CHALMERS

D3SI

d3-projekt: Säkerhetsinventering

http://d3si.berra.se

Erik Bergström
Marko Jusufovic
Martin Persson
Per Ängsved
Fredrik Johansson
Martin Millnert
Yildirim Zaynal

Rapport - Säkerhetsinventering - 2006
CHALMERS TEKNISKA HÖGSKOLA
Avdelningen för Datorvetenskap
Göteborg 18 maj 2006

Sammanfattning

Tanken med projektet, säkerhetsinventering, uppstod då de lokala administratörerna av datorsystemen på sektionerna D, IT, E och MV på Chalmers tekniska högskola, MEDIC, bestämde sig för att de behövde en genomgång av och ett utlåtande om deras säkerhet från studenter och från hackare. Efter att ha blivit hackade flera gånger så fick de ett ökat begär av att förstärka sin säkerhet. Ett första steget var att byta ut det gamla studentdatorsystemet mot StuDAT, där många förbättringar genomfördes. Ett senare steget var att anföra en grupp studenter att intrångstesta deras system och kartlägga eventuella säkerhetssvagheter. Detta ger MEDIC en mer realistisk syn på deras system och vilka typer av intrång som är möjliga från insidan och från utsidan. Vårt jobb som intrångstestare var definerat i sammarbete med MEDIC, däribland vad vi var och inte var tillåtna att göra eller testa. Att arbeta etiskt har varit ett ledord i vårt projekt och har återhållit vårt arbete i vissa aspekter. Då det var möjligt att kompromettera den personliga integriteten hos en användare var vi tvugna att avbryta. Bara genom att bryta mot de etiska reglerna kunde vi gräva djupare i systemet, vilket vi inte fick, vi kunde då enbart spekulera och skapa våra egna hypoteser.

Abstract

The aim of this project, Security Inventory, owned its purpose when the local computer administrators of the sections D,IT,E and MV on Chalmers university of technology, called MEDIC, decided that they need an opinion of their security coming from the local students and from remote hackers. By being hacked several times they felt an important urge to fortify their security. The first step taken was to change to old system into StuDAT, where many improvements where made. The second step taken was to assign a group of students to penetration test their system and map their security flaws. This would give MEDIC a more realistic view of their system and what kind of attacks that are possible from the inside and from the outside. Our job as penetration testers, were defined by cooperation with MEDIC, in what we where allowed to do/try or not. Working ethically has been a keyword in our project, and have possibly detained our work in some manners. But as it was possible to come to a point where the privacy of the users could be broken, we were forced to abort. Only by breaking the ethical code we were able to dig more deeply into the security of the system. As we where not allowed, we were only to speculate and create our own hypothesis.

INNEHÅLLSFÖRTECKNING

Kapitel 1

Inledning

1.1 Problembakgrund

MEDIC har upprepade gånger blivit hackade. På senare år skedde två stora hack där den onde hackaren knäckte ett dåligt patchat Solaris-system och tog sig in och bytte ut centralt placerade binärer så att lösenord kunde samlas in från samtliga användare i ett och samma svep. För att råda bot på detta designade man om studentsystemet och isolerade studentdatorerna dels från varandra, och dels i någon mån från resten av systemet. Man använder inte längre centralt placerade binärer på samma sätt som tidigare (SSH etc), och man kan inte logga in på varje enskild dator längre vilket man kunde tidigare.

1.2 Uppgift

Gå igenom IT-säkerheten i MEDICs datorsystem. Påvisa möjliga säkerhetsproblem och ge förslag på åtgärdsplan på dessa. I uppdraget ingick både den tekniska översynen men även sådant som en kunskapsinventering hos personalen.

1.3 Syfte

Syftet med detta projekt var att genomföra en säkerhetsinventering av MEDICs system, för att eventuellt finna svagheter och komma med konstruktiv kritik. Detta skall i sin förlängning leda till att MEDIC kan uppdatera och förbättra systemet.

1.4 Avgränsningar

- Etisk hacking, innebärandes att vi ej fick inkräkta på folks privatliv. Vi fick ej använda studenters konton som språngbräda för vidare attacker. Vi fick ej avlyssna personlig trafik.

- Användare (studenter) tappar bort lösenord, detta skall ignoreras i arbetet.

- För att få göra attacker som misstänktes kunna resultera i systems nertid var vi tvugna att först inhämta tillstånd att utföra testet.
- Vi fick inte röra alla system.

Kapitel 2

Tidigare arbeten

2.1 Hacking

Att "hacka" ett givet system är en mödosam uppgift om den görs väl. Det är ett stort och omfattande jobb som kan göras av den som så önskar.
Att hacka ett system kan delas in i fyra olika steg, från det att man tar reda på så mycket som möjligt om systemet till att attacken är utförd och alla spår är undanröjda.

2.1.1 Rekognoscering

Rekognoscering går ut på att ta reda på så mycket som möjligt om systemet man ska attackera. Viktigt steg för dem som vill attackera ett specifikt system. "Script-kiddies" som bara vill ta sig in var som helst har inte mycket nytta av detta steg.

- Kolla upp administratörernas kontonamn och riktiga namn, titta på bilder de har och personlig informatiom. Man kan hitta information som gör att man vid senare tillfälle kan gissa rätt lösenord till konton.
- Surfa in på målets hemsida, slå i kataloger, sök i media- och jobbannonser och newsgroups efter administratörers postningar för att se ifall de avslöjar detaljer om sitt system.
- Whois[22] kan ge telefonnummer, adresser, DNS:er.
- Zonöverföring med DNS kan ge alla värdar som finns, åtminstone de som är uppmappade med DNS.
- Fysisk tillgång: titta omkring på området, trashing (leta efter nyttig information i sopor), piggy-backing (följa efter personer som har tillgång till särskilda lokaler.

2.1.2 Skanning

Att skanna: att gå genom och besiktiga punkt för punkt, att noga undersöka, att titta ingående på eller i, att rannsaka.

Anledningar till att skanna:

- Ta reda på vilka datorer som finns tillgängliga och åtkomliga på ett nätverk.
- Ta reda på vilka portar som är öppna för anslutning och gissa, med ett programs hjälp, vad för program som kör där.
- Identifiera vilket operativsystem som körs.

Skanning är en nödvändighet för att kunna komma åt system utifrån. I princip anger man vilket mål, dator eller nätverk samt vilka portar på dessa man vill skanna. Man kan få problem om brandväggar är i vägen. De kan filtrera vissa portar och datorer vilket gör att informationen man får fram inte nödvändigtvis är tillförlitlig.
Det finns bra och dålig skanning om man inte vill bli upptäckt.

- Dålig skanning är t.ex. ARP-stormar (Address Resolution Protocol) på ett LAN (Local Area Network) eller parallell skanning av ett större nät i hög fart från en enda värddator utan någon ansträngning att dölja sig.
- Bra skanning är den skanning som ger dig den informationen du behöver eller vill ha utan att du blir upptäckt.

Det finns många olika program, alla med olika specialiteter: Nmap[14] är mest känt. Det är det bästa programmet för att ta reda på vilka datorer som är tillgängliga på nätverket och vilka tjänster de kör. Cheops-ng[3] används för att rita kartor över nätverkstopologier. Firewalk[6] används för att hitta filtrerade portar i en brandvägg. Xprobe[23] är bra på att identifiera OS. Programmet pOf[17] gör passiv OS-identifiering genom att titta på trafik som läcker från de tilltänkta mottagar- och sändarparet.

2.1.3 Skaffa tillgång till systemet

Det finns tre grader av tillgång till ett system:

- Fjärrtillgång (“remote access”) till systemet.
- Ett användarkonto, oftast med begränsade rättigheter på systemet.
- Fysisk tillgång till systemet.

För att få tillgång till ett system kan man utnyttja svagheter i tjänsterna den kör. För att utnyttja svagheter i system kan man skriva program som gör det åt en. Dessa, oftast väldigt små programkodsnuttar, kallas för “exploits”. Det finns olika typer av exploits: exploits som används då man har ett konto på systemet och exploits som utnyttjas av en hackare som har fjärrtillgång. Fjärrexploits är farligare än lokala exploits i den bemärkelsen att vem som helst som har nätverkstillgång till ett system kan attackera det. De kan ge lika mycket rättigheter i systemet i fråga, men användandet av de lokala exploitsen begränsas till de personer som har tillgång till ett system.
Det finns olika ändamål med exploits:

- De kan få en dator att hänga sig, en så kallad DoS-attack (“Denial of Service”).
- De kan ge hackaren ökade rättigheter.
- De kan ge otillbörlig tillgång.

Vilken typ av attack som kan utföras beror vanligen på sårbarhetens natur.
Ett annat sätt att komma in på system är genom att knäcka lösenord till konton. Det finns ett antal metoder och program man kan använda sig av för detta ändamål.

- Man kan skapa listor med vanligt förekommande lösenord eller använda sig av en så kallad “brute force”-attack då alla möjliga kombinationer av lösenord skapas. Lösenorden testas ett åt gången för att se om det ger tillgång.
- Har man tillgång till en fil innehållandes det krypterade lösenordet kan man använda sig av program som John The Ripper[9] för att ta reda på lösenorden, eller egentligen den teckenföljd som genererar hash:en i fråga.

2.1.4 Behålla tillgång och undanröja spår

Om man har fått tillgång till ett system och är angelägen att behålla den kan man installera ett så kallat “rootkit”. Ett rootkit är oftast en liten samling program som ersätter originalen och döljer sin egen samt den illasinnade hackarens existens från användare och administratörer. Om rootkitet också innehåller en bakdörr kan man ansluta till systemet igen utifrån. Ett vanligt program för att sätta upp en backdoor är Netcat[12]. Vid intrång måste attackeraren akta sig från att fastna i loggar. Det farligaste för den illasinnade hackaren är att vara spårbar och sedermera kunna bli upptäckt. IDS (Intrusion Detection System) är system som analyserar logdata från t.ex. datorer eller nätverksövervakning och ger varningar när någonting ser misstänkt ut.

Kapitel 3

Att hacka en driftenhet

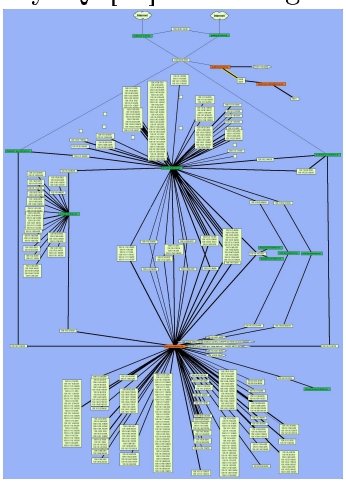
3.1 Organisation

MEDIC-historia[24]
När Data- och Informationsteknik (DoIT) bildades 2002 fanns två driftorganisationer; en på Elektrosektionen samt en på Matematiska Vetenskaper. Dessa beslutade man slå ihop till en gemensam organisation för det nya DoIT, E samt MV: MEDIC. Under 2002 byggdes MEDIC-organisationen upp i formen av ett projekt och den förste januari 2003 skulle den vara igång. MEDIC skulle ha två chefer, en från vardera tidigare driftorganisationer, där den ena ansvarade för teknisk drift och den andra för personal och ekonomi.
Om MEDIC
MEDIC bestod av omkring tjugo fast anställda personer samt ca. sju Helpesk / diverse-personer. Den tekniska driften var indelad i tre ansvarsområden med var sin chef. De tre cheferna var vid slutet Andreas Jonasson, Frans Englund samt Henrik Lindgren.
Datornätverket som finns i EDIT-huset vilket alla MEDICs tjänster och datorer använder driftades inte av MEDIC självt, utan driftades och driftas fortfarande av CDG[2] (Chalmers Datornätgrupp). Däremot var det så att en del personal i MEDIC även jobbade i CDG och hade därför bra koll systemen däremellan. I verkligheten är det så att gränserna mellan olika organisationer inom IT-driften är mer eller mindre artificiella och påhittade, jämfört med den tekniska realiteten. Detta är möjligen också en av orsakerna till de organisatoriska problem som existerar, men det är inte detta projekts uppgift att undersöka detta.

3.2 Infrastruktur

MEDIC driftade studentdatorsystemet för studenterna på D, IT, E och MV. Man hade en filserver vid namn Hades som höll alla studenters användarkontons hemkataloger, såväl som labkonton och en betydande mängd personalkonton. Man hade en kombinerad skrivar- och webbserver kallad Osiris, en stor mängd förhållandevis isolerade studentdatorer samt ett antal beräkningsdatorer (Jackass, Fairy, Persephone) och även fjärrinloggningsmaskiner (Remote1,2,3,4). MEDIC drev även epostservern

mail.chalmers.se, mail.chalmers.se började som mail.medic.chalmers.se men strax efter att ny fin större hårdvara köptes in fick man ta över ett centralt epostansvar i ungefär samma veva som samtliga civilingenjörs- samt arkitektstudenter fick "@student.chalmers.se" tillagt till sina epostadresser. Till denna eposttjänst finns en webbpostserver kopplad som kör det populära öppna källkodsprogrammet SquirrelMail[20]. Man driftade en databasserver, delphi, med databasserverprogramvara från Oracle[15], MySQL[10] och PostgreSQL[18].



Nätkarta som visar subnätsindelning och routning inom Chalmers nätverk.
På bilden ovan visas en nätkarta som CDG tillhandahåller[13], där man på ett hjälpligt sätt kan se subnäten och till vilka routra de är anslutna.

3.3 Projektet D3SI

2005-09-02 - 2005-09-09

- Vi har vårt första möte med vår beställare (Henrik Lindgren). Under mötet avhandlas riktlinjer och upplägg av projektet.
- Vi framställer utvecklingsplan, kravspecifikation, konstruktionsdokument och verifieringsplan.
- Möte med MEDIC där frågor kring systemet behandlas och vissa begränsningar av projektet införs, begränsningar gällande studenters/anställdas integritet. Det framkommer att spårbara intrång på systemet är ett mindre viktigt problem. Sekretessavtal nämns.
- Fem föreläsningar hålls: Att arbeta i team, projektledning, kravhantering och uppföljning av projektplanering.
- Wikin skapas.[4]
- Anton Crona, som var placerad i projektet från början, meddelade att han inte fick göra D3-projektet på grund av för få lästa poäng. Ny person var Fredrik Johansson.

2005-10-06 - 2005-10-27

- Första mötet med vår handledare Per Zaring. Under mötet diskuteras andra projektformer. Ett kompetensdokument över projektmedlemmarna samt ett direkt avtal med MEDIC är övriga punkter på dagordningen.
- Kunskapsinventering av projektmedlemmarna.
- Projektplanen uppdateras och överlämnas till handledare och examinator.

2005-11-06 - 2005-11-07

- Förberedelser inför redovisning och genomförd muntlig redovisning av projektet.

2005-12-01 - 2005-12-27

- Social manipulation ("Social Engineering") studeras.
- Studerade gamla intrång på MEDICs system

2006-01-19 - 2001-01-28

- Två möten hålls där projektets framtid diskuteras och arbetsuppgifter delas ut.
- Skanning av MEDICs nät och övrig research påbörjas.

2006-02-01 - 2006-02-06

- Möten med förberedelser till halvtidsredovisningen.
- Halvtidsredovisning av projektet.
- Fortsatt research.
- Fredrik Svensson blev petad från projektet, trots att gruppen precis hade enats om att ge Fredrik en chans till, av examinator och handledare. Fredrik hade missat obligatoriska moment under hösten.

2006-02-13 - 2006-02-28

- Möte. Arbetsuppgifter delas ut. Övrigt som diskuteras: Lista över datorer på Chalmers, presentation för MEDIC, öppna accesspunkter för trådlöst wlan, resultat av skanningar samt att få total kontroll över en klientdator.
- Kontakt med MEDIC tas angående datorer i grupprummet.
- Fortsatt research om hacking och om Kerberos, ett autentiseringssystem.
- Konfiguration av brygga för avlyssning mellan datorer, se appendix N.

2006-03-12 - 2006-03-18

- Möte: Uppdelning av arbete i grupper om två personer samt diskussion om avlyssningen som genomförts. (Se bilaga O)

- Fortsatta scans av systemet.
- Fortsatt research: Att starta upp en dator genom nätverket, remote exploits, BIOS-cracking, läsning av SANS säkerhetsböcker.
- Genomgång av loggar från avlyssning.

2006-03-20 - 2006-03-25

- Möte.
- Möte med beställare, Henrik Lindgren. Vi informeras om MEDICs omorganisation. Kunskapsinventeringen utgår då MEDIC inte lägre existerar utan övergått till att vara en del av ITS.
- Fortsatt research: SANS[25] säkerhetsböcker, remote exploits, verktyg för att cracka lösenord.
- Vi fick administratörsrättigheter på UNIX genom att starta upp genom nätverket och ändra administratörslösenordet (lite klumpigt :-)). Se appendix A.
- Lösenordsfiler tagna från Windowsdator. Cracking ska ta fram lösenord för administratörsrättigheter. Se appendix B.

2006-03-26 - 2006-03-31

- Möte där hackningen av UNIX- och Windowsdatorerna diskuteras; hur det gått till och hur det kan utnyttjas. Se appendix O.
- Fortsatt research: SANS säkerhetsböcker, verktyg för utförlig skanning, remote exploits, test av diverse hackningsverktyg.
- Vi fick administratörsrättigheter på Windowsdator. Se appendix B.
- BIOS-lösenord framtaget. Se appendix C.
- Installation av Linux (Gentoo[7]) på en av datorerna i grupprummet.

2006-04-01 - 2006-04-08

- Fortsatt research: SANS säkerhetsböcker, att cracka lösenord, verktyg för att hacka.

2006-04-09 - 2006-04-15

- Möte där det diskuteras hur arbetet fortskridit och hur de utökade rättigheterna på systemet kan utnyttjas.
- Fortsatt research: SANS säkerhetsböcker, mailinglistor, verktyg för att hacka.

2006-04-16 - 2006-04-22

- Möte.

- Fortsatt research: SANS säkerhetsböcker.
- Skanning av systemet utifrån.

2006-04-24 - 2006-04-30

- Möte.
- Fortsatt research: SANS säkerhetsböcker, mailinglistor, remote exploits, google hacking, RSS ("Radio and Space Science") subnät, att cracka lösenord, Ladok betygsdatabas,
- Skannat klart systemet utifrån.
- Skannat systemet inifrån.
- Warwalking hos Fysik för att hitta accesspunkter.
- Uppgradering av wikin.

2006-05-01 - 2006-05-04

- Kontrollerat larm på datorer i EDIT och Idéläran. Se appendix P.
- Warwalking i EDIT för att hitta öppna accesspunkter.
- Undersökt intressanta subnät.
- Fortsatt research: exploits, att cracka lösenord, DNS cache poisoning, att införskaffa mailadresser.
- Diskuterat projektet med beställare.

2006-05-05 - 2006-05-09

- Fortsatt research: exploits
- Förberedelser inför slutredovisningen: Skrivit ner den muntliga redovisningen, förberett datorer för demonstrationer.
- Slutredovisning
- Demonstration

2006-05-12 - 2006-05-18

- Rapportskriveri

Kapitel 4

Resultat

4.1 Labkonton

Typ av tjänst

Vid kursstarten delas labkonton ut till de som skall deltaga i kursen. Dessa skall användas vid laborationsmoment.

Hur hittades den?

Genom att kolla upp vilka kurser som innehåller labmoment och sedan gå till första föreläsningen och skriva upp sig på ett labbkonto med något påhittat namn.

Vad för sårbarhet?

Man kan använda dessa labkonton som språngbräda för vidare illasinnade operationer i Chalmers nätverk. Svagheten i systemet med labkonton grundar sig på att det finns väldigt många kurser som har labmoment där de som delar ut labkonton inte har ett system för att kontrollera vilka som tar ett konto. Man skriver under ett papper med ett personnummer - inte särskilt svårt att förfalska.

Hur viktigt och allvarligt är problemet?

Detta är en ganska allvarlig svaghet då man kan få tillgång till konton genom vilka man är totalt ospårbar i systemet och detta är ett hot.

Förslag på åtgärder

Något form av system till alla som delar ut labkonton för att kunna kontrollera vilka det är som de delar ut labkonton till. En checklista där de bockar av alla registrerade studenter och därefter delar ut kontot är ett exempel. Ett annat alternativ är att slopa labkonton helt och bygga in alla de funktioner som kräver labkonton i det vanliga studentdatorsystemet StuDAT.

4.2 WLAN

Typ av tjänst

Trådlösa nätverk som finns tillgängliga på olika platser runt omkring på Chalmers.

Vad för sårbarhet?

Trafiken över det trådlösa nätverket sänds ofta okrypterat. Man kan avlyssna trafik som sänds och få tag på användbar information som t.ex. lösenord och användarnamn.
Det finns olika ställen på Chalmers där anställda som har satt upp egna små trådlösa accesspunkter och sedan inte krypterat dem. Dessa nätverk står öppna för alla oavsett om man är en student på Chalmers eller inte. Ansluter man till dem får man tillgång till Chalmers nätverk från insidan. Vissa accesspunkter är t.o.m så snälla att de ger full obehindrad NAT:ad (Network Address Translation) "layer3"[16]-tillgång, och utan några särskilda svårigheter är man ospårbar, återigen.

Hur hittades den?

Genom att gå runt på Chalmers med en laptop ("warwalking") som har ett trådlöst-nätverkskort, kan man hitta dessa nätverk alldeles för snabbt och alldeles för enkelt.

Hur viktigt och allvarligt är problemet?

Vi bedömmer detta problem vara ganska stort eftersom det erbjuder en person som vill ställa till med problem en möjlighet att få tillgång till nätverket och dessutom vara helt anonym och ospårbar på ett enkelt sätt.

Förslag på åtgärder

Kontrollera vilka det är som satt upp de oskyddade nätverken och se till att de krypterar dem.

4.3 Olarmade datorer

Typ av tjänst?

De datorer och skärmar som finns i första hand i datorsalarna är larmade för stöldskydd samt även för intrångsskydd.

Hur hittades de olarmade datorerna?

Datorerna står i labsalar och i grupprum. Att finna dem är inte svårt. Genom att gå runt på Chalmers och titta in i labsalar och datarum kan man se vilka som inte är kopplade till något larm. Att sedan komma in i en labbsal är inte heller svårt då trafiken in och ur dem är hög under skoldagar, vilket möjliggör "piggy-backing".

Vad för sårbarhet?

En olarmad dator eller bildskärm kan med lätthet packas ner i en väska och stjälas. När ett larm utlöser på en larmad dator kvällstid rycker Securitas ut. Deras utryckningstid varierar beroende på var väktaren befinner sig för tillfället och om han har andra larm som gått före. Vid ett tillfälle uppmättes en responstid på 45 minuter vilket vi tycker är oacceptabelt. Sårbarheten möjliggör också nollställning av BIOS genom öppnande av datorn, vilket vanligen förhindras av larmet.

Hur viktigt och allvarligt är problemet?

Vi bedömmer detta problem vara stort då man med lite kunskap och tillgång till en olarmad dator kan få total tillgång till studentkonton.

Förslag på åtgärder

Se över samtliga datorer, se till att de är ordentligt larmade.

4.4 Osäkra datorer

Typ av tjänst?

Datorer som startar med nätverksuppstart aktiverat och som har resetknappen inkopplad.

Hur hittades sårbarheten?

Vid omstart av en dator ser man sökning efter en DHCP/BOOTP-server. Detta är indikationer på att nätverksboot fungerar.

Vad för sårbarhet?

Med lite kunskap kan man få total tillgång till studentkonton och rootbehörighet på datorn.

Hur viktigt och allvarligt är problemet?

Med rootbehörighet på en RedHat-klientmaskin får man möjlighet att utnyttja svagheter i nätverksfilsystemet. Man kommer åt alla studentkonton och deras personliga saker så som lösenord som kan finnas sparade i hemkatalogen, webbläsarens inställningsfiler t.ex. Det går att ändra på diverse webbsidor tex www.dtek.chalmers.se.

Förslag på åtgärder

Vi föreslår att man skall koppla ur resetknappen på samtliga klientdatorer samt inaktivera möjligheten till nätverksuppstart.

4.5 Säkerhet vid fjärråtkomst

Typ av tjänst

Möjligheten att kunna erhålla tillgång till MEDICs system från en plats utanför MEDICs och Chalmers system och nätverk.

Hur hittades den?

Genom att aktivt skanna efter tjänster som körs på de olika maskinerna kan man få fram intressanta resultat, som visar vilka tjänster som körs på datorerna inom Chalmers.

Vad för sårbarhet?

Genom att analysera resultat från skanningar kan man bokföra vilka versioner av olika program som sannolikt körs på datorer och vilka operativsystem de har. Detta är i sig inte en sårbarhet, tills dess en svaghet som möjliggör för utnyttjande från distans upptäcks i en av alla dessa tjänster. När detta händer finns det tidsfönster för en anfallare att komma in, antingen med hjälp av publikt tillgängliga sårbarheter ("exploits in the wild") eller med ej publikt tillgängliga sårbarheter, tills dess att en patch finns för sårbarheten och dess att den är installerad på systemet som tillhandahöll den sårbara tjänsten.

Hur viktigt och allvarligt är problemet?

Detta problem är svårfrånkomligt och så länge man har snabba och resoluta rutiner för installation av säkerhetspatchar är hotet minimerat. Det kan finnas enskilda datorer som inte har centraladministrerade system för att installera säkerhetspatchar och dessa är alltid sårbara. Det är viktigt att minimera dessa till antalet och för de som måste finnas kvar se till att det finns goda rutiner för applicerandet av säkerhetspatchar.

Förslag på åtgärder

Det är viktigt att minimera de icke-centraladministrerade systemen till antalet och för de som måste finnas kvar se till att det finns goda rutiner för applicerandet av säkerhetspatchar. Vidare är centraladministrerade system inte i sig säkrare än separat administrerade system, utan först om säkerhetspatchar appliceras snabbt är systemet tillräckligt säkert inom detta området. Görs detta har man gjort vad man kan i patchväg, och det som återstår som försvar mot detta hot är kraftigare brandväggar. Brandväggar ligger dock inte i linje med den öppna universitetskulturen, och är inte första försvarslinjen mot hot från distans. Första försvarslinjen mot distanshot är att se till att det inte existerar några svagheter att utnyttja i första läget, som beskrivet ovan.

4.6 Defaultlösenord

Typ av tjänst

Osäkra skrivare och strömförsörjningsenheter (UPS) har påträffats.

Vad för sårbarhet?

En del skrivare och andra enheter är ej lösenordsskyddade eller skyddade med standardlösenord. Med tillgång till skrivares webbinterface finns möjlighet att utföra DoS-attacker: Stänga av, starta om eller sätta andra felaktiga inställningar. Strömförsörjningsenheter kan startas om eller stängas av helt vilket slår ut alla maskiner kopplade till dem.

Hur hittades de?

Målen har hittats i samband med skanning av nätet för att sedan utsättas för lösenordstestande. Standardlösenorden som har påträffats är så enkla att vi beviljades tillgång efter ett till fem försök.

Hur viktigt och allvarligt är problemet?

Det är av varierande karaktär. En skrivare innebär kanske inte ett direkt säkerhetshot, om man inte kan duplicera utskriftsjobben den gör. En UPS däremot är mer allvarligt, då det skulle kunna betyda dyrbar nedtid om den stängdes av. Skulle det ske en DoS mot dessa system skulle det troligen upptäckas ganska omgående dock, men ändock skulle nedtiden betyda onödiga utgifter i form av felsökning och arbetsförluster.

Förslag på åtgärder

Lösenordsskydda alla liknande system maskiner som redan finns ute i systemet samt inför rutiner för att lösenordsskydda nya enheter som köps in.

4.7 UNIX-Nätverksfilsystem - NFS

Typ av tjänst

Det distribuerade filsystemet.

Hur hittades den?

Problemet var känt sedan tidigare.

Vad för sårbarhet?

När man väl autenticerats vid klientmaskinens uppstart kontrolleras det inte att ens användar- och gruppid i RPC-begäran stämmer när man frågar efter filer. Om man kan ändra på sina uppgifter när man begär något från en filserver som kör NFSv2, så som hades, görs ingen kontroll på att det stämmer och man kan få tillgång till samtliga studentkonton och kataloger som finns i de delade resurser man har tillgång till.

Hur viktigt och allvarligt är problemet?

Vi bedömmer detta problem vara allvarligt då det kan ge en användare möjlighet att gå igenom studenters konton med fullständiga rättigheter och ändra på diverse hemsidor som ligger på filservern.

Förslag på åtgärder

Ett sätt att lösa problemet är att kräva autentisering för varje begäran som görs till NFS-servern, något som är möjligt i NFSv4. Detta skulle dock ge ett stort påslag i form av en kraftigt ökad mängd Kerberos-tickets som skulle krävas, vilket systemet troligen inte är dimensionerat för. En variant av taktik att hantera situationen är att inte ge klientdatorerna mer tillgång än den minimala tillgång de behöver, och separera på resurser som man inte behöver ha tillgång till från en klientmaskin i StuDAT, t.ex sektioners hemsidor och så vidare.

4.8 Servrar

Typ av tjänst

Servrar som tillhandahåller diverse tjänster vilka håller systemet uppe.

Vad för sårbarhet?

Inga större svagheter funna. Maskinerna kör vissa gamla versioner av tjänster vilka har visat vara svåra att knäcka trots deras ålder. Servrarna är även på egna nät separerade från klienter vilket är en stor fördel sett ur säkerhetssynpunkt.

Hur hittades de?

Information om servrar och deras tjänster har tagits fram via omfattande skanning av nätet. Servrar har utsatts för exploits utan större resultat.

Hur viktigt och allvarligt är problemet?

Att servrarna håller en hög säkerhetsnivå är oerhört viktigt då all tänkbar information kan ses/redigeras med rätt rättigheter.

Förslag på åtgärder

Att fortsatt uppdatera system tidigt för att minimera riskerna för intrång. Att hålla servrar separerade från system de inte absolut måste vara länkade till. Här är det även väldigt viktigt att personalen har ett mycket högt säkerhetsmedvetande och inte utför adminstration från "offsite"-system som inte går att lita på.

Kapitel 5

Slutsats

5.1 Resumé

MEDIC uppfanns under samtal våren år 2002, driftsattes i slutet av samma år och avslutades under våren år 2006. Vår projektgrupp hade i uppgift att undersöka säkerheten i MEDICs olika datorsystem. För att klara av den här uppgiften har vi studerat ämnet datorsäkerhet ingående. Vi har i takt med att vi har lärt oss i ämnet applicerat denna kunskap och testat i den mån vi har haft tillåtelse olika aspekter i systemen. Vi har kommit fram till att hotbilden är avsevärt mycket större på plats på Chalmers än den är från Internet.

5.2 Kritisk diskussion

De olika planeringsdokumenten vi skrev i början har vi haft mindre nytta av, till stor del p.g.a. att deras upplägg inte passade vårt projekt. Dock kunde vi, åtminstone med den kunskap vi nu har, planerat mer, utefter vad som passade projektet. Men i början hade vi inte riktigt den kunskap som krävdes för detta, vare sig när det gällde MEDICs system eller datasäkerhet överhuvudtaget. Projektets natur, och vår okunskap, gjorde att vi inte hade någon klar uppfattning om vad vi konkret skulle göra eller uppnå, förutom att vi skulle försöka hitta säkerhetshål, en potentiellt oändlig uppgift. Men allt efter att projektet fortlöpte lärde vi oss mer och mer och fick en klarare bild av vår uppgift. Så här i efterhand kan vi se klarare hur vi kanske borde gått tillväga. En sak vi borde gjort tidigare är att läsa SANS-böckerna, vilka visade sig vara en väldigt god källa till information om hacking.
Från början var det tänkt att vi skulle träffa någon från MEDIC varje månad för att stämma av vad vi gjort o.s.v. Men, delvis, p.g.a. att vi inte gjorde så mycket under hösten, då vi läste flera tidskrävande kurser parallellt blev detta inte av då. En annan sak som försenade vårt arbete var att det tog väldigt lång tid innan vi fick ett grupprum, något som vi har haft väldigt stor nytta av och som underlättat arbetet avsevärt, framförallt att vi hade egna datorer, bl.a. två som var identiska med StuDATs klientdatorer, som vi kunde testa angrepp mot och ifrån och använda i vårt kartläggande av systemet. Denna försening verkade delvis bero på att vi inte förväntades göra något under hösten, och så blev det också.

Projektet skulle önskat haft ytterligare en handledare, en från exempelvis ITS (eller fd MEDIC) som var kunnig på området och som hade kunnat fungera som bollplank och en lots i det område vi arbetat inom. I brist på denne har vi tagit oss fram primärt med hjälp av Google och böckerna från SANS Track 4.
Vår tilldelade handledare har kunnat hjälpa oss med projekt-formalia men vi har saknat sakkunskaplig hjälp som kanske hade varit bra att ha. Som det blev var vi ensamna, lämnade lite åt vårt eget öde, något som ställde stora krav på Millnert som projektledare att få gruppen att jobba framåt utan något konkret stöd från något håll. När så MEDIC, våra beställare, upphörde att existera och vår kontaktperson Henrik flyttade till en annan position och blev onåbar under en lång tid och till synes tappade lite av intresset i projektet, var det svårt att komma framåt.
Det var även tänkt att vi skulle ha en person på MEDIC som vi kunde be om tillåtelse för att testa olika exploits o.dyl., men p.g.a. omorganiseringen fungerade detta inget vidare. Det var inte förrän i slutet av april som vi hade en person vi kunde be om tillåtelse att verkligen få testa potentiella säkerhetshål, och även då fungerade det lite långsamt.

5.2.1 Så hur har vi nu faktiskt arbetat?

Vi kom igång sent och började läsa in oss på datasäkerhetsområdet medan vi väntade på att få tillgång till kontaktpersoner. Från början hade vi inte tillräckliga kunskaper för att dela upp arbetet mellan oss. Arbetet har inte skett i klassisk organiserad projektform där rollerna är klart definierade och hårda deadlines sätts upp då delleveranser ska ske från projektmedlemmarna. Det var orealistiskt att dela upp systemet i arbetsområden och tilldela dem till olika projektmedlemmar från början då vi saknade tillräcklig kunskap om systemet och tillräcklig erfarenheter för att klara av ett område på egen hand. Istället har saker dykt upp allteftersom vi arbetat och vi har samarbetat för att de olika projektmedlemmarna inte var och en hade tillräckliga kunskaper.
Alla projektmedlemmar är överens om att vi har lärt oss ofantligt mycket inom säkerhetsområdet.

Alla medlemmar kommer läsa vidare inom området och skall läsa säkerhetsinriktningen på D.

5.2.2 Projektledarens synpunkter på det hela

Det har varit ett annorlunda projekt och det har varit väldigt krävande att leda, dels på grund av att det har varit mycket ansvar i projektledarrollen då den har varit kritisk i projektets framfart eftersom övrig hjälp lite har lyst med sin frånvaro. Detta har dock varit väldigt givande för mig och jag har utvecklats ordentligt i rollen och växt till mig mycket vilket medlemmarna håller med om.
Jag har gjort flera lärdomar under projektets gång om projektledning när man är "on your own". Konkreta exempel på saker som skulle gjorts annorlunda är t.ex att

initialt jobba mer självständigt - "sätta igång projektet" - och i starten utvärdera kunskapen hos projektmedlemmarna och göra en rekognoscerande exkursion med de en eller två med störst färdigheter i ämnet för att samla övergripande information om målet och bilda ett grepp om projektuppgiften.
Eftersom projektet inte var av klassiskt mått blev en effekt att medlemmarna kände sig lite vilse stundtals när det stod stilla i projektet, detta kan motverkas genom dels mer strikta mötesrutiner och även, hur det nu skulle gå till, bättre organiserat utdelande av arbetsuppgifter.
Jag vidhåller att det är en stor utmaning att leda ett projekt i detta område, då det kräver en mycket stor kunskap för att på ett bra sätt kunna dela ut arbetsuppgifter som är överkomliga för den enskilde projektmedlemmen att ta tag i. Jag har själv behövt läsa vidare i området för att kunna leda gruppen, vilket slåss med behovet att ägna tid åt att leda medlemmarna och fördela arbete, vilket är projektledarens roll som en slags schemaläggande spindel i nätet. För att kunna göra det på ett bra sätt behöver man ha kunskapen initialt, men då det inte är möjligt blir det en betydligt tyngre uppgift och en ordentlig utmaning att hinna med att lära sig själv i tillräckligt hög takt för att överhuvudtaget kunna dela ut uppgifter till medlemmarna. Mycket av det arbete gruppen har gjort har bestått av att lära sig saker, det är egentligen det vi i huvudsak har sysslat med och således har vi lärt oss en fantastiskt massa inom området.

5.3 Fortsatt arbete

En säkerhetsinventering tar aldrig slut. Att upprätthålla säkerheten är ett kontinuerligt arbete. Här följer några exempel på projekt som skulle kunna genomföras efter vårt arbete.

- Aktiv logganalys
  Att skapa alternativt konfigurera ett program för aktiv logganalys för att kunna upptäcka och isolera intrång på systemet.

- Säkerhetsinventering av personal
  Inventerna personalens kunskap vad gäller datasäkerhet och att vid behov utbilda. Att även eventuellt ta fram en säkerhetshandbok.

- D3SI v.2
  Ett projekt liknande vårt men mer specifikt inriktat på vissa tjänster.

REFERENSER

[1] Bios. http://www.bioscentral.com/.

[2] Chalmers datornätgrupp. http://www.cdg.chalmers.se.

[3] Cheops-ng. http://cheops-ng.sourceforge.net/.

[4] D3-projekt: Säkerhetsinventering. http://d3si.berra.se.

[5] Ethereal: A network protocol analyzer. http://www.ethereal.com/.

[6] Firewalk. http://www.packetfactory.net/firewalk/.

[7] Gentoo. http://www.gentoo.org.

[8] Iscs dns. http://www.isc.org/products/DHCP.

[9] John the ripper. http://www.openwall.com/john/.

[10] Mysql. http://www.mysql.com.

[11] Nessus. http://www.nessus.org.

[12] Netcat. http://netcat.sourceforge.net/.

[13] Netmap. http://antwatch.cdg.chalmers.se.

[14] Nmap. http://www.insecure.org/nmap/.

[15] Oracle. http://www.oracle.com.

[16] Osi model. http://en.wikipedia.org/wiki/OSI_model.

[17] p0f. http://lcamtuf.coredump.cx/p0f.shtml.

[18] Postgresql. http://www.postgresql.org.

[19] Pxe-linux. http://syslinux.zytor.com/pxe.php.

[20] Squirrelmail. http://www.squirrelmail.org/.

[21] tftpd. http://www.kernel.org/pub/software/network/tftp/.

[22] Whois. http://www.whois.net/.

[23] Xprobe. `http://xprobe.sourceforge.net/`.

[24] Britt-Marie Henrikson Birgitta Carlsson. *Utvärdering av MEDIC*. Chalmers Planering och Uppföljning, 2006.

[25] Ed Skoudis. *Track 4, Hacker Techniques, exploits and incident handling*. SANS Institute, 2003.

Bilaga A

StuDAT Linux klient-dator säkerhet

Alla StuDAT linux-klienter kör operativsystemet Red hat linux. Ett ganska bra val då Red hat är en intuitiv och en bra distribution. Fördelarna som har gjort Red hat vald av MEDIC är i min åsikt deras patchningsystem i första hand. Andra fördelar som kommer med Red hat är att det är en av de enklaste linux operativsystemen som finns.

Säkerhetsinställningar

En SLK-dator har flera säkerhetsinställningar. Det vanliga är att den är låst och larmad, BIOS:et är lösenordsskyddad och enbart studenter med legimita konton kan logga in. Man kan inte installera program som kräver systemrättigheter, och inte heller ändra systemkonfigurationer. Men man kan däremot läsa vissa systeminställningar, som kan ge mer information om vad som finns installerat hur det är konfigurerat på datorn. Man har inte rättigheter att läsa loggar, vilket är bra.

Roota en SLK-dator

Har man fysisk tillgång till så kan man lätt få root-rättigheter på datorn. Samma metoder kan användas på vilken klientdator som helst. Detta tar vi upp i bilaga I. En annan svaghet funnen är att alla BIOS-lösenord är samma. Lyckas man cracka en dators BIOS-lösenord så har man tillgång till alla.

Då man har lyckats att få utökade rättigheter på SLK-datorn kommer man åt mer än vi förväntade. Att roota en klient dator, där operativsystemet är linux har varit givande, och kan lätt användas som en språngbräda för att exploita systemet i sig mer. Med våran restrikterade handlingar både från etisk håll och från MEDIC så har vi inte kunnat praktisera användning av dessa svagheter. Vi har däremot rapporterat hypoteser och teorier om hur man ska kunna göra eller vad man ska kunna komma åt.

Vad vi kommer åt

Vi kommer åt alla studenternas, vissa labbkonton och även vissa administratörers hemkataloger. Vi hade inte rättigheter att skapa filer i deras kataloger, och fick skriva egen kod som ändrar vårt UID (User Id) till det UID vi vill.

Hypoteser

Tack vare vårat UID program så kan vi vara vem som helst, detta kan vi utnyttja till vår fördel. Se bilaga M för källod. Eftersom konfigurationsfilerna för varje konto ligger i hemkatalogen så kan man välja vad som ska startas automatiskt eller inte. Ett exempel vore att installera en keylogger som kan samla lösenord från varje konto som finns. På detta vis så kan vi få en hög med personliga lösenord både till systemet, remote desktops,ssh, mail och allt som man loggar in på. Ifall lösenord är sparade i Firefox webbläsare, så kan man komma åt den filen som sparar lösenord och utnyttja den.

Bilaga B

StuDAT Windows klient-dator säkerhet

Informationssökning

Vi behöver veta all relevant information om Windows som kan ge oss utökade rättigheter. Det första man tänker på är hur Windows hanterar lösenord och användarnamn. Alla lokala konton sparas i SAM filen, och i Windows Xp så har man utökad säkerheten genom Syskey, som är ett extra lager av kryptering på hashen i SAM filen. SAM filen kan inte ändras eller läsas då Windows är igång. Den nyckel som används för att kryptera lösenord är genererade av Syskey. Denna nyckel krypteras även med en slumpmässigt genererad systemnyckel, och sparas i registret. Krypteringen som utförs på nyckeln är MD5. Anledningen att denna nyckel krypteras är om man kan läsa den i klartext från registret, kan den användas för att dekryptera lösenord. Annan relevant information är att man program som körs automatiskt då Windows startas är initierade ifrån registret.

Säkerhetsinställningar som finns

Administratör konton har blivit avstängd, och enbart konton med begränsade rättigheter finns. Windows är även säkrad genom antivirusprogrammet Etrust, och kör dagliga uppdateringar på virus databaset och även på själva programmet. antivirus programmet avvecklade alla försök av att försöka köra exploits, och program som var ett möjligt hot mot systemet. Vissa tangentbordskommandon är avstängda och man har inte rättigheter att ändra mycket förutom personliga inställningar. Försök av att aktivera administratör kontot var lönlösa, men förmodligen inte omöjliga.

Försök av att utöka sina rättigheter

Då BIOS-lösenordet var knäckt kunde man ändra inställningarna till att boota med cd-skiva. Mer om hur BIOS-lösenordet knäcktes finns på bilaga C. När man har bootat in sig till DOS-läge, har man systemrättigheter. Man kommer åt vilken fil som helst, och kan lägga till och ta bort filer med full frihet. Med program utvecklade för att ändra Windowskonton genom bootskiva, utförde jag försök att aktivera administratörkontot, men har inte lyckats. Eftersom man har full rättigheter så

kommer man åt SAM och registret. Med hjälp av program som Bkhive så kan man dumpa systemnyckeln från registret och sedan dumpa lösenordshashen från SAM filen med hjälp av Pwdump och systemnyckeln. Lösenordshashen kan sedan knäckas med brute force program som t.ex. L0phtcrack, eller om man har tillgång till en rainbowcrack tabell, så går det mycket snabbare. I vårat fall så kvittar detta då vi får enbart tillgång till konton med lägre rättigheter. Vi får försöka skapa ett konto med administratör rättigheter. Genom att använda kommandot Net så kan man skapa användare och lägga till vilken grupp man vill. men tyvärr så lyckades detta inte i DOS-läge, då programmet enbart fungerar i Windows-läge.

Windows systemsvagheter

Det lättaste sättet att få utökade rättigheter vore att hitta en exploit som gav det, men klienterna är dagligen patchade automatiskt, och använder sig av en kraftfull antivirus program. Att försöka hitta exploiter vore lönlöst. Annat definitivt sett att lyckas är att skriva sin egen exploit, men detta kräver en hel del kunskap, och dessutom tid. Detta sätt är väldigt ovanligt för vanliga hackere och script kiddies. Vi har lyckats få full rättigheter genom bootskiva, och detta ska vi utnyttja. Försök av att byta logon.src, Windows skärmsläckare, till till CMD.exe ska ge ge oss CMD konsol med systemrättigheter i Windows. Detta är för att Windows själv kommer starta konsolen, då igentliga skärmsläckaren ska gå igång. På så vis kan man skapa konton utan att behöva logga in. Detta gav inget resultat för att skärmsläckaren för systemet var avstängd.

Windows registret

Ifall vi har tillgång till registret så kan vi lägga till program att autostarta då Windows startas. Tanken är att få en CMD-konsol med systemrättigheter. Försök av att använda Windowsprogrammet Reg för att editera registret från konsol fungerade inte och kunde enbart köras från CMD-läge inne i Windows, och tyvärr inte i DOS-läge. Det ska finnas andra program som man kan använda för att editera registret utanför Windows miljön, men har inte testats. I registret kan man ta bort starten av antivirus programmet, och kan därför stänga av olika program som förhindrar eventuella försök att utöka sina rättigheter.

Admin på SWK-dator

Lyckas man få administratör rättigheter på en S.W.K-dator så finns det inte mycket man kan komma åt. Man får fulla rättigheter att på den datorn enbart. Detta kan utnyttjas på olika sätt, några exempel framgår.

- Filserver: spara filer på datorn, warez.
- Zombie: kan användas vid en distributed DOS attack
- proxyserver: för att dölja sin ip, då man utför en hack.

- keylogger: kan installera keylogger och samla på sig lösenord av studenter som loggar in på den datorn.

Nehebkau.medic.chalmers.se

När man loggar in mot medic från en SWK-dator så autentiseras lösenodet och användarnamnet på servern nehebkau.medic.chalmers.se. Autentiseringsprotokollet som används är Kerberos.

Åtgärder

1. Använda moderkort med dubbla BIOS minnen. När standard BIOS:et är omställd så kan inställningarna sparade på reserv minnet kopieras över till standard minnet.

2. Bättre konfiguration av nätverksboot. Att helt och hållet avaktivera nätverksboot skulle vara säkrast men då tappar man mycket funktionalitet som administratörerna får med nätverksboot.

Bilaga C

BIOS-cracking

För att komma åt BIOS:et så måste man antingen omställa BIOS (Basic Input Output System) inställningarna till standard eller cracka det. Detta kan åstadkommas genom olika sätt. Det första man gör är att ta reda på information, om moderkortet och dess BIOS. Denna information kan senare användas för att åtsadkomma vårt mål.

Informationsökning

BIOS:et information sparas i ett speciellt minne, som kallas för CMOS. Till skillnad från vanlig pc minne så är sparas informationen i även om man stänger av datorn. CMOS-minnet är väldigt litet och kan är vanligen 64kb, och behöver ett batteri för att förvara inställningarna som man ändrar, då datorn är stängd. CMOS minnen kallas även för NVRAM (Non-volatile RAM). CMOS står för "Complementary Metal Oxide Semiconductor", och används för att den har fördelen att förbruka väldigt lite ström, jämfört med andra halvledare. Systemet använder även CMOS checksum som feldetektering. Varje gång BIOS-inställningarna ändras så genereras det en checksum genom att addera alla byte i CMOS minnet, och sparar det i den minst signifikanta byten av summan. När systemet startar om så räknas checksumet igen och jämförs med sparade värdet. Vid fel ger den felmeddelandet "CMOS Checksum Error", och ställer om BIOS:et till standard värden.
Fortsatt sökning på BIOS specifikt(versionnummer och programvara) så får vi reda att det finns bakdörrar. Dessa bakdörrar är tillagda från tillverkarna, och är olika lösenord. Genom lite googling så kan man få fram de lösenord som är för det BIOS:et som vi ska cracka. Mycket information om hur BIOS fungerar i detail finns på bioscentral[1].

Exploita BIOS:et

1. Attack 1:
   Nu när vi vet mer om BIOS:et så vet vi hur vi kan utnyttja dess svagheterna också. Vi vet att inställningarna som man gör (ex. lösenord) sparas med hjälp av lösenordet. Så om man tar bort batteriet ifrån moderkortet så borde

inställningarna ställas om till standard. Detta har MEDIC förebyggt genom att låsa datorn och larma den. Ifall man skulle försöka öppna datorn så går larmet. Detta förebygger även mot stölder.

2. Attack 2:
   Andra svagheten är är felsökningsmetoden CMOS checksum. Ifall vi lyckas att ändra några byte i minnet så kommer inte det beräknade checksumet att vara lika med det sparade. Detta leder till omställning av BIOS:et till standardvärden utan några lösenord. Detta lyckades inte heller, då man behöver root-rättigheter i linux och administratör rättigheter i windows för att kunna ha tillgång till CMOS minnet genom programvara.

3. Attack 3:
   Tredje svagheten som hittades var bakdörrarna. Alla testade lösenord var oeffektiva.

4. Attack 4:
   Alla svagheter som har koppling direkt med BIOS:et var lönlösa. Detta betyder inte att det inte är möjligt. Det finns fortfarande olika sätt, som är indirekta attacker mot BIOS:et. Från svaghet 2 så vet vi att om vi lyckas få utökade rättigheter så kan man både läsa och skriva till BIOS:et. Detta kan vi åtstadkomma genom antingen exploits som ger utökade rättigheter i operativsystemet, eller genom en PXE-boot. Vi vet att pxe-nätverksboot är igång så vi kan komma åt systemet genom nätverksboot. Hur detta görs mer specifikt tar vi hand om i bilaga I. Genom en pxe-boot så låter vi datorn starta ett linuxsystem genom nätverket, som man har root-rättigheter i. har vi rooträttigheter så kan vi komma åt BIOS:et genom programvara. Vi kan antingen ändra checksumet av BIOS:et genom att skriva över vissa delar av BIOS:et eller att cracka det med en BIOS-cracker.

Bilaga D

Konstruktionsdokument

Här ska vi lägga Konstruktionsdokumentet...

Bilaga E

Kravspecifikation

Här ska vi lägga kravspec...

Här ska vi lägga kravspec...

Bilaga F

Utvecklingsplan

Här ska vi lägga utvecklingsplanen...

Här ska vi lägga utvecklingsplanen...

Här ska vi lägga utvecklingsplanen...

Bilaga G

Verifieringsplan

Här ska vi lägga verifieringsplanen...

Här ska vi lägga verifieringsplanen...

Bilaga H

Kompetensbeskrivning

Martin Persson

Förmågor

Programmerat i Java, C, Basic samt Visual Basic. Samt PHP, CSS, HTML. Haskell i skolan. Löjligt bra på att google:a. Allmänna begrepp och termer om säkerhet, lite om hur det funkar. Läser datasäkerhetskursen, massa nytt men en del bekant också. Väldigt små Linux / UNIX-erfarenheter. Lite hemma, och mer i skolan. På gymnasiet gjordes lite Linux-relaterade saker: Installera en Linux-dator, låsa ner den med iptables. Spelar mycket i Windows, kör den automatiska uppdatering. Inga erfarenheter med Windows Server-saker. Har använt Novell, men kommer inte ihåg det alls, förutom att det var slött.

Vilja

Vill lära sig Linux/UNIX-delen mycket mer. Likt det vi läser i datasäkerhetskursen nu. Lära sig mer praktiskt hur man faktiskt skyddar sig, och hur attacker utförs, så att man skall veta hur man skyddar sig mer.

Per

Förmågor

Programmeringsspråk, de man läst på Chalmers (Haskell, Java, lite assembler i Dig- och Dat, lite C (i datasäk.)). Läst C och C++ på gymnasiet. Visual Basic såklart. Informationssystemdriftsasssistent i Lumpen. Satt och administrerade arméns datorer och nätverk i fält. Gick en kurs i TCP/IP i lumpen, en vecka. Windows NT två veckor. Lärde sig lite routing och SQL, men inte så mycket. Har installerat Linux några gånger, jobbar lite med det. Är nätverksadministratör på Guldhedens Studenthem. Arbetsuppgifter på Guldheden att ta hand om den server (web, mail, dns), annars mer "vanlig" nätverksadministration; lägga till nya användare och så. Windows-kunskaper, vanlig användning.

Vilja

Vill lära sig mer konkret hur man gör intrång i datasystem, av intresseskäl. Samt att kunna förhindra attackerare, att kunna sätta upp ett system som görs säkert mot intrång.

Erik

Förmågor

Programmeringsspråk: C, C++, Java, Haskell, Perl, Visual Basic, Basic. Samt webtjaffs (HTML, PHP, CSS). Bash-scripting. Assembler enl. Dig-och-Dat-kursen. Kört Linux primärt i tre år, både som server och workstation. Gillar att leka med säkerheten, leka paranoid. Lekt lite med FreeBSD, även lite OpenBSD men det är rätt liten skillnad. Väldigt lite, körde det på en Alpha. Läst alla Chalmers säkerhetskurser, datasäkerhet, krypto och nätverkssäkerhet. Gått igenom TCP/IP, syn/ack-probing för att ta reda på vad det finns för datorer bakom firewall t.ex.

Vilja

Mest intresserad av att lära sig hur man döljer sina spår, samt försvaret på det hur man då från andra sidan kan upptäcka att man har blivit hackad. Öka kunskapen i säkerhet; ex. vad skall man logga, vad skall man leta efter. Om man själv skall hacka, vad för fällor finns för, vad kan loggas, vad skall man se upp med.

Marko

Förmågor

Programmeringsspråk: Java, Haskell, C men glömt en del. Kan HTML och JS.
Linux/UNIX-erfarenheter: Använt det i skolan. Kan använda terminaler i arbetet.
Windows-erfarenheter: Kan använda det som användare.
Läser datasäkerhetskursen nu, där det mesta är nytt.

Vilja

Få inblick hur det fungerar, säkerhet överhuvudtaget. Hur man tar sig in och hur man skyddar sig. Tycker det är roligt med social engineering.

Fredrik

Förmågor

Programmeringsspråk: Java, Haskell. Sett assembler i Dig-och-dat, och C i datasäkerhetslabben. Känner sig inte ha spetskompetens inom något område. Är ganska trevlig dock. Tycker sig kunna ganska mycket om allt med datasäkerhet, en bred

men grund kunskap. Tycker det är väldigt intressant. Linux/UNIX-erfarenheter: Vad som har lärts i skolan. Gillar Windows mest, som användare.

Vilja

Vill lära sig det mesta, är detta området han vill inrikta sig på i utbildningen. Känner det som att det är detta han vill jobba med i framtiden.

Yildirim

Förmågor

Programmeringsspråk: Java, Haskell. Har programmerat C/C++ i gymnasiet, och även Chalmers. Lite bash-scripting. Plus dig-och-dat assemblern. Känner sig ha bra kunskaper inom Linux. Har testat remote exploit saker, och kan lokala exploits. Läser datasäkerhet nu, har läst nätverkssäkerhet, Unix internt, och datakommunikation. Har datasäkerhet och datakommunikation som inriktning. Windows, kan det mesta användargrejerna. Testat Server och försökt hacka det remote. Modifierade en annan upphovsmans kod (C-kod). Fanns one command-exploits som testades också. Kan olika metoder för att få information om system och dess svagheter: Nmap, Netcraft, Nessus, Retina security scanner, Languard, Nikto.

Vilja

Liknande viljor med Erik, hur man lär sig att dölja sina spår. Även hur man döljer sina spår. Vill även lära sig mer scripting-språk. Känner mer för att lära sig Perl, hellre än bash pga att han kan det (Perl) sämre och det känns viktigare.

Millnert

Förmågor

Programmeringspråk: Chalmers, Java, Haskell, (försumbart MPD och sett lite Erlang). C i flera kurser, Unix Internt och datasäkerhet, väldigt lite hemma.
Linux/UNIX-erfarenheter: Installera första Linux-datorn 2002 när jag flytta till Hemmet. Hade en FreeBSD '99 bakom WinGate. Har 10-talet linuxdatorer idag. Nätverksadministratör på Chalmers Studentbostäder, Linux. Routing (med iptables), firewalls, systemutveckling i Java, XML-RPC plus lite Cisco IOS. Ej certifierad dock, men kan litegrann basic switch-administration. Läser datasäkerhet nu, läst Unix Internt och Avancerad Internetteknologi (applicerbarhet). Windows, sett lite Windows NT, lite Novell och kört en Advanced Server 2000 ett tag. Inga större kunskaper om server-sidan dock.

Vilja

Vill få mer kunskap om system-säkerhet, ur MEDIC-synvinkel. Dvs hur man skyddar så stora och komplexa system på bra sätt. Även utöka kunskaperna i unix/linux-säkerhet då det är det tilltänkta inriktningsområdet och då jag även vill jobba med det i framtiden. Är även projektledare och vill kanske mest av allt skaffa sig erfarenheter i det och lära sig det bra.

Bilaga I

PXE-boot

Live-CD som bootar från nätverket

Installera dhcpd[8] (config: Bilaga K) och tftpd[21] (config: Bilaga L). Kopiera över "gentoo" och "gentoo.igz" från Gentoo live-cdn[7] till tftproot, grub-configen till tftproot/pxelinux.cfg/default. Kopiera även pxelinux.0 från pxe-linux[19] till tftproot-katalogen.
Sen är det bara att starta tftpd och dhcpd för att boota datorn via nätverket.

Bilaga J

Nessus sårbarhetsscanner

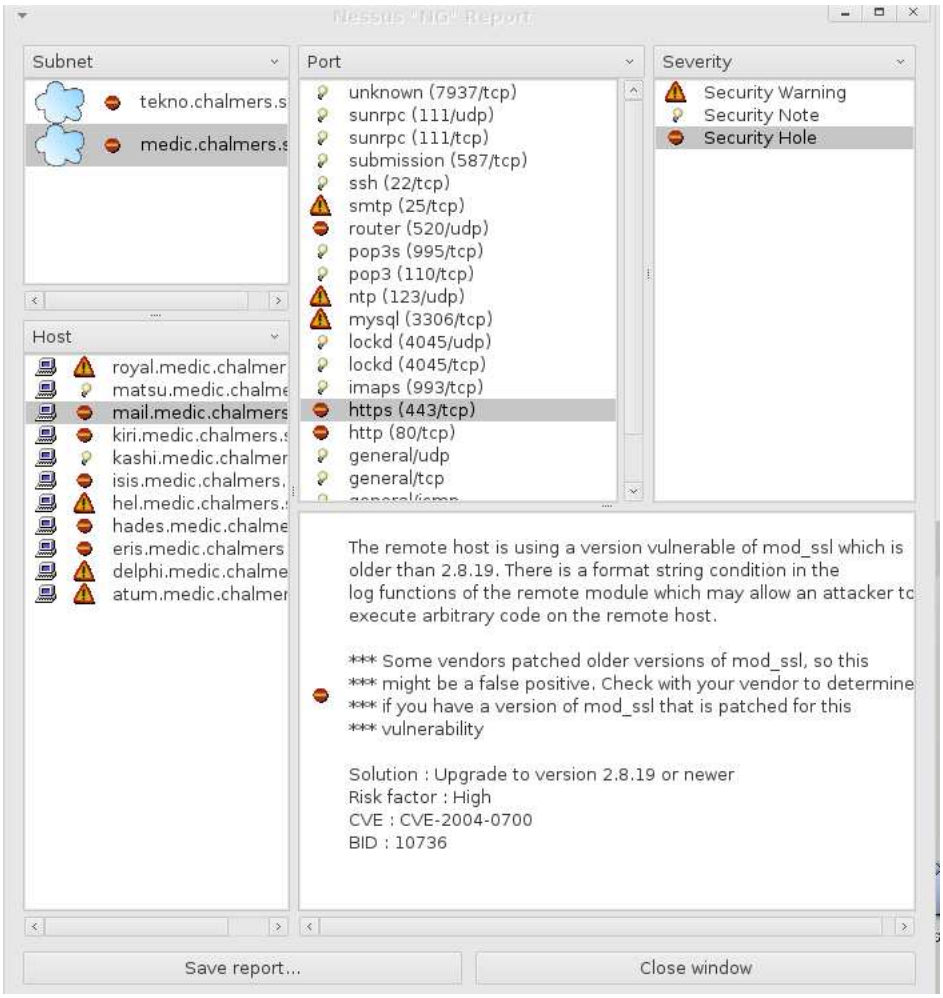
Nessus[11] är en sårbarhetsskanner som tar reda på alla tjänster som körs på den maskin som man scannar, och tar reda på versionsnummer som den den jämnför i sin databas och tar reda på ifall den är sårbar. Efter ha scannat klart så tar den fram en rapport över alla tjänsterna och dess möjliga sårbarheter. Nessus finns för Mac Os, Windows, Linux, BSD, och även Solaris, och dess databas innehåller sårbarheter för alla operativsystem och tjänster. Den kollar bara inte versionnummer utan har även avancerade funktioner för att testa olika svagheter i olika tjänster. Ett exempel är att den testar olika SQL-Injection-kommandon och rapporterar vilka kommandon som är effektiva. En sårbarhetsscanner kan spara mycket tid för en sårbarhetstestare men ska inte helt och hållet litas på. Man ska gärna använda andra program och även manuellt kolla sårbarbeter med hjälp av nessus och säkerhetshemsidor. Att enbart lita på nessusrapport kan ge en känsla av falsk säkerhet.

Skanning

Bilden ovanför visar skanning av några viktiga medic- och teknodatorer. En av fördelarna med nessus är att det kan skanna flera datorer parallellt. Denna skan tog ca 5-10 minuter.

Rapport

Rapporten som skapas efter scanning är klar.



Som ni ser på rapporten så har ett par sårbarheter blivit funna, men har inte blivit testade. Att testa svagheterna krävde mycket kunskap om den tjänst, och dessutom så fanns det inga färdigskrivna exploits att använda. Tiden som skulle ha tagit för att bekanta sig med tjänsten och ta reda på sårbarheterna i mer detailj, och sedan skriva en exploit var för orimlig, och därför blev ingen del av vårat arbete.

Bilaga K

DHCPd-config

```
# dhcpd.conf
#
# Sample configuration file for ISC dhcpd
#

# option definitions common to all supported networks...
option domain-name "example.org";
option domain-name-servers ns1.example.org, ns2.example.org;
ddns-update-style ad-hoc;

default-lease-time 600;
max-lease-time 7200;

# If this DHCP server is the official DHCP server for the local
# network, the authoritative directive should be uncommented.
#authoritative;

# Use this to send dhcp log messages to a different log file (you also
# have to hack syslog.conf to complete the redirection).
log-facility local7;

# No service will be given on this subnet, but declaring it helps the
# DHCP server to understand the network topology.

subnet 192.168.0.0 netmask 255.255.255.0 {
        option routers 192.168.0.1;
        option broadcast-address 192.168.0.255;

        next-server 192.168.0.1;
        filename "pxelinux.0";

        #guest computers
         pool {
```

```
                range 192.168.0.128 192.168.0.232;
                allow unknown clients;
        }

}
```

Bilaga L

tftpd-config

```
# /etc/init.d/in.tftpd

# Path to server files from
INTFTPD_PATH="/tftproot"

# For more options, see tftpd(8)
INTFTPD_OPTS="-s ${INTFTPD_PATH}"
```

Bilaga M

setuid

Kör som root.

```
#include <unistd.h>
#include <sys/fsuid.h>
#include <stdio.h>

int main(int argc, char *argv[]) {
        int uid = atoi(argv[1]);
        int gid = atoi(argv[2]);

        int retval = setuid(uid);
        printf("uid now: %d, was: %d...\n", uid, retval);
        retval = setgid(gid);
        printf("gid now: %d, was: %d, launching shell...\n",
                    gid, retval);

        char *args[2];
        args[0]="/bin/bash";
        args[1]=NULL;
        execvp(*args, args);

        return 0;
}
```

Kompilera på den lokala maskinen med:

```
gcc -o <name> <c-file>
```

Kör igång med:

```
./<name> [uid] [gid]
```

Bilaga N

Brygga

En brygga är när man sätter ihop flera nätverkskort i en dator till en enda, så att den ungefär fungerar som en switch. Med hjälp av denna brygga kan man avlyssna trafik i ett nätverk genom att sätta upp en brygga mellan de punkterna man vill avlyssna. Sen kommer trafiken gå obemärkt genom bryggan och med tex ethereal[5] avlyssnas.
Så här gör du för att sätta upp en brygga i Linux:

ifconfig eth0 0.0.0.0 promisc up
ifconfig eth1 0.0.0.0 promisc up
brctl addbr br0
brctl addif br0 eth0
brctl addif br0 eth1
ifconfig br0 0.0.0.0 up

Bilaga O

Mötesprotokoll 2006-03-27

Närvarande

- Martin
- Yildirim
- Millnert

Frånvarande

- Övriga

Mötet

Mötet bestämmer att Erik skall hålla hemsidan uppdaterad med aktuell information, t.ex. den publika delen.
Jämför kurshemsidan http://www.ce.chalmers.se/edu/year/2004/course/EDA335/ . Detta eftersom det är han som kan wikin bäst och kör den på sin dator hemma.

Status Windowshackning

Har kommit åt BIOS via PXEboot (nätverk). Sen "dog" (resettades) BIOS på ena Windowsdatorn efter en massa reboots. (INTRESSANT!) Sen kunde man boota en bootcd, och läsa Windows SAMfil och systemfiler. Först dumpar man SYSKEY, sen använder man det för att dumpa LM och NTLMhashen från SAM. BKHive och pwdump 2 (plockar ur alla användera och adminstratörer.) "Sequels are always better." Skicka detta till en rainbowtablecracker och sen har man lösenordet på "femton minuter en kvart". Vad kom man åt sen då? Allt möjligt, behöver utforskas mer utförligt. Eventuella remotesystem är ännu otestade då datorn försökte skicka en massa loggar (satt på NOMAD), kan lösas med en brygga som brandväggar t.e.x.

Status Linuxarna

PXEboot, bootcd, montera diskar, kopiera av systemet. Chroot in till det monterade systemet, passwd, nytt rootpass, reboot utan bootcd, voila. Som root kommer

man åt.. mycket. Millnert och Yildirim skrev ett litet changeuid program för root, som gör setuid & setgid. Tar UID och GID som parametrar och man kan då bli vem som helst, t.ex Viktor Fougstedt, och bläddra runt bland deras hemkataloger och hitta sshnycklar och dylikt. Man kan komma åt allt på de NFSexports som finns, webservrarnas (www.dtek/etek/mtek etc, ej webmail, troligen). När vi hade möte med MEDIC i höstas nämnde dom att det fanns en enkel sak, och sa att vi kanske redan hade hittat den. Millnert föredrar DNSattack ("cache poisoning") och ARPpoisoning.

Remote

Per har scannat och satt upp listor över olika tjänser och versioner och letat exploits till dessa. Det arbetet får fortsätta.

Bilaga P

Kontroll av larm

Koll av larm i NC och EDIT.

Ideläran:

Sladden kommer ur larmdosan i väggen, går igenom båda datorerna och skärmarma och landar sen i en tom ände, dosan lyer rött:

grupprum 4,6,7,16 olarmat.

Läget ser nominellt ut, alla kablar sitter i vad vi kan se, men dosan lyser rött:

grupprum 10,11,14,13.

Läget ser nominellt ut, båda kontakterna sitter i dosan och det lyser inte rött inuti dosan:

grupprum 5,15,8,9,12.

Den öppna salen på nedervåningen:

De flesta dosor ser ok ut men det finns en vars slinga är bruten och dosan lyser rött.

EDIT:

32xx (ovanför basen, har röda cat-kablar för larm)

3507 har cat-kabel för larm.

3358 har röda cat-kablar för larm, men det finns minst en olarmad dator. saknar larmdosa helt.

3354 tre av dosorna lyser (3 av slingorna, 3 av raderna), en enda lyser inte.

5352 2 av 3 slingor har dosa om lyser rött

5355 1 av 6 lyser inte rött, 5 gör det!

6217 2 av 3 av slingornas dosor lyser rött.

6225 (ej datorer, bara skärmar + tgb/mus i salen) ej inkopplat något larm någonstans, och här är det TP-liknande kabel för larm.

6360, 6355 inga larm alls, inga datorer. som 6225, frånsett att det inte ens är draget något larm.

6352 datorer + skärmar, larmat med slingor, 2 av 3 dosor lyser rött.

Bilaga Q

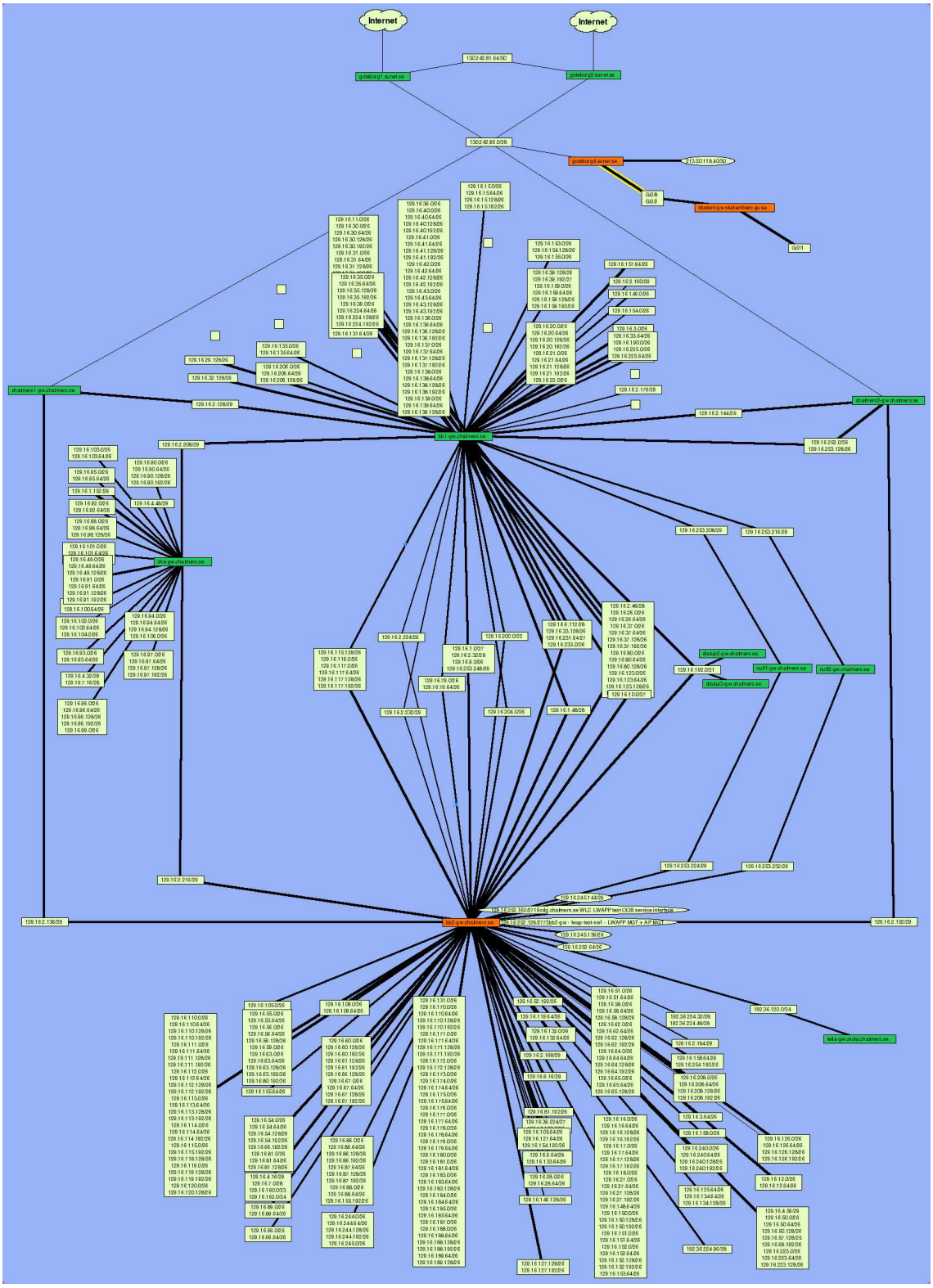
Scan av hades

```
# Nmap 4.01 scan initiated Sat Apr 29 00:33:55 2006 as:
nmap -sV -O --host_timeout 120000 -oA 129.16.30.194 129.16.30.194
Interesting ports on hades.medic.chalmers.se (129.16.30.194):
(The 1662 ports scanned but not shown below are in state: closed)
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          SunSSH 1.1 (protocol 2.0)
25/tcp    open  smtp         Postfix smtpd
111/tcp   open  rpcbind       2-4 (rpc #100000)
139/tcp   open  netbios-ssn  Samba smbd 3.X (workgroup: MEDIC)
2049/tcp  open  nfs           2-3 (rpc #100003)
4045/tcp  open  nlockmgr      1-4 (rpc #100021)
7100/tcp  open  font-service Sun Solaris fs.auto
7937/tcp  open  nsrexec       1 (rpc #390113)
7938/tcp  open  rpcbind       2 (rpc #100000)
32771/tcp open  ypserv        1-2 (rpc #100004)
MAC Address: 00:03:BA:08:D5:C3 (Sun Microsystems)
Device type: general purpose
Running: Sun Solaris 9|10
OS details: Sun Solaris 9 or 10
Uptime 34.331 days (since Sat Mar 25 15:39:13 2006)
Service Info: Host:  hades.medic.chalmers.se; OS: Solaris

# Nmap run completed at Sat Apr 29 00:35:09 2006 -- 1 IP address
(1 host up) scanned in 74.130 seconds
```

Bilaga R

Chalmers nättoppologi


Internet
Internet
goteborg1.sunet.se
goteborg2.sunet.se
goteborg2.sunet.se
student-gw.studentbam.gu.se
chalmers1-gw.chalmers.se
chalmers2-gw.chalmers.se
M1-gw.chalmers.se
chs-gw.chalmers.se
kd-gw.chalmers.se
data2-gw.chalmers.se
data3-gw.chalmers.se
null1-gw.chalmers.se
null2-gw.chalmers.se

Bilaga S

Ordlista

- ARP-storms
  Förfrågningar efter IP-adress till en mängd datorer.

- Brygga
  Dator som (fysiskt) befinner sig mellan två datorer och lyssnar på all trafik mellan dessa (utan att påverka trafiken).

- DNS
  System för namnuppslagning av datorer

- Dos-attack, Denial of Service-attack
  En attack som på något sätt hindrar auktoriserade användare från att använda någon tjänst som de ska kunna använda.

- Exploit
  Program/kodsnutt som utnyttjar en svaghet i ett system för att t.ex. ge otillbörlig tillgång till ett system.

- Piggy-backing
  I vårt fall: Att få otillbörlig tillgång till en datasal genom att följa efter en person som är på väg in.

- Roota, Få Root
  Skaffa sig root-access på en dator.

- Rootkit
  Verktyg som gör att hackaren kan ha kvar access och uttnyttja systemet då han kommit in.

- Scan, Scanna
  Ett sätt att undersöka en dator utifrån genom att skicka olika former av data till den och utifrån dess svar dra vissa slutsatser om den.

- Script-kiddies
  Oerfarna datoranvändare som utnyttjar program som andra personer skrivit för att göra intrång.

- SSH
  Protokoll för krypterade anslutningar mot andra datorer. Programmet ger ett shell till måldatorn för kommandoexekvering.

- Trashing
  Även kallat dumpster diving. Att söka igenom sopor i jakt på information om ett system/personer.

- War-walking (även war-driving, war-flying etc.)
  Att gå runt med en bärbar dator och leta efter trådlösa nätverk där man kan ta sig in.