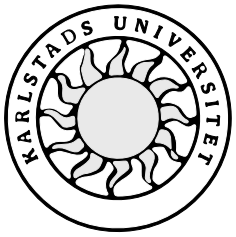



Datavetenskap

---

**Andreas Skoglund och Kristin Berg**

# Analys av loggar vid intrång i Windows NT

---

Examensarbete, C-nivå

2002:27

# Analys av loggar vid intrång i Windows NT

**Andreas Skoglund och Kristin Berg**

Denna rapport är skriven som en del av det arbete som krävs för att erhålla en kandidatexamen i datavetenskap. Allt material i denna rapport, vilket inte är vårt eget, har blivit tydligt identifierat och inget material är inkluderat som tidigare använts för erhållande av annan examen.

Andreas Skoglund

Kristin Berg

Godkänd, juni 05, 2002

Handledare: Stefan Lindskog

Examinator: Stefan Lindskog

# Sammanfattning

Detta examensarbete behandlar loggning i samband med intrång i Windows NT (NT). I början av projektet gjorde någon utomstående ett intrång på vår dator. Angriparna använde vår dator för att lagra olagliga spel. Detta är bara en av många anledningar till att göra ett intrång. Genom logginformationen som genererades under detta intrång kunde vi analysera och förstå vad angriparen hade gjort med vårt system. För att fortsätta detta arbete började vi reproducera egna intrång genom att använda speciella program. Dessa intrång genererade i sin tur loggar som vi kunde analysera. Den här analysen hjälpte oss att bestämma om loggarna i NT fungerar tillräckligt bra eller inte. Vidare har vi under detta arbete lärt oss att detektera ett intrång med hjälp av loggarna.

Det här examensarbetet svarar även på frågorna vilka angriparna kan tänkas vara, vad svagheterna i ett system är och hur loggning fungerar i NT. Dokumentet är baserat på våra egna erfarenheter.

# Analysis of audit logs from intrusions in Windows NT


## Abstract

This Bachelor's project considers auditing in relation to intrusions in Windows NT (NT). At the beginning of this project someone made an intrusion in our system. These intruders were using our computer to store illegal computer games. This is just one of several purposes to conduct an intrusion. Through the audit logs that were generated during this intrusion, we were able to analyze and understand what the intruder had done with our computer. To proceed this work we started to use special programs to reproduce known intrusions by ourselves. These intrusions generated logs that we could analyze. The analysis helped us decide whether the logs in NT are sufficient or not. From this work we also learned how to detect an intrusion from the logs.

This Bachelor's project also answers the questions: who the intruders might be, what the weaknesses in a system can be and how auditing works in NT. The document is based on our own experiences.

# Innehållsförteckning

# Figurförteckning

# Tabellförteckning

# 1 Inledning

Loggning i *Windows NT* (*NT*) är ett relativt outforskat ämne. Ett ämne som vi gärna vill ta reda på ifall det går att utnyttja mer och bättre.

Vår uppfattning var att loggningen i NT inte fungerar helt tillfredsställande. Åtminstone behövs en bättre förståelse för vad som är viktigt att logga, och vetskapen att man faktiskt kan upptäcka intrång utan att skaffa ytterligare program till datorn. Som vanliga användare av NT, innan vi började gräva i information om loggar, hade vi inte ens en aning om att loggning var inbyggt i systemet. Det finns ju faktiskt redan information att tillgå och använda vad det gäller att upptäcka intrång.

Det svåra med loggning är att det inte fungerar att logga allting. Det skulle medföra stora försämringar i prestanda och det skulle dessutom bli oerhört svårt att sålla information om vad som är viktigt och vad som är falska alarm eller åtminstone onödigt. Det skulle annars bli ett alltför tidsödande och hopplöst arbete att sitta och kolla igenom hundratals rader i loggfilerna. Loggfiler tenderar att växa snabbt och ta upp mycket plats. Därför behöver man bestämma vad som ska loggas och vad som inte behöver loggas. Helt enkelt göra en avvägning av vad som är nödvändigt att logga, och samtidigt se till att det inte tar upp plats i onödan. Målet är att hitta en bra balans som upptäcker flest antal intrång med minsta möjliga loggning.

Vi kommer i den här rapporten framförallt att utforska loggning för att kunna upptäcka intrångsförsök. Vårt mål är att genom loggningen kunna upptäcka intrång eller intrångsförsök. För att enklare se vad som kan utvinnas ur loggfiler kommer vi att reproducera några intrångsförsök att utgå ifrån. Genom att utföra intrången och sedan analysera vad som genereras till loggfilerna tror vi oss kunna få en bra bild av hur väl loggningen fungerar.

I kapitel 2 ges en sammanfattning av Stefan Lindskogs introduktion till Windows NT [10] samt en förklaring av Internet Information Server (IIS). En mer ingående beskrivning av säkerheten i NT görs i kapitel 3. Kapitel 4 behandlar hur loggningen i NT fungerar både övergripande och mer detaljerat. Motivet med intrång, svagheter i systemet samt olika typer av angripare beskrivs i Kapitel 5. I kapitel 6 redogörs för de lyckade intrångsförsök som vi själva har reproducerat. Där följer även en analys av vad som behöver loggas för att dessa intrång ska upptäckas. I kapitel 7 förs sedan en diskussion om hur loggningen skulle behöva fungera för att vara tillfredsställande, enligt vår åsikt. I kapitel 8 diskuterar vi hur ett fortsatt arbete skulle kunna se ut.

I bilaga A finns förklaringar till engelska ord och förkortningar som kommer att användas i rapporten. De nya ord och begrepp som introduceras i texten kommer att markeras kursivt för att visa att förklaring finns i bilagan.

Bilaga B innehåller en förklaring och berättande text till ett intrång som skedde på vår server under tiden som examensarbetet utfördes. Här berättar vi vad det var för intrång, vad som hände och vad de gjorde. För den intresserade beskriver vi även här masken Nimda[1] som infekterade vår server i samband med intrånget.

I bilaga C visas en IIS-logg från ett intrång där angriparna (i det här fallet vi) har använt IISHack, se kapitel 6.5.

## 1.1 Bakgrund

Bakgrunden till detta examensarbete var från början att analysera intrångsdata insamlat via två stycken projektarbeten som genomfördes i kursen Tillämpad Datasäkerhet, som gavs på Karlstads universitet åren 2000 och 2001. Dessa projekt gick ut på att försöka göra intrång på ett särskilt konfigurerat NT-system. Vår uppgift blev att försöka reproducera dessa intrång och komma fram till hur de kan upptäckas i loggfiler. Det vi hade till hjälp, för att försöka reproducera intrången, var aktivitetsrapporter och slutrapporter från kursdeltagarna samt loggfiler från maskinerna.

Tyvärr var kopplingen mellan rapporterna och loggfilerna dåliga och vi hade svårt att hitta loggarna till de gjorda intrången i loggfilerna. Istället valde vi att reproducera egna intrång som vi ansåg vara intressanta, se kapitel 6, och analysera de loggar som skapades av dessa intrång.

[1] Nimda är en mask som fungerar ungefär som ett datavirus fast det klarar att sprida sig självt

# 2 Översikt av Windows NT

NT är ett komplett 32-bitarsoperativsystem, utvecklat av Microsoft. Det har stöd för processer, multipla trådar, symmetrisk multiprocessing och distribuerad databehandling. NT är ett enanvändarsystem med stöd för flera samtidiga processer och är gjort för att kunna köras på en rad olika persondatorer. Systemet är till största delen skrivet i C, förutom att de delar som kommunicerar med hårdvaran är assemblerprogrammerade. NT använder en objektmodell för att hantera sina resurser.

## 2.1 Systemarkitektur

NT har både en skiktindelad- och client/server-systemarkitektur [2].

I ett skiktindelat system tillhandahåller varje modul en mängd funktioner som andra moduler, från en högre nivå, kan anropa. Dessa system blir lättare att modifiera och testa, och ett skikt kan bytas ut mot ett annat.

I client/server-modellen delas operativsystemet upp i ett antal processer, som kallas serverprocesser. Varje sådan process erbjuder en typ av tjänst (exempelvis minneshantering). Applikationsprogrammen exekverar som klientprocesser. En applikation begär en tjänst genom att skicka ett meddelande till en server. Alla meddelanden som skickas i systemet går via en så kallad mikrokärna, som exekverar i *kernel mode*. Övriga delar, både klient- och serverprocesser, exekverar i *user mode*. Med denna typ av modell kan en server fela utan att övriga delar av systemet påverkas.

I NT används client/server-modellen för att kunna tillhandahålla multipla OS-miljöer åt användaren. *NT Executive* är den delen av NT som exekverar i *kernel mode*. Denna del består av ett antal moduler som implementerar virtuell minneshantering, resurshantering, I/O- och filsystem, interprocesskommunikation, samt delar av säkerhetssystemet. Dessa delar kommunicerar med varandra genom ett antal väldefinierade (interna) funktioner.

Kärnan är en av NTs mest hårdvarunära delar och ansvarar för schemaläggning av trådar, avbrottshantering och synkronisering av processer. Den andra delen *Hardware Abstraction Layer* (*HAL*), ligger under kärnan och manipulerar hårdvaran direkt. I HAL-skiktet finns plattformsspecifik kod.

## 2.2 Processer

En process är en exekverande instans av en applikation. Varje process består av minst en tråd (eng. thread). NT fördelar CPU-tiden mellan systemets trådar. En process består av kod som har laddats från en exekverbar fil samt globala och statiska variabler. Processen äger även andra resurser som filer, trådar och dynamiska minnesareor som har skapats under dess livslängd. När en process avslutas återlämnas eller förstörs dessa resurser.

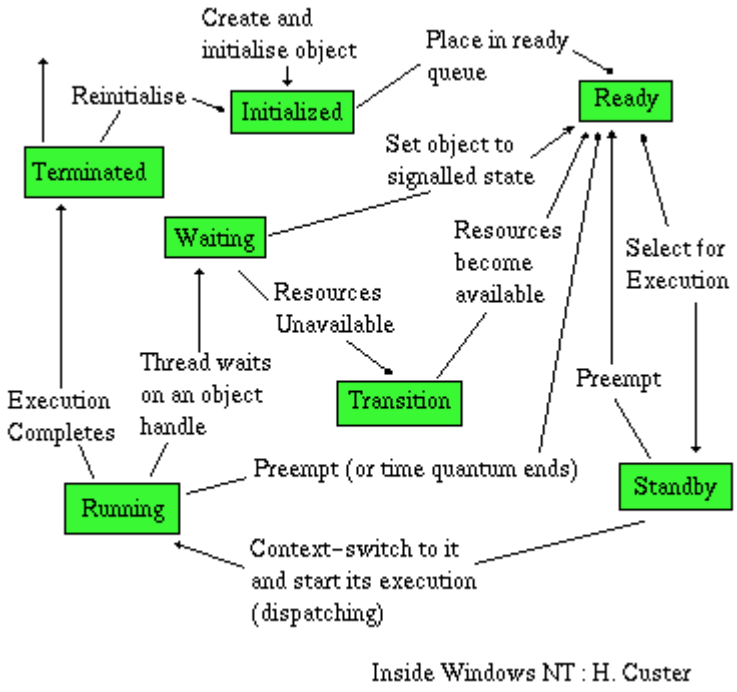
Det förekommer två olika typer av processer. *Graphic User Interface* (*GUI*)-baserade processer kan jämföras med en applikation som körs i Windows. En konsol-baserad process, å andra sidan, kan liknas vid en MS-DOS-applikation, där all in- och utmatning är teckenbaserad.

NT-processer implementeras som objekt. Till en process hör en lista av block som definierar det virtuella adressutrymmet som har tilldelats processen. Det är NTs virtuella minneshanterare som är minnesallokerare för processerna i systemet.

## 2.3 Trådar

NT schemalägger CPU-tid till alla aktiva trådar i systemet. Trådar representeras som objekt och består av ett antal attribut, och kapslar in ett antal tjänster.

En tråd kan befinna sig i ett av sex olika tillstånd: ready, standby, running, waiting, transition eller terminated. De trådar som befinner sig i ready-tillståndet kan bli schemalagda för exekvering. Det är den tråd som har högst prioritet som får exekvera först. Den utvalda tråden sätts till standby och får vänta på sin tur att exekvera i en processor. När tråden får tillgång till processorn ändras tillståndet till running som innebär att tråden exekverar instruktioner i processorn. Om tidskvantat för tråden tar slut eller om den blir avbruten av en högre prioriterad tråd, återgår den till ready. Behöver den däremot vänta på en händelse övergår den till tillståndet waiting. Terminate-tillståndet är slutstationen för en tråd. Om resurserna inte finns tillgängliga för en tråd som är redo att exekvera befinner sig tråden i transition-tillståndet. Figur 2.1 visar en översikt av vilka tillstånd trådarna kan befinna sig i samt övergångarna mellan de olika tillstånden.




*Figur 2.1: Trådarnas tillstånd och övergångar*

I NT används en preemptive CPU-schemaläggare med multipla prioritetsnivåer. Schemaläggningen inom varje prioritetsnivå sker med hjälp av en Round-Robin-algoritm. Preemptive innebär att en högre prioriterad tråd kommer att avbryta, och gå före, en lägre prioriterad. Prioriteten kan delas in i de två klasserna realtid och variabel. För realtidsklassen gäller att trådarna har en fast prioritet som aldrig ändras under deras livslängd. I variabelklassen tilldelas trådarna en initial prioritet som sedan kan höjas eller sänkas.

## 2.4 Minneshantering

Minnesarkitekturen i NT är baserad på ett virtuellt minnessystem, med en s k flat, linjär adressrymd som går att komma åt med 32-bitarsadresser. *Virtual Memory Manager* (*VMM*) ansvarar för att översätta logiska adresser till fysiska adresser, där faktiska data finns lagrade. VMM ser med hjälp av en minneskarta till att trådarna i systemet inte har möjlighet att komma åt andra processers eller operativsystemets minnesareor. När det fysiska minnet blir fullt är det VMMs uppgift att överföra delar av minnets innehåll till en disk. När data flyttas till en disk innebär det att sidor frigörs och kan användas på nytt. Operativsystemet finns på de höga adresserna och användararean på de låga adresserna.

## 2.5 Filsystem

NT har stöd för flera filsystem som *File Allocation Table* (*FAT*), *High Performance File System* (*HPFS*) och *New Technology File System* (*NTFS*) [1]. Varje typ av filsystem har egna filsystemsdrivrutiner som dynamiskt laddas vid behov. NTFS är ett helt nytt filsystem som är utvecklat för NT. Det har en snabb återställning av diskdata efter en systemkrasch, stöd för mycket stora filer, *POSIX*-stöd och *Unicode*-filnamn.

## 2.6 Internet Information Server

Internet Information Server 4 (IIS) ingår i paketet Option Pack och kan hämtas hem från Microsofts hemsida [20]. IIS är en webserver och är integrerad med NT server. Den gör det enkelt att publicera information och dela med sig av mjukvara till webben.

De två vanligaste tjänsterna som IIS tillhandahåller är HTML-server och FTP-server. Den tillhandahåller även tjänster för e-post, SMTP-server, och nyhetsgrupper, NNTP-server. Den tjänst som intresserar oss mest är HTML-servern eftersom den har många sårbarheter, t ex IIS Unicode buggen [12]. När vi har gjort våra intrång på testservern har några av de allvarligare, som vi anser, intrången utgått från just sårbarheter i IIS HTML-server.

# 3 Säkerhet i NT

När en användare loggar in i ett NT-system måste hon/han autentisera (eng. *authenticate*) sig genom att ange en användaridentitet och ett lösenord. Om inloggningen blir godkänd skapas en process för användaren samt en *accesstoken* som kopplas till processen. I accesstoken finns ett SäkerhetsID (*SID*), som identifierar användaren i systemet. När en ny process skapas, ärver den nya processen skaparens accesstoken. En accesstoken är viktig för att den håller all nödvändig användarinformation tillsammans för att ge snabb accessvalidering samt att den tillåter varje process att själv modifiera sin säkerhetskaraktäristik på ett begränsat sätt utan att påverka andra aktiva processer.

## 3.1 Identifiering

En användare identifieras i NT genom att användarnamnet är relaterat till ett internt SID. SID är ett numeriskt värde som är unikt inom en domän (eng. *domain*). När ett nytt konto skapas på ett system bildas ett tillhörande SID som lagras tillsammans med det nya kontot. SID-numren återanvänds aldrig och ett nyskapat konto kan alltså aldrig få ett gammalt SID-nummer.

## 3.2 Autentisering

I NT kan en användare autentisera sig med ett lösenord. Lösenordet lagras tillsammans med SID och övrig information om användaren i en databas som *Security Accounts Manager* (*SAM*) är ansvarig för. Databasen kan lagra två hashade versioner av lösenordet. När användaren skriver in sitt lösenord hashas det och jämförs med det hashade lösenordet i databasen. Om de två lösenorden stämmer överens är användaren inloggad. SAM-filen är alltid öppnad av systemet i ett s k exklusivt tillstånd och går därför inte att kommas åt när NT är igång.

## 3.3 Accesskontroll

Varje objekt i systemet är associerat med en *Access Control List* (*ACL*). Listan består av ett antal *Access Control Entries* (*ACE*). Varje ACE är associerad med en användare- eller grupp-SID och innehåller de funktioner som den här användaren är tillåten och inte tillåten att utföra

på det här objektet. En ACE som inte tillåter åtkomst sätts före de som tillåter åtkomst i ACL. En användare som inte har någon ACE i listan har ingen access alls till det objektet.

Ett objekt kan även ha en NULL ACL eller en tom ACL. En NULL ACL innebär att objektet inte har några restriktioner. En tom ACL, å andra sidan, betyder däremot att ingen användare kan komma åt det här objektet. Ett nyskapat objekt tilldelas oftast ACLen från sin skapare.

När en användare autentiseras i systemet skapas en accesstoken för denne. Denna token kallas primärtoken och innehåller bl a användarens SID och SIDs från de grupper som användaren är medlem i. Denna token jämförs med ett objekts ACL för att bevilja eller neka användaren access till det här objektet.

## 3.4 Loggning

I NT ansvarar *Security Reference Monitor* (*SRM*) och *Local Security Authority* (*LSA*) tillsammans med Event Logger för loggningen. Olika typer av händelser grupperas i kategorier och loggningen görs utifrån dessa grupper. Om loggning ska tillämpas och i så fall vad som ska loggas bestäms av *audit policy*. LSA ansvarar för audit policy och ger den till SRM. För mer utförlig information om loggning se kapitel 4.

## 3.5 Nätverksstöd

Att dela ut (eng. share) är en mekanism som tillåter en användare att utbyta information och resurser med andra användare i ett nätverk. Utdelning gör att kataloger (eng. directories) och skrivare kan kommas åt via ett nätverk. Olika sorters åtkomst kan ges beroende på om användarna behöver läsa, skriva eller ändra informationen inom det som delas ut. Man kan även begränsa användarna till en specifik person eller en hel grupp. Utdelning kan exempelvis göras på en fil, mapp eller en hel hårddisk.

En domän är ett nätverk av datorer som delar en SAM-databas och en säkerhetspolicy. En domänkontrollerare (eng. *domain controller*) är en NT-Server som kontrollerar och lagrar vissa data för en domän, inklusive domänkonton och globala grupper. NT stöder såväl lokala som globala grupper. Globala grupper existerar endast i domänkontrolleraren och inkluderar bara kontona på domänkontrolleraren. Globala grupper är synliga och användbara inom domänen och alla s k *trusted domains*. Lokala grupper är endast synliga och användbara på de egna arbetsstationerna. En arbetsstation är en ensamstående dator som kan vara kopplad till ett nätverk.

*Server Message Block* (*SMB*) är det protokoll som används i NT för en rad olika saker. Några av dessa är autentisering, *Remote Procedure Call* (*RPC*) och *Common Internet File System Protocol* (*CIFS*).

# 4 Loggning i NT

I NT finns stöd för loggning. Med loggning kan man upptäcka säkerhetsrelaterade händelser, exempelvis försök att skapa, ta bort samt komma åt systemresurser. Loggning kan användas till att registrera dessa händelser och därmed fungerar det som ett bevis för vad som har hänt i systemet.

I det här kapitlet förklarar vi hur loggningen fungerar för en vanlig användare samt hur det interagerar med systemet i övrigt, på en djupare nivå.

## 4.1 Hur loggning fungerar i NT

Olika moduler, som kallas källor (eng. sources) i NT, kan skicka information om händelser, som administratören väljer, till en säkerhetslogg (eng. *security log*). Händelser som man har valt att inte spara eller logga kommer att kastas. En händelse (eng. event), kan beskriva en högnivåhändelse, som exempelvis att någon loggar in i systemet eller en lågnivåhändelse, som exempelvis att ett program öppnar en fil. Vissa händelser relaterar till andra. Exempelvis relaterar utloggning till en sessions tidigare inloggning och en process som skapas kommer även att avslutas [17].

Vid varje inloggning får användarens processer en unik identifierare som kallas logon-id. Detta id kan användas till att paras ihop med korresponderande utloggning eller så kan särskilda händelser associeras till användaren genom detta id. Det finns två olika typer av inloggningar. Typ 2[2] är en interaktiv inloggning medan typ 3 är en fjärr- eller nätverksinloggning. Inloggningsprocess (eng. logon process) kallas den process som genererar inloggningsförfrågan. Interaktiva inloggningar görs av user32. Nätverksinloggningar genereras av KsecDD eller NTLanMan [17]. Figur 4.3 visar ett exempel på hur ett misslyckat inloggningsförsök kan se ut.

### 4.1.1 Loggar

Loggar (eng. *Audit logs*) är designade för två generella funktioner [14]:

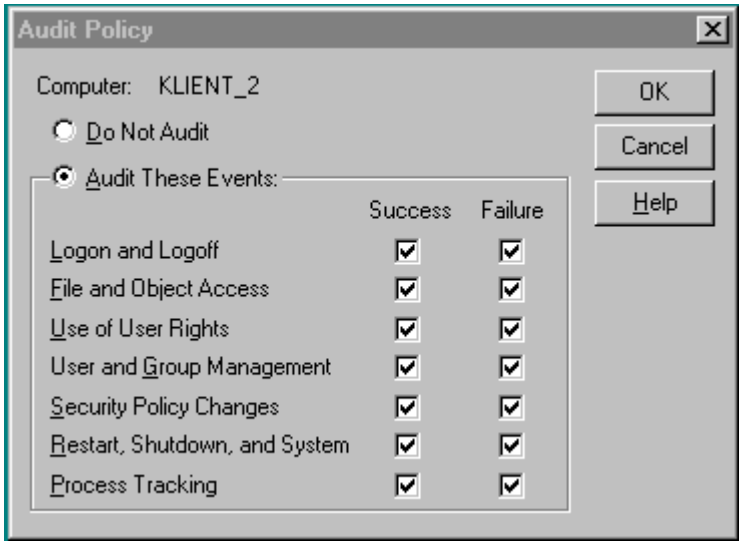
- Upptäcka potentiella systeminkräktare genom att göra dem ansvariga för sina aktioner.
- Fungera som ”efter faktumet” indikeringar av ett säkerhetsproblem.

[2] Typ 2 och typ 3 är de typer som beskrivs i [17], någon typ 1 har vi inte hittat information om.

### 4.1.2 Säkerhetslogg

Vidare finns det sju olika kategorier som man själv kan välja att logga eller inte logga till säkerhetsloggen [17]. Figur 4.1 visar hur fönstret där man kan välja bland de olika kategorierna ser ut. Här kan man även välja om man vill logga bara de lyckade händelserna, de misslyckade händelserna eller både och.

- **Logon and Logoff** – lagrar både interaktiva och nätbaserade in- och utloggningsförsök.
- **File and Object Access** – lagrar information om när en användare har kommit åt en fil, katalog eller skrivare som kontrolleras av ACL.
- **Use of User Rights** – lagrar information om när en användare försöker göra något som kräver rättigheter.
- **User and Group Managment** – lagrar information om skapande, ändring och borttagning av konton/grupper.
- **Security Policy Changes** – lagrar information om ändringar i loggning och rättighetspolicyn genom user manager's policies.
- **Restart, Shutdown and System** – lagrar information om att datorn har startats upp eller stängts ner och om en händelse som påverkar systemsäkerheten eller säkerhetsloggen har inträffat.
- **Process Tracking** – lagrar detaljerad information för programaktivering eller avslut.




*Figur 4.1: Figuren illustrerar vad man kan välja att logga i NT*

Säkerhetsloggning är bara av värde om två kriterier uppfylls enligt Rutstein, se [14]:

- Att systemet måste kunna identifiera användare exakt.
- Att loggarna behöver skyddas från inkräktare så att han/hon inte kan förstöra några bevis från intrånget.

### 4.1.3 Event Viewer

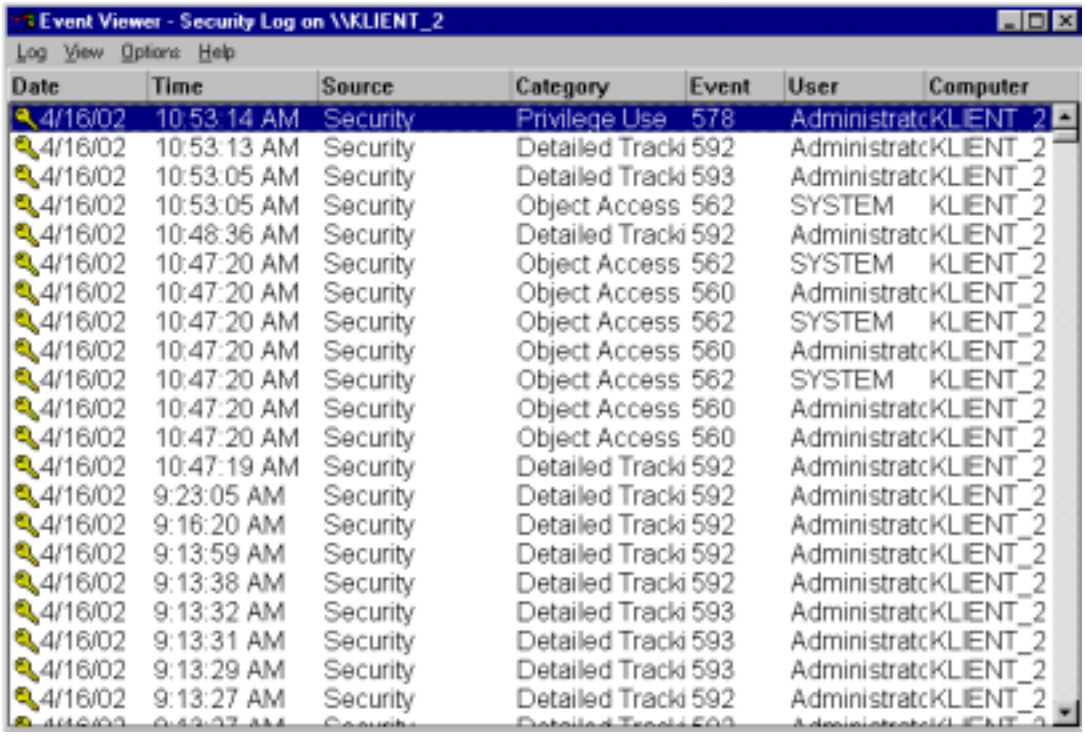
*Event Viewer* är det verktyg i NT som används för att gå igenom loggade händelser. Det finns tre olika sorters loggar:

- Systemloggen sparar fel, varningar eller information som genereras av NTs server system.
- Säkerhetsloggen kan spara giltiga och ogiltiga inloggningsförsök samt händelser som är relaterade till skapande, öppnande eller borttagning av filer och andra objekt.
- Applikationsloggen lagrar fel, varningar och information som genereras av applikations-mjukvara som elektronisk e-post eller databasprogram.

Systemloggen och Applikationsloggen kan läsas av vem som helst. Men endast systemadministratören eller användare med rättigheten ”Manage auditing and security log user right” kan kolla på säkerhetsloggen [8]. Tabell 4.1 visar de symboler, och deras förklaringar, som används i de tre loggarna. Figur 4.2 visar ett exempel på hur säkerhetsloggen kan se ut när den visas i Event Viewer.

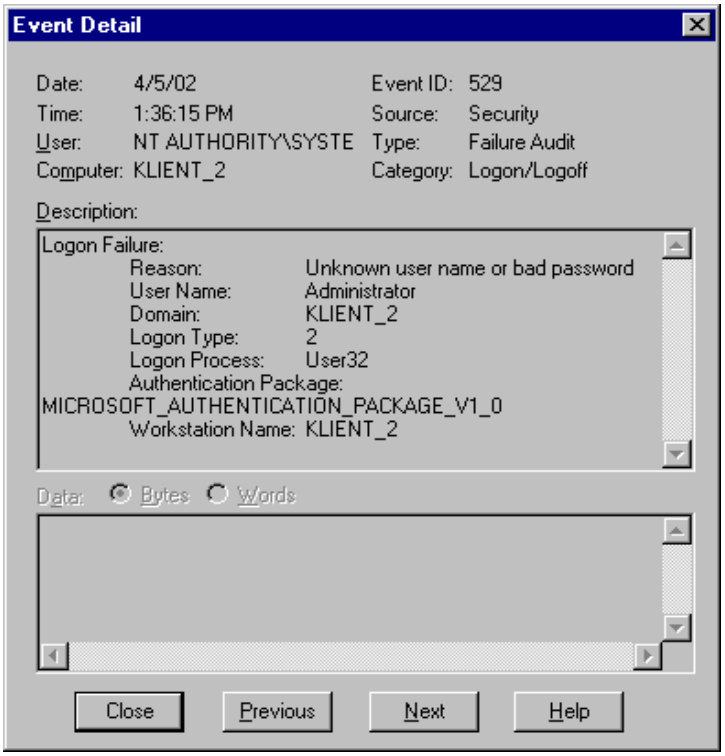
*Tabell 4.1: Symboler i Event Viewer*

| | |
|---|---|
|  | Låset indikerar ett fel eller förbud i säkerhetsloggen |
|  | Nyckeln betyder att det var en lyckad händelse i säkerhetsloggen |
|  | Informationsmeddelande i applikationsloggen och systemloggen |
|  | Varningsmeddelande i systemloggen |
|  | Påvisar ett fel i applikationsloggen |




*Figur 4.2: Security log som visas i Event Viewer*

Varje händelse i loggarna har ett unikt nummer som specificerar vad det är för slags händelse. En händelse (eng. event) #529 indikerar ett misslyckat inloggningsförsök, se Figur 4.3. Fler event ID och deras innebörd finns listade och beskrivna i [14] och [25].




*Figur 4.3: Detaljerad händelse i Event Viewer som visar ett misslyckat loginförsök*

### 4.1.4 Filer som behöver bevakas

Filer som är exekverbara och ligger under katalogen System Root behöver tilldelas vissa rättigheter förutom de som filerna får per default i systemet. Detta gäller alla filer som innehåller kod som kan exekveras av en användare (eng. *user*).

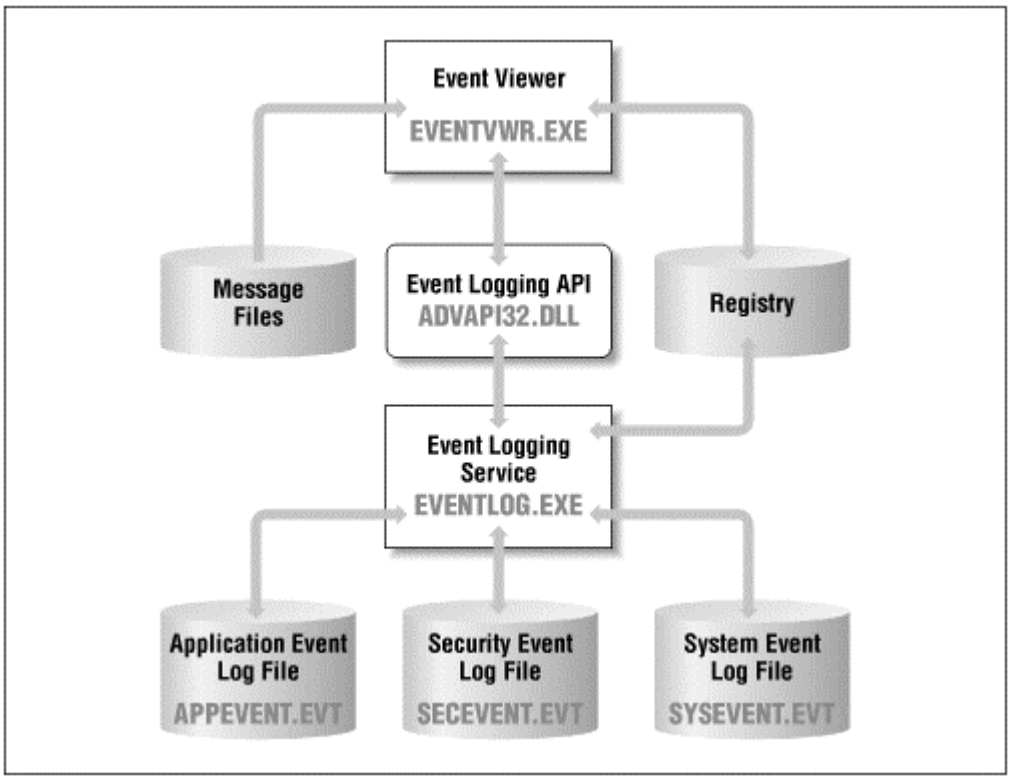
*Tabell 4.2: Filer som är extra känsliga för angrepp*

| Typ av fil | Förklaring |
|---|---|
| *.exe | Körbar fil |
| *.com | Körbar fil |
| *.bat | Scriptfil |
| *.ocx | Kontrollerar OLE |
| *.drv | Drivers |
| *.ini | Konfigurationsfiler |
| *.scr | Skärmsläckare |
| *.cmd | Kommandofiler |
| *.cpl | Kontrollpanelfiler |
| *.mod | Modulfiler |
| *.sys | Systemfiler |

Dessa filer kan innehålla exekverbar kod eller pekare till filer som innehåller exekverbar kod och kan därför bli ett mål för angripare [14].

## 4.2 Händelseloggning i NT– detaljerad beskrivning

Händelseloggning (eng. Event logging) är implementerad som en systemtjänst (eng. system-service) i NT. Win32 är en tjänst eller en process som kontrolleras av Service Control Manager (SCM). *Event logging service* ansvarar för att tillhandahålla all access till händelseloggarna i NT. Figur 4.4 visar en översikt av hur olika filer interagerar med varandra för att händelser ska kunna loggas och sedan läsas av ett program.




*Figur 4.4: Interagerande av ett Win32 program med event logging service*

När NT bootar startas SCM som i sin tur startar alla tjänster som är konfigurerade för en automatisk uppstart. En tjänst är event logging service. När en user-mode process behöver access till händelseloggarna måste den göra en förfrågan till event logging service för att utföra den önskade operationen. Förfrågan görs genom att anropa en eller flera av funktionerna som är definierade av event logging *API*.

Event logging API är en gateway till informationen som finns lagrad i *event logs* och den enda mekanism som finns tillgänglig för att rapportera händelser till NTs event logging service.

När operativsystemets drivrutin (eng. device driver), eller en applikation rapporterar en händelse, skickar den rapporten till event logging service. Där lagras sedan informationen som associeras med varje händelse som en post i en av de tre loggfilerna som finns på den lokala systemdisken.

*Registry* spelar också en stor roll vid loggningen. Alla händelsekällor måste registreras för att bli rätt igenkända. Event logging service använder varje källas registreringsinformation för att hitta de lokaliserade strängarna för varje händelsemeddelande. I Registry finns också alla *”event logging service”*-parametrarna lagrade.

Varje händelseloggfil lagrar information med EVT binary record format. En post är lagrad för varje händelse som skrivs till händelseloggen. Det är bara event logging service som läser

och skriver direkt till händelseloggfilerna [13]. I Figur 4.3 visas en loggpost på ett misslyckat inloggningsförsök.

### 4.2.1 Angrepp och hot mot loggarna

Om en inkräktare inte kan ändra eller rensa i loggarna, så är en möjlig utväg att förstöra loggarna med felaktig information. För applikations- och systemloggarna är detta enkelt att göra, eftersom alla processer har rättigheter att skriva till dessa loggar. Men till säkerhetsloggen kan bara händelser rapporteras av processer som körs i administratörsnivå eller det lokala systemkontot.

Det finns ingen garanti för att informationen som finns i en loggfil är riktig eller att den var rapporterad av den händelsekälla som är indikerad. Förstörelse av event log record data kan ske oavsiktligt p g a mjukvarufel, eller avsiktligt genom missledande händelser som pekar på andra händelsekällor. Inkräktare kan plantera en applikation på systemet som rapporterar händelser som är designade för att vilseleda systemadministratören från deras spår och in på ett annat problem. Det är också möjligt att ersätta en event source's event message file(s) med en trojan av den sort som innehåller felaktiga meddelanden.

Event logging facility på alla datorers operativsystem är oftast huvudmålet för inkräktare. Om datorns loggningsmekanism kan stängas av, så har inte systemadministratören något bevis på vad för illegal aktivitet som har försökts och kanske lyckats bli utfört på systemet. Därför är det väldigt viktigt att event logging service exekverbara filer, registry keys och själva händelseloggfilerna blir säkrade med rätt filsystems- och registry-rättigheter. På system som tillåter användaren att installera applikationer måste, oturligt nog, event log registry key vara skrivbar. Detta för att tillåta installationsprogram att registrera nya händelsekällor. Registry keys är därför känsliga för ett angrepp.

Det är också möjligt att ett avsiktligt angrepp kan vara menat direkt mot händelseloggfilerna. Ett vanligt angrepp är försöket att rensa eller ta bort säkerhetsloggen där viktiga bevis vanligtvis är sparade. Men en angripare skulle behöva logga in på systemet med ett konto på administratörsnivå för att lyckas [13].

## 4.3 IIS-loggning

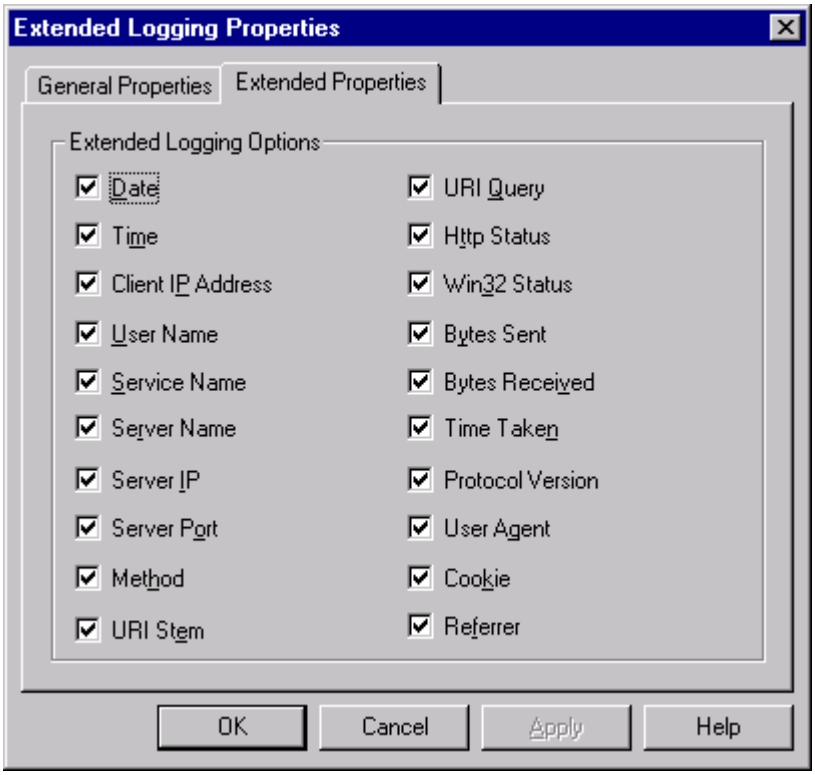
*IIS* står för Internet Information Server och erbjuder World Wide Web (WWW), File Transfer Protocol (FTP), Simple Mail Transfer Protocol (SMTP) och Network News Transfer Protocol (NNTP) tjänster till Windows NT Server 4.0.

I den NT-server som användes under försöken i Kapitel 6 var IIS installerad och gav därmed ökade möjligheter till att göra intrång. Med tanke på detta är IIS-loggen en viktig del i att upptäcka intrång i NT.

IIS kan administreras på två sätt, antingen via en Internet browser eller via Microsoft Management Console (MMC). När man administrerar IIS har man möjlighet att välja om loggning ska vara på eller av för varje enskild tjänst. Det finns fyra olika typer av loggningsformat, se Tabell 4.3. För W3C Extended Log File Format kan man välja vad som ska tas med i en loggningspost (eng. log entry) se Figur 4.5. De övriga tre har ett fixt format, vilket kan medföra att det som är viktigt inte kommer med i logg-filen.

*Tabell 4.3: Olika loggformat i IIS*

| **Loggningsformat** | **Förklaring** |
|---|---|
| W3C Extended Log File Format | Detta loggformatet är flexibelt och kan ställas in så att en loggningspost lagrar mycket information. |
| Microsoft IIS Log File Format | Har ett fixt format som inte lagrar speciellt mycket information. |
| National Center for Supercomputing Applications (NCSA) Common Log File Format | Har ett fixt format som matchar formatet som många andra Internetapplikationer lagrar, vilket kan vara en fördel i vissa lägen. |
| ODBC Log Format | Lagrar loggningsposterna i en databas, vilket också betyder att en databas måste var installerad på systemet. |




*Figur 4.5: Bilden visar vad man kan välja att logga i W3C Extended Log File Format*

# 5 Specificering av intrång

I detta kapitel försöker vi att förklara vad som kan betecknas som ett intrång. Vi tar även upp vad som är sårbart i ett system, vilka typer av angripare som finns samt vad syftet med ett intrång kan vara.

## 5.1 Vad är ett intrång?

Ett intrång (eng. intrusion) är när en person medvetet angriper ett system och försöker att påverka systemet på ett negativt sätt [11].

- Angriparen (eng. intruder) kan försöka att påverka systemets tillgänglighet (eng. availability) genom t ex en Denial of Service (DoS)-attack.
- Angriparen kan försöka att påverka integriteten genom att t ex ändra data i systemet.
- Angriparen kan försöka att påverka på systemets sekretess genom att t ex stjäla lösenordsfilen eller annan känslig information.
- Angriparen kan vidare försöka att påverka autentiseringen genom att t ex logga in med ett stulet lösenord.

## 5.2 Vem gör ett intrång?

I [16] skriver man att det finns tre typer av angripare.

- **Masquerader** är en person som inte är autentiserad att använda systemet, men penetrerar systemets accesskontroll för att utnyttja någon annans konto.
- **Misfeasor** är en legitim användare på systemet som accessar resurser i systemet som denna användare inte har rättighet till. Det kan också röra sig om att användare har autentiserad access till resursen men utnyttjar sina rättigheter på otillåtet sätt.
- **Clandestine user** är en person som tar övergripande kontroll över systemet och använder denna kontroll till att undkomma loggning och accesskontroll eller att ta bort möjligheten att samla in loggar från systemet.

Man kan dela in angripare ytterligare utifrån var de gör sitt angrepp. Två huvudtyper kan särskiljas.

- **Outsider** är en person som inte har ett konto tillgängligt på målsystemet. I denna grupp finns masquerader och vissa clandestine user.

- **Insider** är en person som har ett konto tillgängligt på målsystemet. I denna grupp finns misfeasor och vissa clandestine user.

Till sist har vi de två populära benämningarna hacker och cracker som finns definierade i [26].

- **Cracker** är en person som bryter sig in i system.
- **Hacker** är en person som är duktig på att programmera och kommer på finurliga lösningar till programmeringsproblem.

  Ordet hacker blir ofta felaktigt använt av media. Oftast när de skriver hacker syftar de egentligen på en cracker.

## 5.3 Motiv till intrång

Nedan beskrivs de olika motiv som vi tror är de främsta till att utföra intrång. Vi har, i de fall det är möjligt, tagit med konkreta exempel från de intrång vi gör i kapitel 6.

### 5.3.1 Störa tillgängligheten

Detta innebär att angriparen försöker få datorn otillgänglig, antingen lokalt, t ex genom CPUHog, se kapitel 6.1, eller utifrån, exempelvis med NewTear, se kapitel 6.7. När en angripare lyckas med en sådan attack kan det t ex ge möjligheten för angriparen att använda den angripna datorns identitet och få information som egentligen var tänkt till den otillgängliga datorn. Den här typen av angrepp kallas också *DoS-attacker*.

### 5.3.2 Lagra och sprida olaglig data

Vissa angripare vill alltid ha nya olagliga versioner av program, spel och filmer. Därför behöver dessa filer spridas på något sätt. Där kan ett angripet system komma in i bilden. Eftersom filerna är olagliga vill angriparen inte bli kopplad till dem, vilket kan inträffa om filerna finns lagrade på angriparens egna system. Istället angrips ett annat system för att lagra filerna där. Ett bra exempel på hur ett sådant intrång fungerar finns i bilaga B.

### 5.3.3 Stjäla information

Ofta kan det finnas känslig information lagrad på ett system. Denna information kan vara intressant för en angripare. Därför kan ett mål med intrång vara att få tag på den känsliga informationen. Företagsspionage kan utföras med hjälp av intrång.

### 5.3.4 Språngbräda till andra system

För att försvåra identifieringen av en angripare kan angriparen först attackera ett mindre säkert system och via detta system angripa andra system. Detta gör att i det primära målsystemet kommer det att synas språngbrädans identitet, istället för angriparens.

### 5.3.5 Ta kontroll över systemet

Om en angripare klarar av att ta kontroll över ett system kan personen ifråga t ex stoppa loggningen och ta bort tidigare loggning och därigenom dölja sitt intrång.

### 5.3.6 Ändra och förstöra information

Vissa angripare kan vara missnöjda med ett företag eller en organisation som de vill angripa. Detta ger angriparen en anledning till att göra ett intrång och börja radera eller ändra information. På vår server har det skett ett par gånger att startsidan på webservern har ändrats. Orsaken till detta vet vi inte, men vi tror att detta har gjorts antingen för nöjets skull, att höja angriparens status eller att skaffa sig mer erfarenhet.

### 5.3.7 Erfarenhet

Angriparen vill bara skaffa sig större erfarenhet av intrång eller intrångsprocessen, för att sedan kanske avancera till att göra svårare intrång.

### 5.3.8 Nöje

En "normal" användare av systemet är oftast nöjd med de rättigheter som givits. Men vissa personer tycker kanske det är roligt att göra intrång, och får en kick av att veta att de gör något otillåtet.

### 5.3.9 Status

Inom den s k cracker-kretsen anses det som en häftig grej att utföra ett intrång. Därför vill angripare sprida sina kunskaper om gjorda intrång och därmed höja sin status.

## 5.4 Sårbarheter i system

De orsaker som kan skapa sårbarhet (eng. vulnerability) i ett system kan delas in i sex kategorier enligt Dieter Gollman [4].

### 5.4.1 Förändring av system

En uppdatering av programvara är en förändring i systemet. Med en ny version av en programvara följer ofta nya funktionaliteter och alltför ofta också nya sårbarheter för systemet.

### 5.4.2 Kontroll av argument till kommandon och funktioner

När ett program eller en funktion startas skall syntaxen och argumenten kontrolleras, men detta sker inte alltid. Därför är detta en källa till sårbarhet i ett system. Det kan exempelvis uppenbara sig när ett argument är för långt och bufferten som lagrar argumentet får en s k *buffer overflow*. En sådan kan, i händerna på en skicklig angripare, ge möjligheten att exekvera otillåten kod.

### 5.4.3 Bekväma (eng. convenient) men farliga designegenskaper

Bakåtkompatibilitet, installationsguider, användarvänlighet är alla exempel på egenskaper som underlättar för användaren, men kan vara riktigt farliga ur säkerhetssynpunkt. De sårbarheter som yttrar sig i denna kategori är främst att program kan ge angripare möjlighet att komma in i system utan att behöva logga in eller på annat sätt autentisera sig.

### 5.4.4 Kontrollerade anrop

Det engelska uttrycket är "controlled invocation" och innebär att ett program som vill komma åt vissa resurser i systemet, där användaren egentligen inte får, kan göra det genom att tillfälligt byta mode, från user mode till kernel mode. Tack vare detta kan resursen kommas åt. Naturligtvis skall programmet, när det är klart med resursen, byta tillbaka, men om detta inte sker har användaren fått ökade rättigheter på systemet och ett intrång har skett. Därför skall dessa typer av program kontrolleras mycket noga så att de är fria från sådana svagheter.

### 5.4.5 Kringgå kontroll via en lägre nivå

I ett system finns oftast en accesskontroll som validerar accessen för användare och processer till resurser på systemet. Men om en skicklig angripare lyckas lägga in kod på en nivå under accesskontrollens nivå kan denna kontroll kringgås och angriparen får fritt fram att exekvera otillåten kod eller läsa känslig data. Det kan även vara så att en angripare lyckas få direkt access till minnet och även i detta fall ger det angriparen fritt fram att exekvera otillåten kod.

### 5.4.6 Brister i implementation av protokoll

Abstrakta beskrivningar av säkerhetsprotokoll är fulla av ofarliga påståenden t ex "välj ett slumpmässigt tal". Problemet dyker upp när protokollet skall implementeras. Då kan programmeraren välja en enkel lösning framför en säkrare och mer komplicerad. Detta kan medföra att t ex ett slumpat tal inte blir fullt så slumpmässigt som det egentligen var tänkt. Detta kan angripare utnyttja genom att göra kvalificerade gissningar på slumptalet och därigenom lyckas att t ex identifiera sig som någon annan. Det finns ett par bra exempel i [4].

# 6 Analys av intrångsförsök

Här redovisas ett antal kända intrång som vi själva har genomfört för att skapa loggar. Meningen är att dessa försök ska hjälpa oss att förstå hur man kan upptäcka ett intrång utifrån loggarna. Vi har utfört några olika typer av intrång. Intrång som stoppar tillgängligheten är exempelvis CpuHog, NewTear och NTCrash. Intrång som bryter mot sekretessen är PwDump och IISHack. IIS Unicode Exploit är ett intrång som bryter mot integriteten. Ett intrång som bryter mot autentiseringen är t ex GetAdmin. Idéer till flera av dessa har hämtats från [7] och andra har hämtats från Internet, bl a från packet storm [29].

De event ID som förekommer i resultaten kan hittas både i [14] och [25], där finns en djupare förklaring till alla event ID och en lista över de fält som finns med i händelsen (eng. event).

## 6.1 CpuHog

### 6.1.1 Beskrivning av angreppet

CpuHog är ett litet program som, när det körs, sätter sin prioritet till den högsta tillåtna. Detta gör att CpuHog kommer att ta alla CPU-resurser som finns tillgängliga och NT kommer att bli oerhört trögt. Detta är en s k *Denial-of-Service* (DoS)-attack som görs lokalt på en dator med NT [7].

### 6.1.2 Resultat i loggfilerna

CpuHog skapar endast en event #592 i loggen. Detta visar att en process, CpuHog.exe, har skapats. Väntar man på task manager, där alla aktiva processer visas i NT och där avslutar aktuell process, så kommer även en event #593, att processen har avslutats.

## 6.2 PwDump

### 6.2.1 Beskrivning av angreppet

PwDump är ett program som hämtar och skriver ut alla konton i ett NT-system. Programmet är ursprungligen tänkt att användas i samband med att konton ska synkroniseras med en *samba-server*. Förutom användarnamnet på kontot får man reda på kontots relativa id, de 32 sista

bitarna i SID, det hashade Lanman-lösenordet, NT-lösenordet hashat, kommentarer till kontot och hemkatalogen för användaren. Originalversionen, som följer med samba, har spärren att bara konton med administratörsrättigheter kan köra programmet. Detta är för att PwDump är inne i registrets SAM-delar och hämtar information. Om någon med illegala baktankar, som vill ha reda på lösenorden till kontona, klarar att köra programmet, kan denna person sedan knäcka lösenorden på en annan dator i lugn och ro.

Detta program kan användas till att få reda på lösenord till en NT-maskin [21].

### 6.2.2 Resultat i loggfilerna

PwDump skapar endast en event #592 (en process har skapats), och en event #593 (en process har avslutats) i NT-loggen.

## 6.3 NTCrash och NTCrash2

### 6.3.1 Beskrivning av angreppet

NTCrash är ett program som, när det körs, försöker att krascha NT. Detta gör programmet genom att anropa native-API-interface-funktioner med skräpparametrar. Om funktionen misslyckas med att validera parametrarna kommer systemet att krascha och visa en blåskärm (eng. blue screen). När skaparen av programmet, Mark Russinovich, körde programmet första gången hittade han 13 funktioner som misslyckades med att validera sina parametrar. Alla dessa resulterade i en systemkrasch. Mark påpekade detta till Microsoft som åtgärdade felen och tog med fixarna i Service Pack 1.

Det finns även en senare version av programmet som kallas NTCrash2[3]. Detta program hittade ytterligare 40 API-funktioner som resulterade i att NT kraschade. Dessa 40 API-funktioner rättade Microsoft till i Service Pack 4.

När vi provkörde NTCrash var Service Pack 6 installerat och resultatet blev inte att NT kraschade, istället fick programmet NT att bli oerhört trögt. [23]

### 6.3.2 Resultat i loggfilerna

NTCrash skapar endast en event #592 och en event #593, om man orkar med att vänta ut systemet, när programmet körs utan några flaggor satta. Man kan sätta en flagga, -n, när programmet startas och då kommer det förutom event #592 också kopiösa mängder med event #577 (privilegierad användning) att loggas.

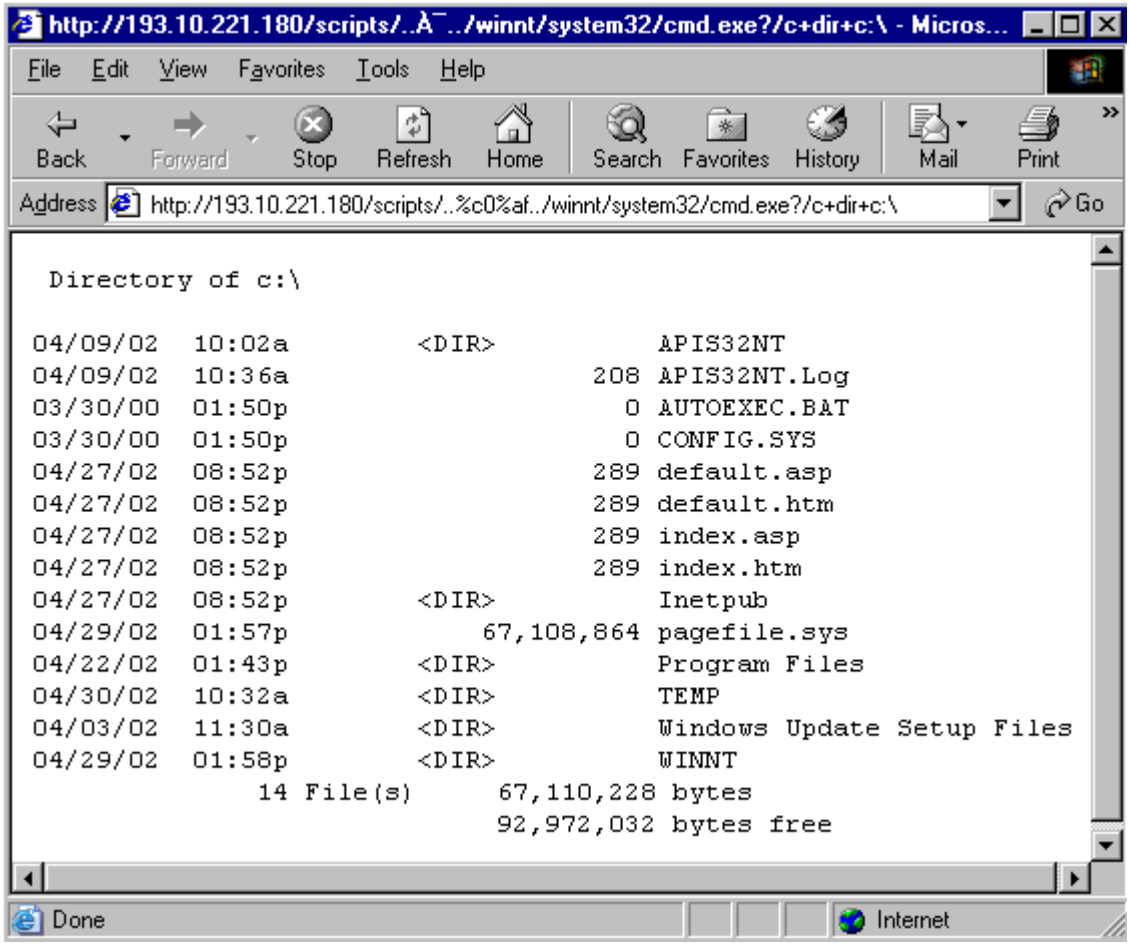
[3] Vi kunde dock inte hitta NTCrash2 och därmed heller inte provköra det.

## 6.4 IIS Unicode Exploit

### 6.4.1 Beskrivning av angreppet

IIS Unicode Exploit är en brist i IIS som gör att program som kan köras av användaren IUSR_maskinnamn, även kan köras av vem som helst från Internet. Detta är oerhört farligt eftersom det finns program som kan hämta hem andra program t ex *tftp*. Dessa program kan ta hem trojaner och bakdörrar (eng. backdoors) till system eller program som ökar rättigheterna åt användare och därmed ger inkräktaren administratörsrättigheter, vilket är förödande.

Orsaken till buggen är att när en Internetanvändare anger en felaktig (eng. malformed) URL till servern, kommer servern inte att avkoda URLens unicode kodning förrän efter kontrollen av URLens sökväg. Detta gör att när en Internetanvändare skriver in adressen
“http://<servernamn || ip>/scripts/..%c0%af../winnt/system32/cmd.exe?/c+dir+c:\”
i sin browser så kommer kommandot ”dir c:\” att utföras och resultatet visas för användaren i browsern, se Figur 6.1.




*Figur 6.1: Bilden illustrerar ett lyckat försök med att använda IIS Unicode Exploit*

Det är viktigt att sökvägen börjar i katalogen ”/scripts” eftersom IUSR_maskinnamn har exekveringsrättigheter i denna katalog. De konstiga tecknen som följer är tecknet ”/” kodat i unicode. När detta blir avkodat får adressen formen:
”http://<server namn || ip>/scripts/../../winnt/system32/cmd.exe?/c+dir+c:\”
som enklare visar vad som sker [12].

### 6.4.2 Resultat i loggfilerna

När vi utnyttjade denna brist såg vi i NT-loggen en event #528, där IUSR_maskinnamn loggar in. Sedan en event #576, användning av rättighet, därefter följde en event #592, cmd.exe startad av SYSTEM och en event #593, process avslutad, utförd av IUSR_maskinnamn. Dessutom fanns det i IIS-loggen ett logentry som matchar tiden av event #593. Beroende på hur man ställt in loggningen av IIS kan man få reda på en hel del från IIS-loggen, bland annat från vilken IP-adress kommandot har utförts.

## 6.5 IISHack

### 6.5.1 Beskrivning av angreppet

IISHack1.5 är ett program som dels utnyttjar IIS Unicode exploit, se kapitel 6.4, och dels utnyttjar en buffer overflow som inträffar i inetinfo.exe[4] när detta program går igenom en "elakartad" asp-fil.

IISHAck1.5 är ett demonstrationsprogram från organisationen Eeye. Det börjar med att kontrollera en IISserver efter en användbar sökväg för exekvering. Om programmet finner en sådan sökväg, kopieras cmd.exe dit. Därefter skapas en asp-fil i samma sökväg med hjälp IIS Unicode exploit som har echo-kommandot som argument. Till sist hämtar programmet asp-filen och om detta lyckas ska man få tillgång till en kommandoprompt, till den attackerade datorn, via telnet.

När vi testade programmet fick vi mycket riktigt tillgång till en kommandoprompt. Men om vi stängde denna telnet-uppkoppling, gick det inte att koppla upp igen utan att starta om servern [19]. Vi hade därmed alltså åstadkommit en DoS-attack.

### 6.5.2 Resultat i loggfilerna

Efter försöket undersökte vi NT-loggen. Där såg vi att det börjar med en event #528, IUSR_maskinnamn loggar in, som direkt följs av en event #576, användning av rättighet.

[4] Inetinfo.exe är det program som tar hand om alla begäran (eng. requests) till web-servern.

Därefter följer en event #592, cmd.exe startas, och efter det en event #593, cmd.exe avslutas. Detta är IIS Unicode exploiten som letar efter en användbar sökväg. Sedan ser vi en event #592, cmd.exe startas igen, och direkt efter en event #593, cmd.exe avslutas. Denna process kopierar cmd.exe till den funna sökvägen, IISHack1.5 byter dessutom namn på cmd.exe till eeyehack.exe, efter organisationen som skapat programmet.

Nästa händelse i NT-loggen är en event #592, eeyehack.exe startas, efter den följer en event #593, eeyehack.exe avslutas. Här skapas asp-filen som döps till eeyerulez.asp och hamnar i samma sökväg som eeyehack.exe. Nästa händelse i NT-loggen är en event #592, mdm.exe startar. Mdm.exe är Machine Debug Manager som, vi antar, startar när något gått fel i NT. Därefter följer en event #592, cmd.exe startar. Det är denna kommandoprompt som används under telnet-sessionen. Dessa två sista programstarter har inga event #593, som är brukligt vid normal användning eftersom de inte har fått något avslutningskommando. Detta gör att dessa program inte avslutas förrän systemet startas om eller när t ex administratören av systemet väljer att stoppa processerna i task manager.

I IIS-loggen ser man tre loggposter en när sökningen efter sökväg sker, en när cmd.exe kopieras och en när eeyerulez.asp-filen skapas, se bilaga C.

## 6.6 GetAdmin

### 6.6.1 Beskrivning av angreppet

GetAdmin [24] är ett program som ger en användare administratörsrättigheter genom att använda sig av en svaghet i en LOW-LEVEL kernel rutin. Denna rutin sätter en global flagga, som tillåter anrop till NtOpenProcessToken att lyckas oavsett vilka rättigheter som användaren har. Detta i sin tur tillåter användaren att binda sin egen process till vilken process som helst som körs på systemet, inklusive de som körs i systemets säkerhetsmiljö, t ex WinLogon. När man har bundit en sådan process, kan man skapa en ny tråd i samma säkerhetsmiljö som processen. Detta utnyttjar GetAdmin mot processen WinLogon och väl gjort är det möjligt att göra vanliga API-anrop som lägger till användaren i administratörsgruppen. Attacken måste göras lokalt på NT.

### 6.6.2 Resultat i loggfilerna

I NT-loggen syns att programmet GetAdmin.exe startas med event #592. Sedan följer en event #560, Object Open, efter den kommer en event #636, Local Group Member Added, nästa är en event #562, Handle Closed, och till sist är en event #593, GetAdmin.exe avslutat.

## 6.7 Newtear

### 6.7.1 Beskrivning av angreppet

Newtear är en DoS-attack som kan utföras från vilken dator som helst i Internet, angriparen behöver bara känna till IP-adressen till mål-datorn. Angreppet bygger på en svaghet i TCP/IP-stacken på NT. Ett par felaktiga (eng. malformed) IP-fragment som återförs till ett UDP paket orsakar att NT blir instabilt och kan krascha, antingen med en blå skärm eller helt enkelt bara boota om.

När vi gjorde försöket med Newtear blev effekten att en blå skärm dök upp och sedan en ombootning. När NT startade igen var det ovanligt segt och NT klagade på för lite virtuellt minne [18].

### 6.7.2 Resultat i loggfilerna

I NT-säkerhetsloggen finns inga som helst loggar som kan indikera att en Newtear-attack har inträffat, om man bortser från att man ser att NT har startats om. I systemloggen ser man en logg som säger att systemet är ”Out of virtual memory” men detta kan ju bero på andra orsaker än att just Newtear har körts.

## 6.8 Diskussion

När vi analyserar intrången, utifrån vad som loggas, kan man dra slutsatsen att NTs säkerhetsloggning inte är till mycket hjälp för att upptäcka vare sig DoS-attacker, t ex CpuHog, NtCrash och Newtear, eller fjärrattacker, som t ex IIS Unicode Exploit och IISHack. Visserligen gav IISHack något mer loggar men dessa styrkte i princip bara informationen man fick i IIS-loggen. Därför anser vi att NT-loggarna inte är någonting att ha vid sådana typer av angrepp.

Däremot när det gäller angrepp som går ut på att få administratörsrättigheter, t ex GetAdmin, får man lite mer intressanta loggar i NTs säkerhetslogg. Man ser att någon process har öppnat ett objekt av typen SAM_ALIAS med namnet ”DOMAINS\Builtin\0000220” och med accessen Add member samt får ett handtag (eng. handle) till objektet, en event #560, d v s någon process vill lägga till en användare i en inbyggd grupp. Detta bekräftas i nästa händelse som är en event #636. Denna händelse är att en användare blir tilllagd i en grupp. Mer exakt är att användaren läggs till i administratörsgruppen. Till sist släpps handtaget och objektet stängs, event #562. GetAdmin är ett lokalt angrepp därför kommer aldrig IIS-loggen ens med i bilden under attacken.

Eftersom GetAdmin är ett lokalt angrepp vilket betyder att en angripare måste ha fysisk kontakt med systemet och ett konto på datorn. Detta medför att angreppet är endast användbart av insider-angripare, d v s anställda på företaget eller liknande. Detta problem med illojala anställda kan möjligtvis lösas genom god personalvård.

Angrepp som får tag på lösenorden till användare på datorn, t ex PwDump, är också oftast lokala och kräver fysisk kontakt med datorn. Vissa går att utföra utan tillgång till ett konto på datorn, medan andra behöver ett konto. PwDump kräver i originalversionen t o m att det körs av en användare med ett administratörskonto. Därför är dessa attacker också bäst lämpade för insider-angripare. I avseende på NTs säkerhetsloggning ger PwDump inga intressanta loggar och är därmed svårt att upptäcka.

IIS-loggen är mer intressant i de fall där denna loggning kan komma att vara inblandad, vilket den gör när intrånget utnyttjar en sårbarhet i någon av IIS tjänster. Detta skedde i intrången IIS Unicode Exploit och IISHack. När dessa två intrång gjordes fick vi loggar i IIS-loggen som klart och tydligt visade vad som hade gjorts på systemet. T ex att en fil har kopierats och bytt namn eller att en fil skapats. Exempel på dessa loggar finns i bilaga C.

Därför anser vi att IIS-loggen är ett viktigt redskap när man ska försöka upptäcka intrång i IIS. På vårt system var det visserligen förhållandevis få poster i IIS-loggen som gjorde det lätt att se vad som loggas vid ett intrång. Ett system som har en ”verklig” funktion ute på Internet kan ha många besökare och får därmed många loggar som inte tyder på något intrång. Dessa loggar kan kanske tas bort på något sätt, t ex att man definierar hur tillåtna anrop till systemet ser ut och låta bli att logga dessa. På detta sätt får man bara de loggar som indikerar att ett intrång kan ha skett.

# 7 Resultat och erfarenheter

Vi tror att om man ska upptäcka så mycket som möjligt av intrången ska man logga all trafik som förekommer på öppna portar samt definiera vad som är tillåten form på trafiken. Då kan alerts genereras när trafik som inte är tillåten kommer in. Detta kan liknas vid ett Network Intrusion Detection System (*NIDS*). Men ett NIDS kontrollerar trafik på paketnivå, vi vill kontrollera trafiken på applikationsnivå d v s när applikationen får meddelanden. På denna nivå har man möjligheten att se exakt vad som vill göras med applikationen och därmed kan man kontrollera mer exakt vad som försiggår. IIS-loggen är ett bra exempel på loggning på applikationsnivån. Den anser vi fungerar bra och underlättar för att finna intrång. Den ger oss även information om vart intrång kom ifrån, tack vare att IP-adressen loggas. Detta faktum kan underlätta identifieringen av angriparen, om det skulle vara aktuellt.

NT-loggarna är inte bra, som vi påpekat tidigare, dels eftersom de inte är kompletta, t ex skulle vi vilja ha med argumentet när ett program startar, och dels skapas mycket loggnings-poster. Åtminstone skulle man kunna göra förbättringar för att få fram viktig information från NT-loggfilerna automatiskt. Genom att automatiskt gå igenom filerna och generera alerts skulle den säkerhetsansvarige slippa att kolla igenom rad efter rad i loggfilerna. Ett arbete som känns både tidsödande, gammalmodigt och närmare hopplöst eftersom många falska alarm och onödig information sparas. Det optimala skulle vara att hitta ett sätt att direkt förhindra ett intrångsförsök genom att exempelvis låsa kontot så fort en viss händelse som kan antas vara relaterad till ett intrångsförsök har inträffat.

# 8 Slutsatser och fortsatt arbete

Vid intrångsförsöken och intrånget beskrivet i bilaga B har vi studerat NT-loggarna och kommit fram till att dessa loggar inte är tillräckliga för att upptäcka intrång på ett effektivt sätt. T ex har vi funderat över varför inte argumentet till ett program finns med när detta startar. Denna information anser vi vara en viktig del när man ska upptäcka intrång. Detta finns med i IIS-loggen och har varit till stor hjälp i att förstå hur intrången har gått till, när IIS varit inblandad.

Vi har funderat på vad som egentligen är möjligt att logga i NT. Vi har funnit en del tredjeparts-program, Filemon, Pmon och Tokenmon från Sysinternals [28], som verkar lovande. Filemon är ett program som övervakar alla aktiviteter i NTs filsystem. Pmon är ett program som övervakar all aktivitet som sker med processer och trådar. Tokenmon är ett program som övervakar säkerhetsrelaterade aktiviteter. Tokenmon har fått sitt namn efter ett objekt som kallas token i NT. Token objektet lagrar säkerhetsrelaterad information om en process. Tokenmon genererar loggar när användare loggar på/av, när en process startar/avslutar, när någon process använder en annan identitet än den inloggade och när applikationer förändrar säkerhetsprivilegier i sina processer.

Vi gjorde ett par intrång när dessa program kördes men de, framförallt Filemon, genererade de stora mängder loggar och vi hade inte tid att söka genom dessa för att få en god uppfattning ifall de var bra. En intressant observation var dock att Filemon upptäckte en buffer overflow i inetinfo.exe när vi gjorde IISHack-intrånget.

En tänkt fortsättning på detta examensarbete är att använda dessa program, eller något annat program, när ett intrång sker för att se vad som loggas och göra en analys av loggarna för att eventuellt upptäcka om de har den egenskapen som gör att ett intrång lättare kan upptäckas.

# Referenser


[1] H. Custer. *A grand tour of windows NT: Portable 32-bit multiprocessing comes to windows*. Microsoft Systems Journal, 7(4):17-31, Jul-Aug 1992.

[2] H. Custer. *Inside Windows NT™*. Microsoft Press, 1993.

[3] J. Enck. *Advanced Technical Reference Windows NT® Server 4.0*. Que® Corporation, 1997.

[4] D. Gollman. *Computer Security*. Wiley, 1999.

[5] L. Hadfield, D. Hatter, D. Bixler. *Windows NT® Server 4 - Security Handbook*. Que® Corporation, 1997.

[6] H. Hedbom, S. Lindskog, S.Axelsson, E. Jonsson. *A Comparison of the Security of Windows NT and UNIX*. Chalmers, 1999.

[7] H. Hedbom, S. Lindskog, S.Axelsson, E. Jonsson. *Analysis of the Security of Windows NT*. Chalmers, 1999.

[8] J. G. Jumes, N. F. Cooper, P. Chamoun, T. M. Feinman. *Microsoft® Windows NT® 4.0 Security, Audit and Control*. Microsoft Press, 1999.

[9] N. Lambert, M. Patel. *PCWEEK Microsoft® Windows NT™ Security – System Administrator's Guide*. Macmillan Computer Publishing USA, 1997.

[10] S. Lindskog. *Introduktion till Windows NT*. Karlstads universitet, 2002.

[11] S. Lindskog. *Observations on Operating System Security Vulnerabilities.* Chalmers, 2000.

[12] N. Miller. *Microsoft IIS Unicode Exploit*. Lucent Technologies Worldwide Services, 2001.

[13] J. D. Murray. *Windows NT Event Logging*. O'Reilly™ Online Catalog, 1998.

[14] C. B. Rutstein. *guide to Windows NT Security*. McGraw-Hill Companies Inc, 1997.

[15] T. Sheldon. *Windows NT Security Handbook – Everything you need to know to protect your network*. McGraw-Hill Inc, 1997.

[16] W. Stallings. *Network Security Essentials.* Prentice Hall, 2000.

[17] S. A. Sutton. *Windows NT™ Security – guide*. Trusted System Services Inc, 1997.

**URLer:**


[18] http://www.cert.org/summaries/cs-98.02.html

[19] http://www.eeye.com/html/Research/Advisories/AD20001003.html

[20] http://www.microsoft.com/ntserver/nts/downloads/recommended/NT4OptPk/default.asp

[21] http://www.samba.org/samba/ftp/pwdump/README

[22] http://www.symantec.com/avcenter/venc/data/pf/w32.nimda.a@mm.html

[23] http://www.sysinternals.com/ntw2k/info/ntdll.shtml

[24] http://support.microsoft.com/default.aspx?scid=kb;EN-US;q146965

[25] http://support.microsoft.com/default.aspx?scid=kb;EN-US;q174074

[26] http://www.tuxedo.org/~esr/jargon/jargon.html

[27] http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/MS01-020.asp

[28] http://www.sysinternals.com/ntw2k/utilities.shtml

[29] http://packetstormsecurity.nl/

# A Terminologi

*Access Control Entry* – inträde i ACL som innehåller ett SID och en mängd access-rättigheter. En process med matchande SID är antingen tillåten de listade access-rättigheterna, förbjuden dem eller tillåts dem med loggning [2].

*Access Control List* – fil och skrivarserver kontrollerar resursens ACL innan den kan tillåta en användare att komma åt en fil eller använda en skrivare. Om användaren inte finns listad i ACLen får resursen inte användas [8].

*Accesstoken* – ett objekt som unikt identifierar en användare som har loggat in. En accesstoken är bunden till användarens alla processer och innehåller användaren, SID, namnet på de grupper användaren tillhör, de privilegier som användaren har, default ägaren av objekt som användarens processer skapar och default ACL som ska läggas till de objekt som användarens processer skapar [2].

*ACE* – se Access Control Entry.

*ACL* – se Access Control List.

*API* – se Application Programming Interface.

*Application Programming Interface* – en mängd rutiner som ett applikationsprogram använder för att efterfråga och utföra lower-level services som utförs av ett operativsystem [2].

*Auditing* – möjligheten att detektera och spara säkerhetsrelaterade händelser, särskilt försök att skapa, komma åt eller ta bort objekt [2].

*Audit Policy* – avgör de typer av säkerhetshändelser som ska loggas [9].

*Authenticate* – validering av användarens login-information [2].

*Buffer overflow* – inträffar när en buffer i ett program inte klarar av att lagra all den information som tilldelas den. Ofta innebär detta att en del i minnet skrivs över och om fel minne skrivs över med rätt kod kan denna kod exekveras.

*CIFS* – se Common Internet File System Protocol.

*Common Internet File System Protocol* – förbättrad version av SMB-service som finns tillgänglig för användning över Internet [8].

*Denial of service attack* – en av de vanligaste angreppen mot NT och andra operativsystem, förhindrar användande av systemet genom att använda alla tillgängliga resurser eller genom att stänga ner systemet [8].

*DoS attack* – se Denial of Service atack.

*Domain* – en grupp av datorer som innehåller domän kontroller som delar konto- information och har en centraliserad konto- databas [8].

*Domain controller* – en server i en nätverksdomän som accepterar användarenas login och initierar deras autentisering [2].

*Event log* – Default service som sparar system-, applikations- och säkerhetshändelser i event log filer [8].

*Event Logging Service* – tjänsten i NT som har hand om loggning.

*Event Viewer* – det verktyg i NT som används för att gå igenom loggade händelser [8].

*FAT* – se File Allocation Table.

*File Allocation Table* – filsystem som traditionellt används i MS-DOS [2].

*Graphic User Interface* – det grafiska gränssnitt som syns gentemot användaren.

*GUI* – se Graphic User Interface.

*HAL* – se Hardware Abstraction Layer.

*Hardware Abstraction Layer* – ett dynamiskt länkbibliotek som skyddar NT executive från variationer i olika försäljares hårdvaru-plattformar med uppgiften att maximera operativsystemets portability [2].

*High Performance File System* – filsystem som designades för OS/2 och utvecklades för att ta hand om begränsningarna i FAT [2].

*HPFS* – se High Performance File System.

IIS – se Internet Information Server.

*Internet Information Server* – se kapitel 2.6 och 4.3.

*Kernel mode* – det privilegierade processor mode som NTs systemkod körs i [2].

*Local Security Authority* – skapar en accesstoken för varje användare som har kommit in i systemet via inloggning [9].

*LSA* – se Local Security Authority.

*NIDS* – nätverksbaserad intrångsdetekteringssystem som kontrollerar trafiken på nätverket.

*NT* – se Windows NT.

*NT executive* – den del av NTs operativsystem som körs i kernel mode [2].

*New Technology File System* – avancerat filsystem speciellt designat för NT [2].

*NTFS* – se New Technology File System.

*Portable Operating System based on unIX* (POSIX) – en IEEE-kommitté som upprätthåller en mängd operativsystemstandarder som drivs av UNIX-industrin [17].

*POSIX* – se Portable Operating System based on unIX.

*Process* – kan utgöra ett virtuellt adressområde, ett exekverbart program, en eller flera trådar som exekverar, lite av användarens resurskvoter och systemets resurser som operativsystemet har allokerat till processens trådar [2].

*Registry* – databas i NT som kontrollerar datorn, innehåller alla system- och programkonfigurationsparametrar [8].

*Remote Procedure Call* – anrop av funktioner från en annan dator.

*RPC* – se Remote Procedure Call.

*SAM* – se Security Account Manager.

*Samba-server* – en server till Linux miljön, som möjliggör kommunikation mellan Linux och NT.

*Security Account Manager* – databas av säkerhetsinformation som inkluderar användarkontonamn och lösenord [2].

*Security Log* – en rapport där misstänkt information kan filtreras och spåras [8].

*Server Message Block* – ett nätverksprotokoll som definierar ett specifikt format för datapaket som ska överföras i nätverket [2].

*Security identifier* – används för att unikt identifiera varje användare, NT arbetsstation och server på nätverket.

*Security Reference Monitor* – en komponent i NT Executive som jämför en accesstoken för en process med ACL för ett objekt för att avgöra om processens trådar har tillåtelse att öppna ett handtag till objektet [2].

*SID* – se Security identifier.

*SMB* – se Server Message Block.

*SRM* – se Security Reference Monitor.

*Tftp* – se Trivial File Transfer Protocol.

*Trivial File Transfer Protocol* – verktyg som tillåter icke-autentiserad filöverföring till någon tftp-server [8].

*Trusted domain* – förtroendeförhållande mellan två domäner. De användare som tillhör en domän (trusted) har tillåtelse att använda resurser i den andra domänen (trusting) [2].

*Unicode* – en standard som har16 bitars kodning av symboler vilket gör att hela världens symboler kan representeras[2].

*User* – en användare i systemet.

*User mode* – platsen för det mesta av NT koden, sägs vara en icke-privilegierad processor mode [8].

*Virtual Memory Manager* – NT executive komponent som implementerar virtuellt minne [2].

*VMM* – se Virtual Memory Manager.

*Windows NT* – operativsystem från Microsoft.

# B Intrånget

Torsdagen den 14 februari 2002 började någon ett intrång på vår server. Intrånget kom helt oväntat och låg utanför våra förväntningar med examensarbetet. Visst hade vi satt upp en server som vi själva skulle prova att hacka oss in på, men att någon annan också skulle försöka hade vi inte förväntat oss. Det visade sig ändå bli ett bra problem då vi fick jobba med något som verkligen har hänt, och som vi själva egentligen inte kan producera exakt. Nu visste vi ju ingenting – inte vad de var ute efter eller vad de hade gjort. Vi började arbetet med att försöka förstå exakt vad de hade gjort utifrån loggarna. Många frågor uppstod; Vilka var det som hade gjort detta? Var kom de ifrån? Vad hade de för avsikter?

## B.1 Detta gjordes vid intrånget

Inkräktaren som först hittade till vår server gjorde det med hjälp av någon form av program som scannar nätet och som testar ifall datorn, som den undersöker, kör Internet Information Server 4.0 (IIS). Självklart kördes IIS på vår server. Nästa steg för inkräktaren var att kontrollera ifall IIS var opatchad, d v s att inga säkerhetshöjande tillägg har installerats. Detta görs genom att köra ett script eller program som testar olika former av kommandon där IIS Unicode Exploit [12] utförs mot IIS. Inkräktaren finner att vår server verkligen är opatchad, och börjar med lite egna kommandon mot IIS. Allra sist hämtar inkräktaren en fil utifrån. Denna fil hette ServUDaemon.exe. Filen är ett program som när den körs är en ftp-server. Det innebär att när programmet startas, med hjälp av IIS Unicode Exploit, kommer vår server att agera som en ftp-server åt inkräktaren.

När vi kollade i ServUDaemon's inställningsfil såg vi att ftp-servern lyssnade på port 6780. Det innebar att vi inte hade några loggar som övervakat trafiken till och från denna ftp-server. Men vi vet ändå att uppladdning har skett ända tills hårddisken blev full. Detta kan vi se dels i Windows Event Viewer och dels i Windows Utforskare. Detta skedde alltså på torsdagen den 14 februari och fortsatte på fredagen, oupptäckt av oss. Lördagen var lugn i avseende på loggposter i Windows Event Viewer och i IIS-loggen. Söndag kväll rent av exploderade loggarna i loggposter. Detta berodde på att Nimda-masken hade hittat vår server och infekterat den. På måndagen upptäckte vi intrånget. Tydligen hade även Karlstads universitets systemadministratör också upptäckt något skumt på måndagen eftersom Internet inte fungerade. Vi fick senare reda på att systemadministratören hade dragit ut kontakten till Internet för

vårt lilla nät eftersom onormalt mycket trafik hade flutit till och från maskinerna i detta nätverk, framför allt vår server.

## B.2 W32.Nimda.A@mm

W32.Nimda.A@mm (Nimda) är en mask (eng. worm) som genom att skicka mängder av e-post utnyttjar många metoder för att sprida sig själv. Masken skickar ut sig själv genom e-post, letar efter öppna utdelade nätverk, försöker kopiera sig själv till opatchade eller redan känsliga Microsoft IIS webservers, och är som ett slags virus som infekterar både lokala filer och filer på avlägsna nätverk som är utdelade.

När masken anländer genom e-post utnyttjar den MIME exploit [27] som tillåter viruset att exekvera bara genom att läsa eller förhandsgranska filen. Om en användare på ett system besöker en infekterad webserver, kommer användaren att bli erbjuden att ladda ner en .eml-fil (Outlook Express), som innehåller filen bifogad. Masken kommer även att skapa öppna nätverk, genom att dela ut på den infekterade datorn, som tillåter åtkomst av systemet. Under denna process skapar masken ett gästkonto (eng. guest account), med administratörsrättigheter.

Genom att använda Nimda-masken som en leveransmekanism har en angripare möjligheten att infektera en sårbar IIS-server över nätverket och därmed skapa ett lokalt konto med administratörsrättigheter på den server som fungerar som måltavla, oberoende av vilken enhet (eng. drive) som IIS-servern är installerad på.

Masken söker efter webservers genom att använda slumpmässiga genererade IP-adresser. Genom att använda ”unicode web traversal exploit”, kopierar masken sig själv till web-servern som admin.dll via tftp. Infekterade maskiner skapar en lyssnande tftp-server för att överföra en kopia av masken. Denna fil exekveras sedan på web-servern och kopieras till mängder av platser. Dessutom försöker masken exploatera redan infekterade web-servers genom att använda filerna root.exe eller cmd.exe som finns i avlägsna exekverbara web-kataloger.

Har datorn en gång infekterats av Nimda är det möjligt att systemet har blivit accessat av en otillåten användare.

Skador som orsakas av Nimda:

- Stora mängder e-post skickas ut. Masken använder MAPI[5] för att skicka ut sig själv som readme.exe (readme.exe syns dock inte i mailet som en bifogad fil).
- Modifiering av filer. Nimda ersätter ett stort antal legitima filer med sig själv.
- Försämring av prestanda. Masken orsakar troligtvis att systemet blir slött.
- Förändring av säkerhetsinställningar. C drive öppnas och blir utdelad.

[22]

## B.3 Kommentarer

När angreppet utfördes hade vi dålig koll på hur inställningarna i IIS skulle vara. Detta gjorde att IIS-loggen inte var så utförlig som den kan vara. På grund av detta var vårt största problem att veta vad angriparna hade haft för argument när de kört t ex cmd.exe. Detta medförde dessutom att detektivarbetet i att klargöra vad som skett blev en jobbig och långdragen process. Man fick kontrollera både i IIS-loggen och i NT-loggarna för få lite klarhet om händelserna. Dessutom var tidsangivelserna i IIS-loggen och NT-loggarna inte synkroniserade. IIS-loggen angav tiden en timme tidigare än NT-loggen. Ett annat bekymmer som IIS-loggen har är att den endast loggar när en begäran är klar, inte när den har begärts. Detta gjorde att vissa loggar som syntes i NTs säkerhetslogg inte kunde hittas i IIS-loggen.

För att undvika att fortsatta intrångsförsök utifrån gjordes, installerade vi ZoneAlarm. ZoneAlarm är en brandvägg (eng. firewall) som kan stoppa all inkommande och utgående trafik, men med inställningsmöjligheter att tillåta trafik från vissa IP-adresser eller en grupp av IP-adresser. Vi ställde in så att endast de datorer som vi själva skulle göra intrång med hade tillåtelse att komma in med trafik till servern. Mycket riktigt slapp vi fler allvarliga intrång. Däremot råkade vi tillåta IIS att fungera som ett serverprogram för alla Internetanvändare vilket medförde att vi har haft ett antal begäran (eng. requests) till web-servern. En del som försökt utnyttja IIS unicode buggen och en del som lyckats att ändra web-servern startsida med hjälp av IIS unicode buggen.

[5] Messaging Application Programming Interface, en samling med standardiserade C-funktioner som underlättar för program att skapa och skicka e-post.

# C IIS-logg

Nedan visar ett exempel på tre poster i IIS-loggen. Alla tre är från när vi gjorde ett intrångsförsök med IISHack, se Kapitel 6.5. Varje post består av följande fält:

**date time c-ip cs-username s-sitename s-computername s-ip cs-method cs-uri-stem cs-uri-query sc-status sc-win32-status sc-bytes cs-bytes time-taken s-port cs-version cs(User-Agent) cs(Cookie) cs(Referer).**

Första posten visar hur det ser ut när IISHack försöker hitta en exekverbar katalog. Nästa post visar när cmd.exe kopieras till den funna exekverbara katalogen och får ett annat namn. Sista posten visar hur IISHack skapar filen eeyerulez.asp. Visserligen är denna post avkortad av utrymmesskäl, men de viktiga delarna är fortfarande med. T ex finns argumentet till cmd.exe kvar som visar hur man kan skapa en fil på system.

2002-04-29 07:25:24 193.10.221.184 - W3SVC1 TDSPDC 193.10.221.180 GET /scripts/../../../../../../../..//winnt/system32/cmd.exe /c%20dir 200 0 615 120 81 80 HTTP/ 1.0 ---

2002-04-29 07:25:24 193.10.221.184 - W3SVC1 TDSPDC 193.10.221.180 GET /scripts/../../../../../../../..//winnt/system32/cmd.exe /c%20copy%20C:\winnt\system32\cmd.exe%20C:\Inetpub\scripts\eeyehack.exe 502 0 382 183 90 80 HTTP/1.0 - - -

2002-04-29 07:25:24 193.10.221.184 - W3SVC1 TDSPDC 193.10.221.180 GET /scripts/eeyerulez.asp - 404 0 604 37 0 80 HTTP/1.0 - - -

2002-04-29 07:25:25 193.10.221.184 - W3SVC1 TDSPDC 193.10.221.180 GET /scripts/../../../../../../../..//Inetpub\scripts/eeyehack.exe /c%20echo%20^<SCRIPT%20LANGUAGE%3d"

…

ð×¼wP]‹Åƒ À%133Éf¹9%02€0…@âúm……Ú%0ei%04i%3dŽ……Â%06ºzð %06B %0ep¶L%3c‡……%0e,%0cƒ%06C %06B gq%06B %05º…ñ´%08,ÕzÐ %06}…ñ³%0eU%3c•¢……¶Ew%2b%05º…ñ–×Ò×zÐ…ß%06}…ñŸ%0cƒ%06C naÂnO¶Ew%2b%05º…ñ,%0c»%06C nuÂ%0c»%06C ¶EÕÅÕÅÕzÐÁ%16ï•zðÉÖzÐ±ï‡ÖzÐ½¶EÒÕ5‰.Ý.Å.ÚÍÕÒÓ(ÓzÐ‰ÍÕÒ(Ó(ÓzÐ‰Í5Á%0c,ÒzÐ ¶E%0eÀÑ%0cÂ¹%0cÂÅ%0eÀå%0cÂ½%3d„„……%0cÂ©ÒÒ¶EÕÕÕÅÕÍÕÕzÐ•ÍÕÕÖzÐµ%0e]¶E1 ÕDm ÕzÐ™%0eu%15¶E%0eM0 ÕÕÒÔÕzðÝzÐ¥%06º„ù§¶EÕÒz²ÓzðÝzÐ¡ŽEñª¶EÕz²ÓÖzÐÅïÕzÐ-nB¶EÕ1 ÕÓÖzÐ¹Ò¶LÔÕÓzðÙzÐ©ïÕzÐ-n,ÕzÐ‘Fzzzz6ºtò8²tò……îà÷ëàé¶·«áéé…ÆéêöàÍäëáéà…Æ÷àäñàÕìõà…Æ÷àäñàÕ÷êæàööÄ…Àýìñ Õ÷êæàöö…ÂàñÖñä÷ñðõÌëãêÄ…ÂéêçäéÄéêêæ…ÕààîËäèàáÕìõà…×àäáÃìéà…Öéààõ…Ò÷ìñàÃìéà……òö·Ú¶·«áéé…äæææàõñ…çìëá…éìöñàë…÷àæó…öàëá…öêæîàñ……æèá«àýà……‡…š®……"%20RUNAT%3d"Server"^>%20^</SCRIPT^>%20>%20C:\Inetpub\scripts\eeyerulez.asp 502 0 355 3104

"%20RUNAT%3d"Server"^>%20^</SCRIPT^>%20>%20C:\Inetpub\scripts\eeyerulez.asp 502 0 355 3104 330 80 HTTP/1.0 - - -