nätverkstjänster

I Windows Server 2008 ersätts Internet Authentication Service (IAS) med Network Policy Server (NPS). NPS är Microsofts implementation av en RADIUS-server. RADIUS är en inofficiell standard för autentisering och åtkomstkontroll av nätverksresurser. NPS används normalt i kombination med RRAS och NAP för att avgöra om en klient har tillräcklig ”hälsa” för att få nyttja nodens resurser.

Dessa tjänster ingår i servrar med rollen Network Policy and Access Services.

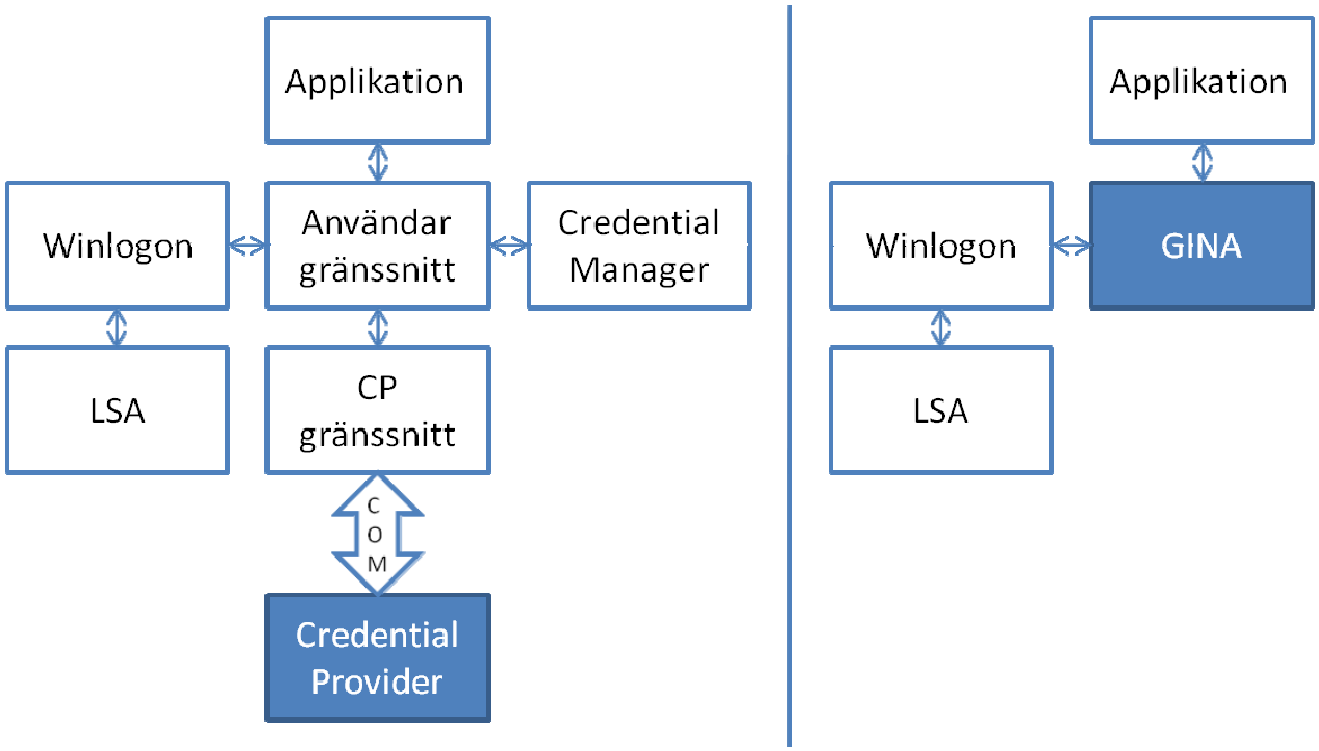
2.5. Behörighetskontroll

När det gäller den logiska vyn av behörighetskontroll i ett informationssystem, så talar man traditionellt om subjekt och objekt. Subjekt utför operationer på objekt. I Windowsfallet kan man se processer som subjekt och resurser som objekt. Resurser kan vara filer, mappar, pipes, tjänster, registernycklar, skrivare, m m. Både subjekt och objekt har säkerhetsattribut knutna till sig. För subjekten rör det sig om (access) tokens och för objekten om security descriptors (åtkomstkontrollistor) .

Behörighetskontroll består av två delar, dels autentisering, där man validerar en identitets äkthet(8), d v s man knyter en identitet till subjekt i systemet, och dels åtkomstkontroll, där man avgör om ett subjekt har åtkomst till aktuell resurs.

2.5.1. Autentisering

Windowskomponenten som har hand om inloggning har t o m XP varit Graphical Identification and Authentication (GINA). GINA är en dll som laddas tidigt i boot-processen och fungerar som ett gränssnitt mellan användaren och Windows inloggningsfunktion (winlogon). GINA är också ansvarig för att starta upp de första användarprocesserna. Microsofts GINA (MSGINA) är utbytbar (eller kompletterbar, s k pass-through GINA) och detta faktum har använts av en del försvarsspecifika lösningar, som har implementerat en egen GINA, t ex för att kunna använda Totalförsvarets Aktiva Kort (TAK) och presentera en egen desktop.
Fr o m Vista (och Windows Server 2008) har dock GINA-konceptet ersatts av Credential Providers (CP), som gör det betydligt lättare att haka in tredje-parts autentiseringslösningar. Denna högre abstraktion innebär dessvärre även att utvecklare förlorar kontrollen över ett viktigt steg i inloggningsproceduren. Tidigare bytte man ut hela Windows inloggningsgränssnitt, d v s den delen som kommunicerar med winlogon (inkl GUI och egna tillägg), och fick således bra kontroll på vad som skedde. Man kunde alltså granska all kod och analysera denna och fick därmed och hög assurans. Numera går det dock bara att lägga till själva autentiseringsfunktionerna (Credential Providers), men inte ersätta GUI eller Credential managers. Dessutom kommunicerade winlogon med GINA via callbacks(9), d v s Winlogon använde funktioner i GINA-biblioteket för att sköta inloggning, skärmlåsning, utloggning och andra inloggningsrelaterade funktioner, och GINA:n kunde kommunicera med Winlogon genom en uppsättning funktionspekare. Numera skapar man istället COM-plugins, d v s utökningar till existerande basklasser för autentisering, vilket har betydligt lägre assurans, eftersom man överlåter kontrollen helt till Windows vad gäller COM-processerna och kommunikationen mellan dem och deras objekt. Detta gör att man aldrig riktigt kan veta hur Microsoft har löst saker och ting och huruvida det här gömmer sig dolda svagheter. Figuren nedan visar konceptuell skillnad mellan CP och GINA.



Figur 1 Konceptuell skillnad mellan CP (Vista) och GINA (XP). De mörka rutorna visar vilka delar man har kontroll över som utvecklare. Pilarna indikerar kommunikationsvägar.

Det bör påpekas att det i framtiden kan dyka upp möjligheter att byta ut en större del av inloggnings-komponenterna, eller att Försvarsmakten kan få tillgång till interna gränsytor om man begär det. Idag finns dock inget publicerat som möjliggör detta.
WS tillhandahåller fyra protokoll för autentisering.

- LM (LAN Manager)
- NTLM (NT LAN Manager)
- NTLMv2
- Kerberos 5

De tre förstnämnda är mindre bra ur säkerhetssynpunkt, då de har väl dokumenterade svagheter(10).

Dessutom har ett antal verktyg för att exploatera dem tagits fram. Det mest kända av dessa är L0ptcrack. Skälen till att Windows inte helt har gått över till Kerberos är naturligtvis kompatibilitet. Kerberos infördes först i och med Windows 2000, resp. XP.

Men även om man har moderna OS i sitt system, så finns det fall där Kerberos ändå inte nyttjas, t ex följande:

- När IP-adresser används istället för DNS eller NetBIOS maskinnamn
- När de kommunicerande maskinerna befinner sig i olika skogar
- När applikationer som nyttjar Windows autentisering inte stödjer Kerberos

I dessa fall används LM/NTLM istället(11). Det är inte klarlagt varför det är på det viset i de två första fallen, men det står helt klart att man lätt kan ha svagare protokoll i sitt system än man trodde.

Det finns inget sätt att alltid tvinga fram användning av Kerberos, t ex genom att sätta en registernyckel.

Dessvärre är inte heller Kerberos 5 fritt från svagheter. Den mest kända svagheten har med förautentiseringen att göra. Det som gör angreppet möjligt är att en tidsstämpel skickas över krypterad med en nyckel som baseras på användarens lösenord. Denna tidsstämpel kan man approximera, eller i alla fall känner till formatet på, och på så sätt utföra en ”known plaintext attack”. Hur en sådan attack går till beskrivs t ex av Frank O’Dwyer (12).

Det finns andra säkrare förautentiseringsprotokoll, t ex tredjepartsprotokollet, men Microsoft har trots detta behållit det något äldre tidsstämpelprotokollet.

2.5.2. Åtkomstkontroll

När en användare autentiserar sig mot systemet (loggar in) skapas ett s k user token. Detta user token innehåller ett antal Security Identifiers (SID:ar) som korrelerar mot var sin (security) principal. En principal är antingen ett användarkonto, en grupp, eller ett maskinkonto. Det finns dock också ett antal special-SID:ar som kommer beröras senare. Varje process som användaren sedan startar associeras med denna användares user token.

SID:ar representeras av 48-bitars tal, där en del talar om vilken typ av konto vi har att göra med (Administrator, Guest, etc) och en del är unik. Den icke-unika delen kallas RID. Den lokala administratören har alltid RID 500 oavsett om man ändrar det populära namnet på kontot. På så sätt kan man alltid identifiera administratörskontot.

Objekt kan ha accesskontrollistor, Discretionary Access Control List (DACL), knutna till sig. Här listas vilka åtkomsträttigheter de olika SID:arna har på det aktuella objektet. Vid åtkomstkontroll jämför Security Subsystem subjektets SID:ar med objektets DACL för att se om åtkomst skall tillåtas.

2.6. Bära med sig Windowsidentiteten

Security Support Provider är en Windows-dll som bl a erbjuder funktionalitet för autentisering till applikationer via gränssnittet Security Support Provider Interface (SSPI). Vid sidan av äldre (och svagare) autentiseringsprotokoll stöds även Kerberos 5. SSPI är en kommersiell variant av GSSAPI (13) med Windowsspecifika tillägg och begränsningar. Kerberos-implementationen följer RFC 1964 vilket gör den tämligen kompatibel.

Windows Kerberos-implementation stödjer bara impersonifiering och inte delegering, även om man m h a restricted tokens numera kan begränsa vad en server får göra. Impersonifiering betyder att servern helt övertar den anropande klientens behörigheter. Delegering innebär att en server endast har åtkomst till aktuellt objekt om den har delegeringsrättigheter *samt* att den anropande klientens har rättigheter till objektet. Detta gör att en attack mot en server inte får lika allvarliga konsekvenser.

Således kan man dra slutsatsen att försvarstillämpningar som nyttjar Windows autentisering har samma egenskaper (brister) som beskrivs för Windows OS-användare i denna analys.

3. Attackmetoder och skydd

3.1. Inledning

När man talar om attacker mot IT-system, så brukar man skilja på insiders och outsiders. De sistnämnda är de flesta i första hand tänker på - en hacker som sitter på sitt rum och via Internet tar sig in i hemliga system. Det är också mot outsider-hotet som de flesta skyddsåtgärder tas – ofta glömmer man bort skyddet mot insiders, eller analyserar det inte tillräckligt. Visst är det så att outsider-attacker generellt sett är vanligare(14) men man måste också ta under övervägande att incidenter som involverar insiders i regel ger upphov till större skada(14). Vad gäller försvarssystem, så skiljer sig dessutom hotbilden något från den generella. Oftast saknas kopplingar till externa nät, och i synnerhet till Internet, så insiderhotet blir därmed det största (enda) hotet. Följaktligen behandlar denna studie i första hand insiderhot. Angreppsmetoderna och motåtgärderna är dock i stor utsträckning de samma. Nedan följer en beskrivning av hur en typisk attack kan gå till. Vi förutsätter att angriparen på ett eller annat sätt har tillgång till nätverket.

Windowsprocesser kan köras antingen i user mode eller i kernel (supervisor) mode. Normalt tillåts endast vissa OS-processer köras i kernel mode, eftersom de har obegränsade rättigheter såväl till mjukvara som till hårdvara. Processer i user mode har betydligt mer begränsningar. Denna separation upprätthålls av hårdvara/firmware.

Följaktligen finns det två angreppssätt, antingen attackerar man kärnan direkt och tillskansar sig på så sätt fullständiga rättigheter, eller så attackerar man ett användarkonto (helst Administrator) för att indirekt skaffa sig rättigheter.

I stort sett går alla attacker i slutändan ut på att komma åt möjligheten att utföra operationer som en användare med så mycket rättigheter som möjligt. På lokala maskiner finns ett antal inbyggda användarkonton där SYSTEM (eller LocalSystem) har mest rättigheter tätt följt av medlemmarna i gruppen Administrators. Kan man erövra något av dessa, så kan man göra allt på maskinen.
I en domän har medlemmarna i grupperna Domain Administrators samt Enterprise Administrators mest rättigheter. Om det föreligger förtroenderelationer (trust) mellan domänerna i en skog, så kan dessa grupper potentiellt ha rättigheter i hela skogen. Eftersom grupperna med mest rättigheter normalt är bättre skyddade så försöker ofta en hacker först attackera ett lättare mål, för att sedan successivt försöka tillskansa sig mer rättigheter i systemet.

När en angripare väl fått tillgång till nätverket, så är första steget en skanning av systemet. Det första en angripare normalt gör är att kolla vilka datorer som finns i systemet. Detta görs enklast genom en s k ping sweep, d v s man försöker pinga olika IP-adresser (eller namngivna datorer) i systemet. De maskiner som svarar vet man är igång. Oftast fungerar inte detta utifrån, eftersom de flesta brandväggar är konfigurerade att filtrera ping, men inne i ett system brukar det gå bättre, eftersom man av underhållsskäl vill kunna pinga noder. Ett undantag tycks dock utgöras av en Server Core installation, som per default inte svarar på ping. Detta är dock till föga tröst, då det finns andra sätt att kolla om en dator lever, t ex genom att skicka ett felformaterat UDP-paket. Normalt får man om datorn lever då tillbaks ett felmeddelande.

Nästa steg är en portskanning. Syftet med detta är att finna vilka portar som är öppna på de tillgängliga datorerna, och gärna vilka tjänster som lyssnar på dessa portar. Det finns flera verktyg som klarar av detta (och mer därtill), varibland några av de mest kända är Nmap (15), Scanmetender(16) och Superscan(17). Denna studie kommer inte gå in djupare på dessa, utan nöjer sig med att konstatera att bra och lättanvända verktyg finns för detta ändamål.

När man talar om angreppspunkter mot datorer, så är det i första hand öppna TCP- och UDP-portar (i transportskiktet) man tänker på. Attackytan är dock betydligt större än så. Nedan tittar vi på

potentiella sårbarheter i protokollen i de andra skikten, såsom datalänk-, nätverks-, och sessionsskiktet.

3.1.1. Datalänkskiktet

I datalänkskikten finner vi bl a Ethernet, PPP och PPPoE. Inga av dessa protokoll är särskilt intressanta att titta på, eftersom Ethernet p g a sin enkelhet inte kan anses utgöra något viktigt angreppsmål i sig och de två sistnämnda normalt inte används i försvarssystem.
Intressantare är då Address Resolution Protocol (ARP), Neighbour Discovery Protocol (NDP) och i synnerhet det för Vista nya protokollet Link Layer Topology Discovery (LLTD)(18).

LLTD är ett protokoll för inhämtning av information om nätverkstopologi och datorers hälsa. Den ingår i Microsoft Rally-konceptet(19) (ett koncept för att förenkla installation och underhåll) och består av en klientdel (mapper), med vilken man kan initiera inhämtningen, och en serverdel (responder), som svarar på begäran. Den senare är implementerad som en kärndrivrutin och är aktiverad efter en defaultinstallation av Vista, resp Windows Server 2008. Eftersom vi här har att göra med ett datalänkprotokoll, så fungerar inte kommunikationen förbi en router. Den fungerar inte heller för datorer med XP och äldre OS, såvida man inte explicit installerat responder. LLTD är tänkt att underlätta arbetet för en nätverksadministratör, men inför också nya säkerhetsaspekter. Eftersom specifikationerna är publika(20), så är det lätt att sniffa och tolka datautbytet om man har fysisk access till nätverkssegmentet. Hoagland et al(21) har visat hur man kan konstruera ett paket som kan få samtliga Responders att svara med sina namn, Ethernet-, IPv4 och IPv6-adresser. Det går dock att få ut mer information än så och även impersonifiera (spoofa) en nod i nätverket.

Den enda omedelbara säkerhetskonsekvensen av ovanstående är att det blir mycket lätt att genomföra en kartläggning av nätverket. Ett visst hot mot tillgänglighet föreligger också, men är inte så relevant i det här fallet, eftersom man antas ha fysisk åtkomst till nätverkssegmentet och då lika gärna kan köra in 230 V i ledningen. Dock – eftersom protokollet är nytt och källkoden är sluten, så är det omöjligt att veta om det föreligger allvarligare gömda sårbarheter. Därför är det en god idé att slå av LLTD responder på alla noder via Group Policies(22).

ARP(23) och NDP(24) är två protokoll som används av Windows för namnuppslagning i länkskiktet, d v s mappning från IP-adress till MAC-adress. Det förstnämnda används för IPv4 och det andra för IPv6. Bägge dessa är nödvändiga för att datakommunikationen skall fungera och återigen kan man konstatera att om man har fysisk tillgång till nätverket, så är det en smal sak att omdirigera trafik till sin egen maskin. Detta belyser hur viktigt det är att kunna lita på end-to-end-autentisering på högre nivå.

3.1.2. Nätverksskiktet

Nätverksstacken i Vista och Windows Server 2008 är nyskriven(25) och integrerar både IPv4 och IPv6 i en och samma stack. Detta innebär att IPv6 inte går att avinstallera. Den går dock att slå av, men observera att man måste göra det via registret för att det ska gälla även för tunnel- och loopback-gränssnitten(26). Både IPv4 och IPv6 är aktiverade efter en defaultinstallation och för att minska attackytan bör man slå av IPv6-funktionaliteten om man inte explicit behöver den. Försök i labbmiljön visade dock att det tycks uppstå (ej utredda) begränsningar i funktionaliteten, bl a vid fjärr-administration.

I Vista och Windows Server 2008 återfinns flera tunnlade protokoll. Det mest intressanta av dessa är det nya Teredo(27), som kapslar IPv6 paket i IPv4 UDP datagram. Det speciella med Teredo är att den (till skillnad från t ex 6to4(28)) kan traversera routrar med Network Address Translation (NAT). Kortfattat går det till så att de kommunicerande parterna upprättar var sin IPv4 förbindelse till en publik Toredo server, som sedan routar trafiken.

Men är inte så lätt att filtrera i en brandvägg, så länge inte brandväggen har stöd för att identifiera

Toredo-trafik.

Vista och Windows Server 2008 stödjer Source Routing(29), vilket innebär att avsändaren av ett IP-paket kan specificera routingen. Detta gör att man under vissa omständigheter utifrån (eller från ett annat subnät) obehörigen kan komma åt interna datorer(30). Source Routing är dock i Vista och Windows Server 2008 per default inställt att ”Not to forward source routed packets”. Något säkrare blir det om man ändrar detta till ”Drop all incoming source routed packets” (31).

SMB2 skiljer sig från SMB genom att antalet kommandon har reducerats från över 100 till 19 (32). Man har även infört pipelining, vilket innebär att en request inte behöver besvaras, innan nästa skickas. Flera anrop kan också slås ihop i ett request. Allt detta gör att nätverkstrafiken minskar. SMB2 är av allt att döma betydligt säkrare än SMB1, men det gäller att komma ihåg att det är SMB1 som nyttjas vid kommunikation med äldre Windowsversioner än Vista och 2008 Server.

3.2. Attacker mot kärnan

Om man kan lägga till eller ändra kod i kärnan kan man i princip göra vad som helst. Det finns två typer av attacker mot kärnan, fysiska och logiska(11). Fysiska attacker sker genom angrepp av drivrutiner som bor i kärnan. Logiska attacker sker mot tabeller, så som GTD, LDT, etc. Attacker mot kärnan är på grund av sin komplexitet inte så vanliga.

3.2.1. Kernel Patch Protection

För att försvåra attacker mot kärnan har Windows tagit fram Kernel Patch Protection (KPP), populärt kallad PatchGuard(33). Detta ger dock inte något fullständigt skydd och flera dokumenterade sätt att gå runt det finns tillgängliga(34). Dessutom kan flertalet antivirusprogram inte köras på datorer med KPP, då dessa program måste kunna modifiera kärnan för att fungera. Av den sistnämnda anledningen har Windows valt att endast inkorporera KPP i 64-bitars-versioner av sina servrar. Eftersom 64-bitars processorer inte ingår i denna studie tittar vi inte mer på KPP

3.2.2. DEP

Data Execution Prevention (DEP) är en funktion som skall hindra applikationer och tjänster att exekvera kod från vissa minnesareor. Detta minskar risken för vissa attacker som har injicerat kod via t ex buffer overflow. Vissa CPU:er har hårdvarustöd för detta, men de är än så länge inte så vanliga inom Försvarsmakten, så vi går inte in djupare på det.

3.3. Attacker mot användarkonton

3.3.1. Scheduler-attack

Om man har tillräckliga rättigheter att köra scheduler på en Windowsmaskin, så kan man lätt anta användaridentiteten SYSTEM (eller LocalSystem). Proceduren är enkel: man schedulerar ett jobb som startar en kommandotolk (cmd.exe). Denna går då igång som SYSTEM. Nu har man fullständiga rättigheter. T ex kan man ta död på filutforskaren (explorer.exe) och starta en ny sådan med fullständiga rättigheter från kommandotolken. På servrar kan man dock normalt bara schedulera jobb om man är administratör. Metoden tillåter dock administratören att ytterligare eskalera sina rättigheter till den högsta nivån SYSTEM, som motsvarar UNIX root. Denna attack provades med framgång på både den fulla installationen, såväl som på Server Core Installationen.

3.3.2. Attacker mot servicekonton

Till skillnad från Windows Server 2003 har även tjänster SID:ar i Windows Server 2008, vilket ökar granulariteten. Tjänster körs ofta som speciella servicekonton som har egna lösenord. Dessa lösenord lagras i klartext i ett ställe i registret som endast LocalSystem har tillgång till. Microsoft har tydligen

konstaterat att kryptering av lösenorden inte skulle ge nämnvärt ökad säkerhet. Detta kan de ha rätt i, då en administratör kan komma åt dessa lösen m h a en scheduler-attack, som vi redan sett. Vad har man för nytta av detta kan man undra, om man redan ”är” SYSTEM? Svaret på detta är att även domänadministratörslösen lagras här och ev även lösen från andra ”trusted” domäner. På så vis kan man alltså även komma åt andra noder i systemet.

3.3.3. Lösenord vs certifikat

Att tillåta användargenererade lösenord är ingen bra policy. Med dagens processorkraft är brute force-attacker lätta att genomföra. T o m Windows Kerberos-implementation går, som vi såg i kapitel 2.5.1, att hacka. En viktig aspekt att ha i åtanke är när man väljer autentiseringslösning är att använda starka nycklar och i synnerhet att distribuera dessa på ett säkert sätt (annan kanal).

3.4. Säkerhetsloggning

Säkerhetsloggning är en viktig funktion för att uppnå spårbarhet, oavvislighet och intrångsdetektering. Windows har en Eventlog som består av tre delar: Application, Security och System Log. I Security Log sparas de säkerhetsrelevanta händelserna i Windows. Applikationer måste logga själva (eller så måste säkerhetskritiska operationer fångas upp) och det är viktigt att man har hög assurans på att de verkligen gör det. Det är två saker man måste se till. Dels måste man slå på loggning av både Success och Failure under Security Policy/Audit Policy och dels måste man samla in loggarna till en central server för att undvika manipulation. Detta gör man t ex med Microsofts egna System Center Operations Manager (SCOM), som tidigare hette Microsoft Operations Manager (MOM). Det finns också tredjepartsprogram för logginsamling som t ex LogEye(35), NetIQ Security Manager(36), eller Open Source-programmet Snare(37). Man kan också använda dedicerad hårdvara, som ArcSight(38).

Default-inställningen på säkerhetsloggen i Windows Server 2008 är ”Overwrite as needed”, vilket står i strid med MUST krav på säkerhetsfunktioner, där man skriver att loggar inte får skrivas över. Således bör detta ändras till ”Do not overwrite events”. Detta räcker dock inte, utan man måste även ändra policy till ”Shut down system immediately if unable to log security audits”, annars kastas helt enkelt de nya loggposterna. Sedan är det upp till logginsamlingen att möta tillgänglighetshotet och se till att loggarna insamlas innan loggen blir full.

4. Härdning av Windows Server 2008

4.1. Inledning

Härdning av en dator betyder egentligen bara att man genom olika åtgärder gör datorn mindre sårbar. Det finns en hel del att göra vad gäller härdning efter en Windows Server 2008-installation. Ett antal verktyg och mallar finns att utgå ifrån, men man kommer inte ifrån att det krävs en hel del handpåläggning för att uppnå en bra konfiguration för en given server.

Ofta talar man om följande åtgärder för att förbättra konfigurationen:

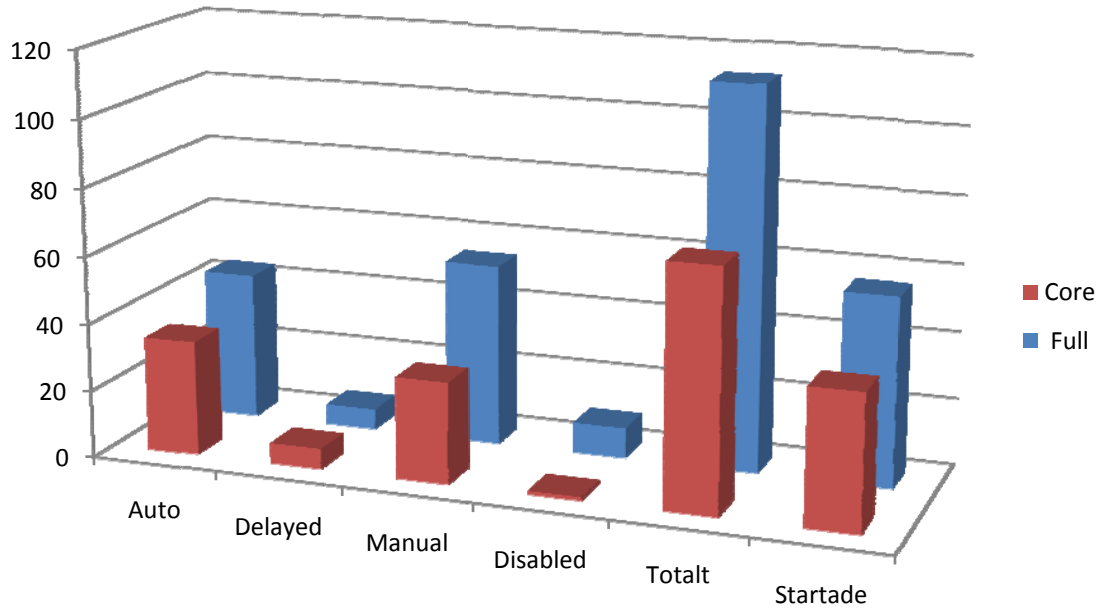
- Stänga av alla icke nödvändiga tjänster
- Konfigurera brandväggen att inte släppa genom mer än nödvändigt
- Hålla datorn uppdaterad med säkerhetsuppdateringar
- Ge användare så låga rättigheterna som möjligt (jfr ”need to know”)

Det finns flera metoder för att uppnå detta, t ex genom säkerhetsmallar (Security Templates), Group Policy Objects, verktyg för säkerhetsuppdateringar, eller manuella handhavanden (t ex registerinsällningar) m h a checklistor.

Ofta kombineras dessa metoder och det finns som alltid i Windows flera sätt att uppnå samma resultat. Gemensamt för metoderna är att man strävar efter att minska angreppsytan genom att stänga av onödiga tjänster, skruva åt rättigheter och ändra säkerhetsinställningar.

4.2. Tjänster

Det är mycket viktigt att verkligen stänga av alla tjänster man inte behöver. Något förenklat kan man säga att om det inte går några tjänster på en maskin, så finns inga angreppsytor. Det är helt enkelt ingen som lyssnar. En jämförelse gällande exekverande tjänster visade att Server Core installationen hade 40 tjänster igång, emedan den fulla installationen hade 56. Skillnaden var alltså bara 16 tjänster, eller 11 % färre hos Core Installationen. I Server Core installationen var 71 tjänster installerade, jämfört med 113 hos den fulla installationen. Hur många som var satta till automatic (går igång vid uppstart), delayed (startar efter fördröjning), manual (startas först när tjänsten behövs) resp disabled (helt avstängd), framgår i figuren nedan.



Figur 1 Jämförelse mellan uppstart av tjänster i en Core, resp Full Windows 2008 Server installation.

Skillnaden avseende tjänster mellan en Server Core installation och en full installation kan tyckas vara liten. Microsoft själva hävdar dock att 70 % av de sårbarheter som upptäckts hos Windows de senaste fem åren inte skulle ha varit applicerbara på en Server Core installation (39). Oavsett hur mycket vikt man ska fästa vid den siffran så är det ändå en god idé att välja en Server Core installation framför en full, om man inte har en bra motivering till varför man behöver en full installation.

En skanning med Nmap i labbmiljön precis efter default-installation ger vid handen att en full installation har 9 portar öppna, medan en Server Core installation har 4.

Full installation

PORT	STATE	SERVICE
53/tcp	open domain	Domain Name System (DNS)
88/tcp	open kerberos-sec	Kerberos—authentication system
135/tcp	open msrpc	Microsoft EPMAP (End Point Mapper) also known as DCE/RPC Locator service, used to remotely manage services including DHCP server, DNS server and WINS
139/tcp	open netbios-ssn	NetBIOS NetBIOS Session Service
389/tcp	open ldap	Lightweight Directory Access Protocol (LDAP)
445/tcp	open microsoft-ds	Microsoft-DS Active Directory, Windows shares
49154/tcp	open	unknown
49155/tcp	open	unknown
49157/tcp	open	unknown

Server Core Installation

PORT	STATE	SERVICE
135/tcp	open msrpc	Microsoft EPMAP (End Point Mapper) also known as DCE/RPC Locator service, used to remotely manage services including DHCP server, DNS server and WINS
445/tcp	open microsoft-ds	Microsoft-DS Active Directory, Windows shares
49153/tcp	open	unknown
49155/tcp	open	unknown

Detta säger dock inte så mycket eftersom flera tjänster kan lyssna på samma (höga) port via processen svchost. Eftersom tjänster i modernare Windowsversioner är implementerade som dll:er (bibliotek) så måste de laddas av en process, med det generiska namnet svchost. Tjänsterna är logiskt grupperade och varje grupp körs som en instans av svchost. För att se vilka tjänster som körs under vilken svchost, så kan man använda kommandot `tasklist /SVC` (40).

Det finns även skillnader mellan de bägge installationerna i brandväggskonfigurationen. Bl a blockeras ICMP protokollet per default i en Server Core installation, vilket medför att servern inte svarar på ping. ICMP har tidigare använts bl a vid buffer overflow-attacker, t ex Ping of death POD (41), samt vid DoS-attacker, t ex Ping Flood(42).

4.3. Patchning

Bland försvarssystem förekommer det att man ”fryser” ett system i och med att systemet blivit granskat, ackrediterat och erhållit ett driftsgodkännande. Det är ju en specifik version som erhållit godkännandet och man vill inte riskera att genomgå processen på nytt p g a förändringar i systemet. Detta tolkas ibland som att man inte heller bör installera säkerhetsuppdateringar.

Ett system är ju dock i viss mening dynamiskt i alla fall, och att frysa system är ingen bra idé annat än i enstaka specialfall (små kritiska stand-alone-nät). Det är naturligtvis viktigt att hålla sina servrar (och för all del även klienter) uppdaterade m a p på säkerhetsuppdateringar. Dock bör man ha en genomtänkt och (av MUST) godkänd rutin för detta. Ofta används Windows Server Update Services (WSUS)(43) för detta ändamål. WSUS är tjänst för att distribuera patchar lokalt i systemet. Servern installeras på en Windows 2008 maskin, men kan administreras via WSUS Administration Console som kan köras från någon annan dator i domänen.

På en stand-alone dator kan man istället för WSUS använda Microsoft Baseline Security Analyzer (MBSA)(44).

Det är såklart olämpligt att koppla ett försvarssystem till Internet, så lösningen är normalt att nyttja en extern internetansluten WSUS server, varifrån man manuellt flyttar patcharna till den interna servern. Detta sker normalt endast på andra tisdagen varje månad, den s k “patch Tuesday”(45). Observera att detta introducerar en attackyta. Rutiner måste finnas för att säkerställa att man inte får in skadlig kod via detta gränssnitt. Det är också viktigt att säkerställa att patchningen inte inför tillgänglighetsproblem och förändrar säkerhetsegenskaperna. En risk med patchning är att vissa åtskruvade säkerhetsinställningar kan återställas till (de mindre säkra) defaultinställningarna. Efter varje patchning rekommenderas därför att man genomför en verifiering av säkerhetsinställningarna, helst automatiserat. För större och kritiska system bör man dessutom prova patchning/verifiering på ett referenssystem innan man ger sig i kast med det skarpa systemet.

4.4. Checklistor och mallar

4.4.1. Security Templates

Om man planerar att driftsätta många servrar kan det vara värt att skapa egna Security Templates(46). I dessa kan man specificera inställningar för tjänster, lösenordspolicies, åtkomsträttigheter (sätta ACL:ar), samt mycket annat. Till skillnad mot tidigare Windowsversioner finns i Windows Server 2008 inga färdiga security Templates(47), men det finns mallar som man kan utgå ifrån och dessa är:

- %systemroot%\inf\Defltsv.inf (för alla servrar)
- %systemroot%\inf\Defltdc.inf (för domänkontrollanter)

Inf-filer är inget man hackar i för hand utan istället används verktyget MMC med Security Configuration and Analysis snap-in. Det går även använda Group Policy Management Console (GPMC), eller om man kör lokalt på en Server Core Installation, kommandoradsverktyget SecEdit.

Det tycks inte finnas bra tredje-parts Security Templates att utgå ifrån, men väl ett antal bra checklists. Se nedan.

4.4.2. Checklistor

Det finns ett antal myndigheter och organisationer som erbjuder checklistor för härdning av Windows. På senare tid har de dock blivit mycket lika, eller identiska. Så egentligen räcker det med att titta på Windows egna riktlinjer(48).

Microsoft själva erbjuder gratisresurser som de benämner Solution Accelerators(49). Här ingår bl a Windows Server 2008 Security Guide(50), samt Security Compliance Management Toolkit(51). Den sistnämnda innehåller GPOAccelerator tool, med vilken man kan man skapa de Group Policy Objects (GPO:s) som man behöver för att göra sina säkerhetsinställningar. Använder man detta verktyg, så får man spårbarhet på vad man har gjort och därmed reproducerbarhet.

Microsoft definierar tre olika nivåer för säkerhetsinställningar:

- Enterprise Client (EC)
- Stand-Alone (SA)
- Specialized Security Limited Functionality (SSLF)

Den sistnämnda är den som är intressant för försvarssystem, eftersom den har högst skyddsnivå.

Även The Center for Internet Security (CIS) har en uppsättning “benchmarks” d v s en samling

rekommenderade säkerhetsinställningar(52). Dessa är mycket lika Microsofts egna och gäller i skrivande stund endast för Windows 2003, XP och äldre operativsystem.

Mera intressant är kanske The National Institute of Standards and Technology (NIST) checklistor(53). Dock har inte heller de ännu någon lista för Windows Server 2008, däremot både för Vista och XP(54).

The Defense Information Systems Agency (DISA) har listor som är specifika för amerikanska försvaret(55) som täcker även in Windows Server 2008. Listorna från National Security Agency (NSA)(56) är identiska med Microsofts egna.

Det finns dock fler mindre officiella checklistor att välja på om man vill säkra upp sin server. En av de bästa är den som förekommer i Appendix A i Hacking Exposed Windows(11).

5. Alternativ till Windows Server

Även om man har ett helt nätverk bestående av Windowsdatorer, så är det fullt möjligt att byta ut en eller flera av Windowsservrarna mot andra alternativ med bibehållen funktionalitet. Att närmare gå in på alla dessa alternativ ligger utanför ramarna för denna studie, men det känns ändå meningsfullt att belysa scenariot med exempel. Vi antar att vi behåller Windows klienter och applikationsservrar, men vill ersätta AD-servern, eller en hel domänkontrollant.

5.1. Samba

En väg att gå är att använda Samba(57), som är en öppen källkods-implementation av Windows nätverksprotokoll smb. Att nyttja öppen källkod har sina säkerhetsfördelar: assuransen ökar eftersom man har tillgång till källkoden och därmed kan granska koden och kompilera själv. Man slipper också hela problemet med ”security by obscurity”, d v s att leverantören bygger delar av säkerheten på att ingen ska känna till vad som händer under huven. Men det finns också nackdelar: ofta är konfigurationsstyrning och kvalitetssäkring sämre, eller i alla fall inte dokumenterad. Samba kan köras på de flest UNIX och Linux-dialekter, och det är här den största fördelen ligger. Detta innebär att man skulle kunna köra Samba i ett Windowsnätverk på ett säkert OS, som t ex Trusted Solaris(58), som är Common Criteria(59) certifierad på assuransnivå EAL4+.

Att köra en UNIX/Linux-maskin som domänkontrollant i en Windowsmiljö har på senare tid börjat bli ett fullt realistiskt alternativ. Redan idag kan en UNIX-server som kör Samba 3.3 integreras i en AD-domän som vilken Windows Server som helst. Dessutom planerar Samba-projektet att under sommaren 2009 släppa version 4.0, som lär kunna ersätta en Windows Server som domänkontrollant helt och hållet. Det skall t o m gå att nyttja verktygen ur Adminpak.msi med Samba 4. Ett alternativ till Samba är Samba-TNG(60).

6. Alternativ till Windows autentisering

Om man avser behålla Windowsservrar i sitt nätverk, men ändå vill förbättra säkerheten med avseende på behörighetskontroll, så finns det möjligheter att lägga en annan autentiseringslösning ovanpå Windows egna. Det finns ett antal försvarsspecifika produkter som innehåller andra autentiseringslösningar och som redan används inom Försvarsmakten. Det ingår inte i denna studie att göra en jämförelse mellan alla dessa, eller att framhålla något framför ett annat, utan endast att ge en kort presentation av några av alternativen. De produkter vars autentiseringslösningar har studerats är:

- Generell Teknisk Plattform (GTP)
- Secure Desktop (SD)
- Säker Inloggning Terminal Server (SITS)
- Comex User Authentication Package (CUAP)

GTP, SD och SITS används idag av system med högre skyddsnivåer och CUAP används i system med lägre skyddsnivåer. Ingen av lösningarna stödjer idag Windows Server 2008. Värt att notera är att fr o m Windows 2008/Vista har Microsoft frångått konceptet med GINA (se ovan). Som en konsekvens av detta måste egenutvecklade GINA-moduler porteras till Credential Provider-modellen. Det är också viktigt att känna till att ingen av lösningarna per automatik tillhandahåller ett av Försvarsmakten godkänt textskydd (kryptering) för högre skyddsnivåer. Detta innebär att all kommunikation som går utanför säkerhetsdomänen (t ex om klienter och servrar finns på olika orter) måste tunnlas genom av TSA godkänd kryptoutrustning, t ex Kryapp 980.

6.1. GTP

Generell Teknisk Plattform (GTP) är en säkerhetsgranskad och kryptoverifierad säkerhetslösning, utvecklad av Logica och Basesoft, som innefattar behörighetskontroll (autentisering och åtkomstkontroll), säkerhetsloggning, skydd mot skadlig kod m fl tjänster för system med högre skyddsnivåer. GTP har en operativsystemoberoende säkerhetsarkitektur och stödjer autentisering av användare, tjänster/program och datorer. Idag stöds lokal inloggning i Windows XP, Windows Server 2003 och Solaris, samt domäninloggning i Windows (AD). Dessutom stöds inloggning, som fjärrterminal, till server med Citrix, Windows Terminal Server, telnet, m fl. GTP stödjer inloggning med TAK, TEID samt mjuka certifikat/lösenord. För stark autentisering krävs TAK. Lösenordsinloggning får endast nyttjas i öppna system. Stöd för Single-Sign-On (SSO) finns för ovan uppräknade miljöer. Detta innebär att användaren endast behöver autentisera sig en gång med sitt TAK, och kan därefter nå de miljöer, datorer och tjänster som hon är behörig till.

Nedan följer ett förenklat användningsfall avseende behörighetskontroll:

Inloggning:

1. Användaren stoppar in sitt TAK i kortläsaren.
2. SecL (egenutvecklad GINA) skickar certifikatet till säkerhetsservern för kontroll.
3. Om certifikatet godkänns, så görs en åtkomstkontroll mot GTP säkerhetsdatabas.
4. Säkerhetsservern returnerar rollinformation (roller, möjliga konton) till SecL.
5. Användaren väljer och signerar sitt roll-/kontoval och skickar till säkerhetsservern.
6. Säkerhetsservern returnerar inloggningsinformation både för GTP och för Windows.
7. Användaren dekrypterar detta med sitt TAK och PIN genom sin privata nyckel.
8. Autentisering sker till GTP – säkerhetsservern returnerar certifikat, sessionsnycklar och

gällande behörigheter signerat av säkerhetsservern.

9. Inloggning sker i Windows och användaren får sina Windowstillämpningar startade (desktop, filutforskare m m).
10. Loggning sker i varje säkerhetsrelaterat steg till GTP loggserver.
11. Om allt gått bra (inkl loggning), är användaren både autentiserad i GTP och inloggad i Windows

Användning av kommersiella program - filåtkomst:

12. Användaren startar nu någon tillämpning, t ex Microsoft Word. Word ärver både GTP och Windows autentisering respektive aktuella behörigheter.
13. Word behöver en fil att arbeta med, den ligger i ett säkert filsystem, varvid användarens TAK-autentisering och GTP behörigheter används för att kontrollera åtkomst och sedan skydda överföringen från filservern med det säkra filsystemet till Word på klienten. Motsvarande gäller för alla program som arbetar med filer.

Användning av kommersiella program - kommunikation:

14. Användaren startar en tillämpning som nyttjar internet standardprotokoll (TCP/IP), t ex en webbrowser (andra program såsom e-post fungerar på motsvarande sätt). Webbrowsern ärver användarens autentisering i GTP med tillhörande aktuella behörigheter.
15. När Webbrowsern försöker använda TCP/IP över nätet till en webserver, så fångas detta upp i TCP/IP stacken av en GTP LSP (61).
16. Efter behörighetskontroll skapas en sessionsnyckel (av säkerhetsservern) för skydd av kommunikationen.
17. Webservern kontaktas med erhållen sessionsnyckel, varpå behörighetskontroll sker mot säkerhetsservern. Om åtkomst är tillåten, delegeras användarens ursprungliga TAK-autentisering och behörigheter till en process i webservern.
18. Denna webserver-instans kan nu försöka komma åt en URL, t ex en PDF-fil i det säkra filsystemet. Denna kräver för åtkomst återigen korrekt GTP autentisering och behörighet. Om denna åtkomstkontroll gick bra överförs filen skyddat tillbaka till användarens Browser.

Användning av specialutvecklad applikation:

19. Klienten ärver sin autentisering från inloggningen (SSO). Servern autentiserar sig mot säkerhetsservern med en nyckel-fil.
20. När klienten önskar samverka med servern sker en ömsesidig autentisering mellan dessa. Om de är behöriga att samverka så distribuerar säkerhetsservern en krypterad tidsbegränsad sessionsnyckel till respektive part.
21. Klienten anropar tjänst/metod hos servern. Kommunikation skyddas med tidsbegränsad sessionsnyckel (som erhållits från säkerhetsservern) för just denna samverkanskanal.
22. Behörighetskontroll och loggning baserad på autentiseringen och aktuella behörigheter sker vid ingången till tjänsten när klienten anropar respektive funktion hos applikationen.

    Såväl inhämtning av sessionsnycklar, åtkomstkontroll och loggning sker normalt utan manuellt inkodade anrop till GTP säkerhetstjänster från applikationen. GTP tillhandahåller ett utvecklingspaket som medger bearbetning av gränssnittsbeskrivningar, IDL(62), så erforderliga anrop sker automatiskt. Alternativt används av aktuellt middleware understödda ”hooks” för att integrera GTP säkerhetsfunktioner. Tillämpningsutvecklaren kan även själv anropa GTP programmeringsgränssnitt (API) från olika programspråk såsom C++, Java och C. Vidare kan GTP nyttjas från andra ramverk, såsom WS-Security(63) genom att konfigurera den att nyttja GTP Kerberos.

Komponenter (tjänster, program) i GTP underkastas samma behörighetskontroll som applikationerna. Kommunikationen mellan olika delar i GTP skyddas på samma sätt som för applikationerna med sessionsnycklar och symmetrisk kryptering. Olika versioner av GTP stödjer olika kombinationer av krypteringsalgoritmer inklusive urval av TSA implementerade algoritmer. Fildelningsfunktionen stödjer f n endast kommersiella algoritmer.
GTP innehåller en Kerberos 5 server (utökad med fält för behörighetsinformation och spårbarhet) som till skillnad från Windows Kerberos-implementation vid förautentiseringen nyttjar tredjepartsprotokollet i stället för det svagare tidsstämpelprotokollet och har säkrare generering och distribution av nycklar och behörighetsinformation. GTP stödjer även delegering vilket ger stöd för tillämpningar som samverkar i flera steg. En server kör normalt med låga behörigheter och får bara åtkomst till aktuellt objekt med sina (delegerings-) behörigheter i kombination med den anropande klientens behörigheter.
Detta är säkrare än Windows impersonifiering, vilket innebär att servern får användarens alla rättigheter. En sådan säkerhetsarkitektur gör en attack mot servern mindre allvarlig.

6.2. SD

Secure Desktop (SD) är en säkerhetslösning utvecklad av SD-labs som innefattar autentisering, åtkomstkontroll och säkerhetsloggning för system med högre skyddsnivåer. SD nyttjar tunna klienter med Terminal Services eller Citrix. Ömsesidig autentisering mellan klient och en proxyserver (Linux) i servernätet sker m h a en Stunnel kompilerad med Försvarsmaktens SSL-bibliotek (FMSSL). Stark autentisering erhålls därmed från klient fram till proxyserver. Grundtanken i SD är att så lite som möjligt finns och behövs på klienten och att Windows används till så lite som möjligt, framför allt inte för åtkomstbeslut.

Nedan följer ett förenklat användningsfall:

1. Användaren ansluter sin hårddisk (som hanteras som hemlig uppgift enligt Försvarsmaktens regler) till en klient-PC och startar upp systemet. Innan inloggning skickas en slumpsträng krypterad med användarens publika nyckel (challenge) till TAK-kortet. Användaren slår sin PIN, slumpsträngen dekrypteras med den privata nyckeln och returneras (response). Om allt gick bra sätts ett slumpat lösen och som man sedan loggar in med. Detta sker internt i minnet på GINA-processen, ett minne som skrivs över efter att lösenordet använts.
2. Användaren får upp en specialutvecklad (klient)desktop och klickar på ”Anslut”, varpå ICA (Independent Computing Architecture)-klienten ansluter till en port på ”local host”, där Stunnel(64) lyssnar.
3. Stunnel skapar en ömsesidigt autentiserad förbindelse mellan klient och proxyserver enligt TLS specifikation RFC 2246(65). Detta sker på följande sätt:
4. Klient och proxyserver kommer överens om SSL-version och Ciphersuit, samt utbyter certifikat.
5. Proxyservern kontrollerar att användaren är upplagd i SD användardatabas och att klientens certifikat är signerat av TSA, giltigt och ej spärrat.
6. Proxyservern verifierar att klienten har den privata nyckeln som hör till certifikatet genom att användaren på nytt ombeds knappa in sin PIN.
7. Klienten kontrollerar att servercertifikatet är signerat av TSA, giltigt, samt betrott (att servercertifikatet finns i klientens godkännandelista).
8. Klient och proxyserver skapar tillsammans en sessionsnyckel med Diffie-Hellman-

protokollet och en TLS-tunnel (AES256) är därmed upprättad.

9. Sedan upprättas en förbindelse mellan proxyservern och terminalservern. Denna är okrypterad och man litar till skalskyddet, d v s att servernätet är fysiskt skyddad.
10. ICA-klienten kontaktar Citrix presentation server på terminalservern. Citrix-servern startar upp en specialutvecklad (server)desktop. Användaren loggas in i AD med personnummer som användarnamn och ett lösenord som hämtas från klientens hårddisk.
11. Serverdesktop begär IP-nummer samt personnummer på den nya uppkopplingen från Stunnel på erhållen IP/port på proxyn. Om inte personnummer från proxy överensstämmer med Windows inloggningsnamn loggas detta och windows-logout sker.
12. Serverdesktop verifierar mot säkerhetsservern att användaren får logga in från aktuell IP-adress. Om detta är ok tilldelas användaren ett slumpat sessionsnummer, som lagras tillsammans med personnumret i SD:s databas (MS SQL) på säkerhetsservern.
13. Startmeny och desktop populeras utifrån användarens rolltillhörighet, som hämtas från SD:s databasen.
14. Varje gång en applikation som är utvecklad med SD SDK vill göra åtkomstkontroll kontrolleras identiteten genom att en funktion anropas med sessionsnumret som inparameter. Sessionsnumret mappas till personnummer via sessionsinformationen i säkerhetsservern och kontrolleras via en lagrad procedur. Om användaren som hör till sessionsnumret är densamma som den i Windows inloggade användaren så returneras personnumret. Omautentisering sker varje timme, d v s punkt 4-8 samt kontroll enligt punkt 11.

Nyttjande av Secure Desktop innebär inte med automatik att applikationer kan köras utan uppsäkring. De anpassningar som måste ske av verksamhetsapplikationer är att säkerställa att de exekverar i en terminal-server-miljö på ett sätt som inte äventyrar säkerheten eller separationen mellan användare. Naturligtvis måste verksamhetsapplikationen även uppfylla kraven på vald säkerhetsnivå (d v s inte använda AD, utan SD:s behörigetskontroll, eller motsvarande). En applikation utvecklad med SD SDK får behörighetskontroll och säkerhetsloggning via ramverket för skapande av lagrade procedurer.

6.3. SITS

Säker Inloggning Terminal Server (SITS) är en säkerhetslösning utvecklad av Steria som innefattar autentisering vid inloggning och utskrift för system med högre skyddsnivåer. Även SITS nyttjar tunna klienter (Terminal Services eller Citrix). Klienten består av en disklös arbetsstation där Windows XP embedded startas upp från CD och (efter att PIN-kod avgetts) kopplar upp sig mot en Windows Terminal Server, där en egenutvecklad GINA, SITS-GINA, påbörjar domäninloggning. En agent som exekverar på domänkontrollanten skickar det krypterade lösenordet till TAK, där det dekrypteras och returneras till GINA:n, varpå GINA:n kan genomföra domäninloggningen.

Kommunikationen mellan klient och terminalserver, resp terminalserver och applikationsserver skyddas med ett 3-DES-rör. Eftersom lösenordet för domäninloggning skickas okrypterat till terminalservern är det viktigt att detta rör är säkert. Det är inte i skrivande stund inte helt klart hur sessionsnyckeln skapas och vilken implementation av 3-DES som används.

På samma sätt som för t ex SD måste det användande systemet se till att övrig trafik mellan servrar i servernätet skyddas och att egenutvecklade applikationer nyttjar autentiseringen på ett av MUST godkänt sätt.

6.4. CUAP

CUAP (Comex User Authentication Package) är en säkerhetslösning som utvecklas av Comex och bl a innefattar autentisering och signering av dokument. CUAP är godkänd att med TEID användas i system med lägre skyddsnivåer.

7. Slutsatser

Ett av de stora problemen med att använda Windows säkerhetsfunktioner är bristen på assurans. Det är inte helt lätt att få tilltro till det som händer under skalet. Detta är också anledningen till varför man bör lägga en annan autentiseringslösning ovanpå Windows egna. Med en egen GINA fick man i alla fall hyfsad assurans på inloggningen, men med CP tvingas man lita än mer på Windows. Detta är i synnerhet på grund av att COM nyttjas. Det bästa är att inte alls lita till Windows behörighetskontroll, utan använda ett helt separat behörighetskontrollsystem och se till att det verkligen ”tar” från användarens token fram till objektet som accessas, d v s att man inte i något steg litar på Windows användarseparation, eller filskydd.

Att Windows Server 2008 innehåller mycket ny kod sänker också assuransen.

Windows 2008 Server är i många hänseenden säkrare än sina föregångare, men dock bör ändå beakta att många gamla välbeprövade attackmetoder kvarstår. Även om det hjälper, så kan man inte känna sig säker ens med installationer av Server Core-typ och maximalt härdade servrar (d v s där man skruvar åt maximalt och sedan öppnar upp bara det som behövs). Ju mer man använder av Windows funktioner (.NET, NetBIOS, m m) ju mer (portar) måste man öppna upp.

Huvudproblemet är bakåtkompatibiliteten. Om man har äldre Windowsmaskiner i sitt system, och i en del andra mindre uppenbara fall (När IP-adresser används istället för DNS eller NetBIOS maskinnamn, t ex) så nyttjas ofta äldre och svagare protokoll (som NTLM). Även för den nyare Kerberos-implementationen finns det dock crack-verktyg.

Dessa faktorer sammantaget gör att man inte utan förbehåll (d v s alternativa åtgärder) bör lita till Windows egna autentiseringslösning i några säkerhetsklassade system.

8. Vidare studier

En planerad uppföljning till denna studie är en mätning av ett typiskt Windows Server 2008-baserat system mot KSF 3.0. Denna studie kan påbörjas först efter att KSF 3.0 har fastställts, vilket planeras ske hösten 2009.

Många väljer Windows bl a på grund av att man vill utveckla applikationer med .NET-ramverket. Det förenklar utvecklingen, men gör också att man lämnar över mycket av kontrollen till Windows. Assuransen blir därmed låg, i synnerhet om man beaktar att .NET är ett gigantiskt ramverk (tar upp ca 20 Gb vid installation). En intressant analys vore att mäta vilken styrka och assurans man kan komma upp till om man utvecklar en applikation med .NET.

En djupare analys av andra säkerhetsfunktioner än autentisering är en annan intressant väg att gå.

9. Acknowledgement

Tack till alla som bistått mig i arbetet med denna studie: Jens Bohlin (MUST), Ivan Christoff (IIT), Jan Danielsson (Basesoft), Jaan Haabma (Basesoft), Ole Nilsson (MUST), Lars Nordin (SD-labs), Mats Pettersson (Comex), P-O Risberg (Logica), Tommy Rausberg (TDC), Pelle Säfstöm (SD-labs), Håkan Söderholm (Steria), Björn Victor (IIT), Jan Wünsche (MUST).

10. Ordlista

Assurans	Tilltro till att produkten/komponenten gör det den utgör sig för att göra
Autentisering	Validering av en identitets äkthet(8)
FMSSL	Försvarsmaktens implementation av OpenSSL. Integrerad med KrAPI och stödjer därmed TAK.
Förstärkt inloggning	Av MUST definierad kravnivå för system som bl a föreskriver att enbart lösenordsinloggning ej får användas och att systemet skall vara kryptokontrollerat, säkerhetsgranskat, penetrationstestat och källkodsgranskat m a p säkerhetskritiska komponenter av en oberoende instans.
H/C	Informationsäkerhetsklass Hemlig/Confidential.
H/R	Informationsäkerhetsklass Hemlig/Restricted.
H/S	Informationsäkerhetsklass Hemlig/Secret.
H/TS	Informationsäkerhetsklass Hemlig/Top Secret.
KrAPI	Krypto API. Gränssnitt som bygger på PKCS #11(66), med ett antal tillägg, begränsningar och förändringar. Används bl a för att kommunicera med TAK.
PIN	Personal Identification Number
RADIUS	Remote Authentication Dial In User Service.
SSL	Secure Socket Layer
Stark autentisering	Av MUST definierad kravnivå för system som bl a föreskriver att algoritmer och protokoll godkända av TSA skall användas och att all information som utväxlas i starkt autentiserad session eller motsvarande skall knytas till rätt avsändare med hjälp av en godkänd kryptografisk mekanism. Systemet skall vara säkerhetsgranskat, av en av TSA godkänd oberoende instans, motsvarande CC EAL4+. TAK bör användas.
TAK	Totalförsvarets Aktiva Kort. Innehåller bl a två 2048 bitars privata RSA-nycklar och upp till sex certifikat samt möjlighet att lagra symmetriska nycklar.
TEID	Aktivt kort för öppna miljöer. Innehåller bl a två 2048 bitars privata RSA-nycklar och upp till sex certifikat
TSA	Totalförsvarets Signalskyddssamordning
TSA-lib	Ett bibliotek med TSA:s implementationer av kryptoalgoritmer
Ö	System för hantering av information som ej omfattas av sekretess enligt sekretess lagen (1980:100).

Denna studie följer Datatermgruppens rekommendationer(67) för svenska datatermer.

11. Litteraturtips

Följande böcker rekommenderas om man vill fördjupa sig i sådant som tas upp i denna studie:

11.1. Böcker

- Hacking Exposed Windows: Microsoft Windows Security Secrets and Solutions. 3rd Edition Scambray, Joel. McGraw-Hill Osborne Media, 4 dec 2007.
- Securing Windows Server 2008: Prevent Attacks from Outside and Inside Your Organization. Tiensivu, Aaron. Syngress, June 2, 2008
- Mastering Windows Network Forensics and Investigation, Steve Anson, Steve Bunting.
- Windows Internals: Including Windows Server 2008 and Windows Vista, Fifth Edition, Mark Russinovich, David A. Solomon, and Alex Ionescu. Paperback, Feb 4, 2009.
- Rootkits: Subverting the Windows Kernel, Greg Hoglund, Jamie Butler. Addison-Wesley Software Security Series, August 1, 2005.
- Incident Response & Computer Forensics, 2nd Edition, Kevin Mandia, Chris Prosise, and Matt Pepe. McGraw-Hill/Osborne, 2003.

11.2. Artiklar

- http://www.symantec.com/avcenter/reference/ATR-VistaAttackSurface.pdf

11.3. Websajter

- http://www.securityfocus.com/bid/
- http://www.winhackingexposed.com
- http://www.informit.com/articles/article.aspx?p=22661
- http://www.schneier.com
- http://www.symantec.com/avcenter/reference/Vista_Network_Attack_Surface_RTM.pdf

12. Referenser

1. L0phtCrack. [Online] [Citat: den 20 jan 2009.] http://en.wikipedia.org/wiki/L0phtCrack.
2. Projects. [Online] [Citat: den 23 apr 2009.] http://storm.net.nz/projects/16.
3. Windows NT - Wikipedia, the free encyclopedia. [Online] [Citat: den 15 jan 2009.] http://en.wikipedia.org/wiki/Windows_NT.
4. Windows Server 2008: Compare Technical Features and Specifications. [Online] [Citat: den 20 jan 2009.] http://www.microsoft.com/windowsserver2008/en/us/compare-specs.aspx.
5. Windows Server 2008: Compare Server Core Installation Options. [Online] [Citat: den 24 jan 2009.] http://www.microsoft.com/windowsserver2008/en/us/compare-core-installation.aspx.
6. Source lines of code. [Online] [Citat: den 24 apr 2009.] http://en.wikipedia.org/wiki/Source_lines_of_code.
7. Windows Server 2008 - Wikipedia, the free encyclopedia. [Online] [Citat: den 1 februari 2009.] http://en.wikipedia.org/wiki/Windows_Server_2008.
8. *H SÄK IT.* 2008-06-01.
9. Callback (computer science). [Online] [Citat: den 17 apr 2009.] http://en.wikipedia.org/wiki/Callback_(computer_science).
10. Defending Against Weak Authentication Protocols and Passwords. [Online] [Citat: den 9 apr

2009.] http://esj.com/articles/2004/11/10/defending-against-weak-authentication-protocols-and-passwords.aspx.
11. **Scambray, Joel.** *Hacking Exposed Windows: Microsoft Windows Security Secrets and Solutions.* 3rd Edition. u.o. : McGraw-Hill Osborne Media, 4 dec 2007. ISBN-10: 007149426X, ISBN-13: 978-0071494267.
12. **O'Dwyer, Frank.** Feasibility of Attacking Windows 2000. [Online] [Citat: den 6 feb 2009.] http://www.securiteam.com/windowsntfocus/5BP0H0A6KM.html.
13. Generic Security Service Application Program Interface. [Online] [Citat: den 12 feb 2009.] http://tools.ietf.org/html/rfc2743.
14. Insider Threat Exaggerated, Study Says . [Online] [Citat: den 19 jan 2009.] http://www.pcworld.com/businesscenter/article/147098/insider_threat_exaggerated_study_says_.html.
15. Nmap - Free Security Scanner For Network Exploration & Security Audits. [Online] [Citat: den 3 feb 2009.] http://nmap.org/.
16. Scanmetender Standard 3.1. [Online] [Citat: den 23 apr 2009.] http://www.softpedia.com/get/Network-Tools/Network-IP-Scanner/Scanmetender-Standard.shtml.
17. SuperScan v4.0. [Online] [Citat: den 23 apr 2009.] http://www.foundstone.com/us/resources/proddesc/superscan4.htm.
18. Link Layer Topology Discovery. [Online] http://en.wikipedia.org/wiki/LLTD.
19. Windows Rally. [Online] [Citat: den 8 mar 2009.] http://www.microsoft.com/whdc/connect/rally/default.mspx.
20. Link Layer Topology Discovery Protocol Specification. [Online] http://www.microsoft.com/whdc/connect/rally/lltd-spec.mspx.
21. **Hoagland.** Windows Vista Network Attack Surface Analysis. [Online] [Citat: den 8 mar 2009.] http://www.symantec.com/avcenter/reference/Vista_Network_Attack_Surface_RTM.pdf.
22. Enable or Disable the LLTD Responder by Using Group Policy. [Online] http://technet.microsoft.com/en-us/library/cc772308.aspx.
23. An Ethernet Address Resolution Protocol. [Online] [Citat: den 9 mar 2009.] http://tools.ietf.org/html/rfc826.
24. Neighbor Discovery for IP version 6 (IPv6). [Online] [Citat: den 9 mar 2009.] http://tools.ietf.org/html/rfc4861.
25. Next Generation TCP/IP Stack. [Online] [Citat: den 7 mar 2009.] http://technet.microsoft.com/en-us/network/bb545475.aspx.
26. IPv6 for Microsoft Windows: Frequently Asked Questions. [Online] [Citat: den 9 mar 2009.] http://www.microsoft.com/technet/network/ipv6/ipv6faq.mspx.
27. Teredo Overview . [Online] [Citat: den 10 mar 2009.] http://technet.microsoft.com/en-us/library/bb457011.aspx.
28. 6to4. [Online] [Citat: den 10 mar 2009.] http://en.wikipedia.org/wiki/6to4.
29. Source Routing. [Online] http://en.wikipedia.org/wiki/Source_routing.
30. Source Routing. [Online] [Citat: den 10 mar 2009.] http://www.iss.net/security_center/advice/Underground/Hacking/Methods/Technical/Source_Routing/default.htm.
31. s 7 2008 2003 Vista XP 2000 Ms Office Tutorials Fixes Tips. [Online] [Citat: den 10 mar 2009.] http://www.windowsreference.com/networking/disable-ip-source-routing-in-windows-server-2008-windows-vista/.
32. Server Message Bloc. [Online] [Citat: den 12 mar 2009.] http://en.wikipedia.org/wiki/Server_Message_Block.
33. Kernel Patch Protection - Wikipedia, the free encyclopedia. [Online] [Citat: den 19 01 2009.] http://en.wikipedia.org/wiki/Kernel_Patch_Protection.
34. Bypassing PatchGuard on Windows x64. [Online] [Citat: den 22 jan 2009.]

http://www.uninformed.org/?v=3&a=3&t=pdf.
35. Secure LogEye. [Online] [Citat: den 12 5 2009.] http://www.logeye.se.
36. NetIQ Security Manager. [Online] [Citat: den 30 mar 2009.]
http://www.netiq.com/products/sm/default.asp.
37. Snare BackLog. [Online] http://www.intersectalliance.com/projects/SnareBackLog/index.html.
38. ArcSight Logger - Log Management. [Online] [Citat: den 30 mar 2009.]
http://www.arcsight.com/products/products-logger/.
39. **Channel 9, Microsoft.** Iain McDonald and Andrew Mason show off the new Windows Server OS. [Online] den 24 May 2006. [Citat: den 01 feb 2008.]
http://channel9.msdn.com/posts/Duncanma/Iain-McDonald-and-Andrew-Mason-show-off-the-new-Windows-Server-OS/.
40. What is svchost.exe And Why Is It Running? [Online] [Citat: den 2 mar 2009.]
http://www.howtogeek.com/howto/windows-vista/what-is-svchostexe-and-why-is-it-running/.
41. Ping of death. [Online] [Citat: den 3 apr 2009.] http://en.wikipedia.org/wiki/Ping_of_death.
42. Ping flood. [Online] [Citat: den 7 mar 2009.] http://en.wikipedia.org/wiki/Ping_flood.
43. Set Up a Disconnected Network (Import and Export Updates). [Online] [Citat: den 12 feb 2009.]
http://technet.microsoft.com/en-us/library/cc720512.aspx.
44. Microsoft Baseline Security Analyzer. [Online] http://technet.microsoft.com/en-us/security/cc184924.aspx.
45. Patch Tuesday - Wikipedia, the free encyclopedia. [Online] [Citat: den 15 feb 2009.]
http://en.wikipedia.org/wiki/Patch_Tuesday.
46. How to apply predefined security templates in Windows Server 2003. [Online] [Citat: den 8 feb 2009.] http://support.microsoft.com/kb/816585.
47. Server Security Policy Management in Windows Server 2008. [Online] [Citat: den 17 jan 2009.]
http://technet.microsoft.com/en-us/library/cc754373.aspx.
48. Security configuration guidance support. [Online] [Citat: den 12 mar 2009.]
http://support.microsoft.com/kb/885409.
49. Microsoft Solution Accelerators. [Online] den 30 mar 2009. http://technet.microsoft.com/en-us/solutionaccelerators/default.aspx.
50. Windows Server 2008 Security Guide. [Online] [Citat: den 10 mar 2009.]
http://technet.microsoft.com/sv-se/library/cc264463(en-us).aspx.
51. Security Compliance Management Toolkit Series. [Online]
http://www.microsoft.com/downloads/details.aspx?FamilyID=5534bee1-3cad-4bf0-b92b-a8e545573a3e&DisplayLang=en.
52. Windows Benchmarks. [Online] [Citat: den 29 mar 2009.]
http://www.cisecurity.org/bench_windows.html.
53. CSRC System Administration. [Online] http://csrc.nist.gov/itsec/.
54. Guide to Securing Microsoft Windows XP Systems for IT Professionals: A NIST Security Configuration Checklist. [Online] http://csrc.nist.gov/itsec/SP800-68r1.pdf.
55. DISA Security Checklists. [Online] [Citat: den 30 mar 2009.] http://iase.disa.mil/stigs/checklist/.
56. Security Configuration Guides - Operating Systems. [Online] [Citat: den 30 mar 2009.]
http://www.nsa.gov/ia/guidance/security_configuration_guides/operating_systems.shtml#microsoft.
57. Samba. [Online] [Citat: den 12 apr 2009.] http://www.samba.org.
58. Trusted Solaris. [Online] [Citat: den 12 apr 2009.] http://en.wikipedia.org/wiki/Trusted_Solaris.
59. Common Criteria - The Common Criteria Portal. [Online] [Citat: den 12 03 2009.]
http://www.commoncriteriaportal.org/.
60. Samba-TNG. [Online] [Citat: den 27 feb 2009.] http://www.samba-tng.org/.
61. Layered Service Provider. [Online] [Citat: den 23 apr 2009.]
http://en.wikipedia.org/wiki/Layered_Service_Provider.
62. Interface description languag. [Online] [Citat: den 23 apr 2009.]

http://en.wikipedia.org/wiki/Interface_description_language.
63. Web Services Security. [Online] [Citat: den 23 apr 2009.]
http://www.ibm.com/developerworks/library/specification/ws-secure/.
64. stunnel - multiplatform SSL tunneling proxy. [Online] http://stunnel.mirt.net/.
65. The TLS Protocol. [Online] [Citat: den 17 apr 2009.] http://www.ietf.org/rfc/rfc2246.txt.
66. PKCS #11: Cryptographic Token Interface Standard. [Online] [Citat: den 17 apr 2009.]
http://www.rsa.com/rsalabs/node.asp?id=2133.
67. Svenska datatermgruppen. [Online] [Citat: den 15 jan 2009.] http://www.nada.kth.se/dataterm.
68. **Tiensivu, Aaron.** *Securing Windows Server 2008: Prevent Attacks from Outside and Inside Your Organization.* u.o. : Syngress, June 2, 2008. ISBN-10: 1597492809, ISBN-13: 978-1597492805.