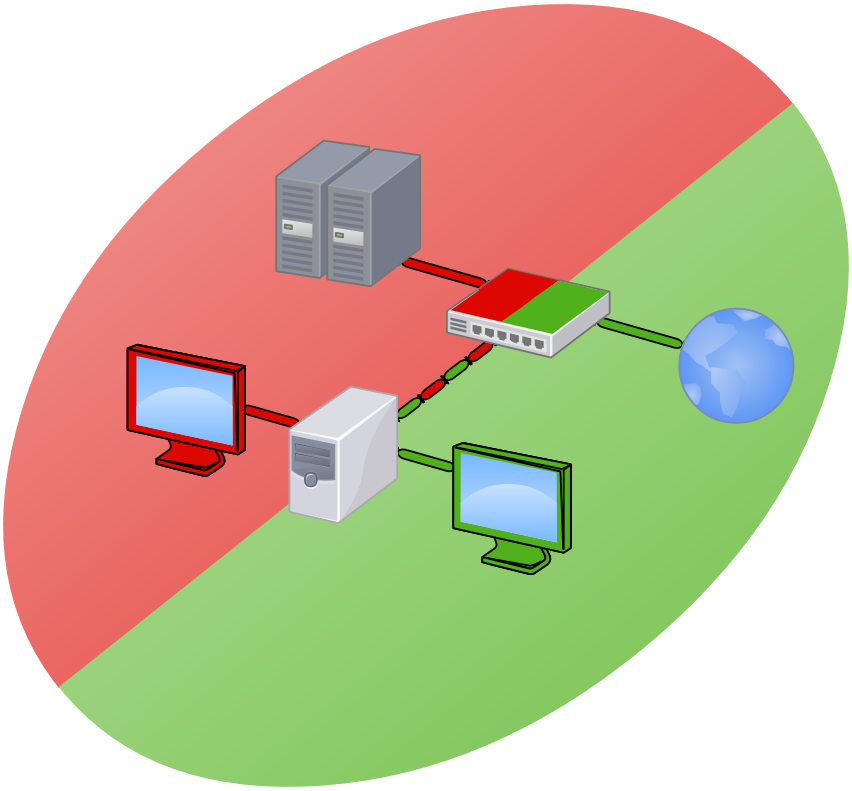
CHALMERS



Dubbla nätverk i samma dator

Blanda rött och grönt

Kandidatarbete inom Data- och Informationsteknik

Jacob Andersson
John Luc
Daniel Olausson
Oscar Holmberg
Erica Löfström
Roger Tedblad

Institutionen för Data- och Informationsteknik

Chalmers Tekniska Högskola
Göteborg, Sverige 2012

Kandidatarbete/rapport nr 2012:29

Dubbla nätverk i samma dator
Blanda rött och grönt
Jacob Andersson, Oscar Holmberg, John Luc, Erica Löfström, Daniel Olausson & Roger Tedblad
Handledare: Pierre Kleberger, examinator: Arne Linde

Institutionen för Data- och Informationsteknik
Chalmers Tekniska Högskola
SE-412 96 Göteborg, 2012
Sverige

Sammanfattning

Många företag och organisationer vill kunna skydda känslig data från obehöriga. En lösning på detta problem är att separera intranät från Internet. Ett sätt att åstadkomma en sådan separering är genom att använda virtuella nätverk (VLAN). I denna rapport undersöks det om säkerheten hos VLAN är tillräckligt hög för att två virtuella nätverk ska kunna ersätta två fysiskt separerade nätverk. För att undvika att använda en dator till vardera nätverk används virtualisering, vilket gör det möjligt att ansluta en fysisk dator till två virtuella nätverk. Miljön bedöms som säker om trafik från det interna nätverket inte är åtkomlig från det externa nätverket med internetåtkomst. Arbetet avgränsades till att endast behandla Linux-baserade operativsystem. Säkerheten utvärderades genom system- och nätverkstestning. Bland annat genomfördes attackerna ARP spoofing, MAC flooding samt manipulerad och dubbel VLAN-tagg. Resultatet visade att det är möjligt att konfigurera en miljö för att säkert ansluta en enda dator till två olika virtuella nätverk. Däremot var säkerheten helt beroende av hur switch och datorer konfigurerades. Det gick dessutom att påverka prestandan i det interna nätverket genom att utföra en MAC flooding-attack från det externa nätverket. Därmed kunde säkerheten i den konfigurerade miljön inte motsvara säkerheten som uppnås med två fysiska datorer kopplade till två olika fysiska nätverk.

Abstract

Many companies and organizations want to protect sensitive data from unauthorized access. One solution to this problem is to separate intranet from the Internet. One way to accomplish such a separation is by using virtual networks (VLAN). This report examines if the security of VLAN is high enough to motivate a replacement of two physically separated networks with two virtual networks. To avoid the usage of one computer for each network, virtualization is utilized to make it possible to connect one single computer to two virtual networks. The environment is considered to be secure if traffic belonging to the internal network is unreachable from the external network with Internet access. The project was limited to cover only Linux-based operating systems. The security was evaluated through system and network testing. Among the executed attacks were ARP spoofing, MAC flooding as well as manipulated and double VLAN tag. The results suggested that it is possible to configure an environment to securely connect one single computer to two different virtual networks. However, the security was entirely dependent on the configuration of the switch and the computers. Furthermore, it was possible to affect the performance of the internal network by executing a MAC flooding attack from the external network. Consequently, the security in the configured environment does not correspond to the security in an environment with two physical computers connected to two different physical networks.

Innehåll

Ordlista

802.1Q	IEEE-standard för VLAN.
ARP	*Address Resolution Protocol*. Ett protokoll som används av nätverksenheter för att associera MAC-adress med IP-adress.
Broadcast	Meddelande som skickas till alla anslutna noder i ett lokalt nätverk.
CAM-tabell	*Content Addressable Memory*. Switchens interna tabell över kända MAC-adresser.
Ethernet	En samling standardiserade metoder för dataöverföring.
FTP	*File Transfer Protocol*. Ett filöverföringsprotokoll.
HTTP	*HyperText Transfer Protocol*. Protokoll som används av webbläsare för att hantera webbrelaterad data.
Hubb	En nätverksenhet som vidarebefordrar mottagna paket till samtliga anslutna enheter, förutom den enhet som stod för utskicket.
ICMP	*Internet Control Message Protocol*. Ett protokoll som används för ping.
IEEE	*Institute of Electrical and Electronics Engineers*. En sammanslutning ingenjörer som publicerar standarder inom data- och elektroteknik.
IP	*Internet Protocol*. Kommunikationsprotokoll som används för överföring av information över Internet.
IP-adress	Mjukvaruadress som används för adressering i nätverk.
ISO	*International Organization for Standardization*. Ett internationellt standardiseringsorgan.
LAN	*Local area network*. Lokalt nätverk.
MAC-adress	*Media Access Control*. Statisk hårdvaruadress som används av nätverksenheter för adressering.
Nätmask	Beskriver hur stor del av en IP-adress som bestämmer nättillhörighet.
RAM	*Random Access Memory*. Arbetsminne i dator.
Repository	Server för lagring och åtkomst till data. Ofta handlar det om versionshanterad mjukvara.
SNMP	*Simple Network Management Protocol*. Ett protokoll för att övervaka nätverksenheter såsom switchar, routrar och servrar.
Subnät	Uppdelning av ett nätverk genom att välja nätmask.
TCP	*Transmission Control Protocol*. Protokoll för pålitlig överföring av data i nätverk.
TCP SYN	Första steget i en trestegshandskakning för att upprätta en TCP-anslutning mellan två enheter.
VLAN	*Virtual Local Area Network*. Virtuellt LAN, även kallat virtuellt nätverk.
UDP	*User Datagram Protocol*. Protokoll för överföring av data i nätverk.
Wi-Fi	*Wireless fidelity*. Olika former av trådlös överföring av information.

1 Inledning

1.1 Bakgrund

Många företag och organisationer behöver skydda känslig intern information samtidigt som de behöver vara uppkopplade mot Internet. För att denna information inte ska bli tillgänglig utifrån av obehöriga efterfrågas ibland någon form av separering mellan intern och extern datakommunikation. Ett sätt att åstadkomma en sådan uppdelning är genom att arbeta med två fysiska datorer, där den ena är kopplad till ett säkert internt nätverk och den andra är kopplad till ett osäkert externt nätverk med internetåtkomst. Uppdelningen möjliggör optimal säkerhet eftersom det inte finns någon fysisk koppling mellan nätverken. Lösningen innebär dock att en dubbel uppsättning hårdvara används, samtidigt som lösningen varken är kostnadseffektiv eller administratörsvänlig [1, 2].

Ett mer kostnadseffektivt och underhållsvänligt alternativ är att logiskt separera de två nätverken med hjälp av virtuella nätverk (VLAN) [1, 2]. Med hjälp av virtualisering kan man dessutom ersätta två datorer med en enda dator, som både kan kopplas upp mot ett internt och ett externt nätverk. Att kombinera VLAN och virtualisering gör det alltså möjligt att använda ett säkert internt nätverk och ett externt nätverk med internetåtkomst på samma dator.

1.2 Syfte

Syftet med detta arbete är att med hjälp av virtualisering och VLAN konfigurera och säkerhetsvalidera en miljö för att säkert kunna ansluta en fysisk dator till två logiskt separerade nätverk, intranät och Internet. Målet är att undersöka om denna miljö, bestående av en Linux-dator och en VLAN-kompatibel switch, kan motsvara säkerheten som uppnås med två fysiska datorer kopplade till två olika fysiskt separerade nätverk.

1.3 Problembeskrivning

Problemet var att undersöka om man med hjälp av VLAN och virtualisering, på ett säkert sätt, kunde använda endast en dator för att koppla upp sig mot två olika nätverk samtidigt. VLAN skulle användas för att ersätta två fysiska nätverk med ett internt, så kallat rött nätverk, och ett externt, så kallat grönt nätverk, med internetåtkomst. Vidare skulle virtualisering användas för att ersätta två datorer med en dator. Denna dator skulle kopplas upp mot de två olika nätverken samtidigt. Problemet kunde brytas ner i tre delproblem: konfiguration av miljön, systemtestning och nätverkstestning.

Konfiguration av miljön gick ut på att konfigurera datorer och switch. Varje dator skulle kopplas upp mot två logiskt separerade nätverk med hjälp av VLAN och virtualisering.

Systemtestningen syftade till att undersöka säkerheten i den virtualiseringslösning som skulle möjliggöra användning av två operativsystem på en enda dator. De krav som ställdes på virtualiseringslösningen var att ingen annan interaktion än klippa och klistra text skulle vara möjlig mellan de två operativsystemen.

Nätverkstestningen gick ut på att undersöka säkerheten i det VLAN-baserade nätverk som skulle konfigureras. Mer specifikt skulle det undersökas om den logiska separeringen av två virtuella nätverk kunde brytas på något sätt.

1.4 Avgränsningar

Miljön bestod endast av två datorer och en switch. Samma datorer och samma switch användes i samtliga tester, någon annan hårdvara testades ej. Endast ett specifikt Linux-baserat operativsystem och en virtualiseringsmjukvara testades. Operativsystemet och virtualiseringsmjukvaran i sig antogs vara säkra, och inga tester rörande säkerheten i dessa genomfördes. Det enda som undersöktes i virtualiseringsmjukvaran var inbyggd funktionalitet som möjliggjorde kommunikation mellan operativsystem.

Fyra specifika uppsättningar testades, nämligen två olika virtualiseringslösningar i kombination med två olika switchkonfigurationer. Inga andra möjliga virtualiseringslösningar eller switchkonfigurationer utöver dessa behandlades. All kommunikation var trådbunden, inga trådlösa alternativ togs i beaktande.

För att hantera VLAN användes standardprotokollet IEEE 802.1Q [3]. Endast VLAN-relaterade attacker utfördes för att undersöka säkerheten i det VLAN-baserade nätverket. Andra potentiella säkerhetsbrister i nätverket än dem relaterade till VLAN togs ej i beaktande.

1.5 Metod

Inledningsvis genomfördes en litteraturstudie, för att ge fördjupad kunskap inom områdena virtualisering och VLAN. Litteraturstudien berörde även ämnen som operativsystem och säkerhet. Säkerhetsaspekter studerades inom varje område, främst med fokus på säkerhetsbrister vid användning av VLAN.

Val av virtualiseringsmodell samt specifik virtualiseringsmjukvara gjordes utifrån litteraturstudien. Först valdes en lämplig virtualiseringsmodell, och sedan undersöktes olika virtualiseringsmjukvaror som fanns att tillgå. Valet av virtualiseringsmjukvara baserades främst på de krav som fanns, till exempel säkerhetsaspekter och klippa/klistra-funktion.

Valet av operativsystem genomfördes i samband med att virtualiseringsmjukvara valdes. Det krav som ställdes på operativsystemet var att det skulle vara Linux-baserat. Vid valet av Linux-distribution var det viktigt att stöd för virtuella nätverkskort fanns samt att hanteringen av dessa fungerade på ett tillfredsställande sätt.

Ytterligare en litteraturstudie med fokus på VLAN-säkerhet genomfördes. Litteraturstudien resulterade i en uppsättning attacker som genomfördes efter att miljön hade konfigurerats och drifttestats. De attacker och tester som genomfördes syftade till att undersöka säkerheten i VLAN samt att kontrollera säkerheten i de två olika virtualiseringslösningarna.

1.6 Relaterade arbeten

[4] inleder med frasen "Virtual LANs were created to isolate LANs - but not for the purpose of security". Med detta menas att virtuella LAN inte klassas som säkra, åtminstone inte utan en säker konfiguration av switch och brandvägg. Även [5] nämner flertalet potentiella säkerhetsbrister relaterade till VLAN.

Enligt [6] har virtualisering blivit allt mer attraktivt för företag, eftersom det sparar på IT-resurser, minskar kostnader samt främjar miljövänlig IT. Dock kan nya säkerhetsbrister introduceras i och med införandet av virtualiseringslösningar. Enligt en undersökning som var riktad mot företag svarade 70 % av de tillfrågade att de använde sig av en virtualiserad server [6]. De övriga tillfrågade företagen svarade att de hade svårigheter att implementera virtualiseringslösningar på grund av de säkerhetsbrister som virtualisering kan medföra. I [6] presenteras därför en metod som motverkar säkerhetsbristerna och som inkluderar användandet av VLAN och brandväggar.

Förutom ovan nämnda arbeten inom VLAN och virtualisering har ett examensarbete från 2011 [7] undersökt säkerheten inom virtualisering kombinerat med VLAN på en Windows-plattform.

1.7 Rapportstruktur

Rapporten inleds med kapitel *2 Teoretisk bakgrund* som omfattar virtualisering, nätverkslagren och VLAN. *Teoretisk bakgrund* innehåller nödvändig teori för förståelse av innehållet som presenteras i efterföljande kapitel. Kapitlet är fristående från resten av rapporten.

Den teoretiska bakgrunden efterföljs av kapitel *3 Konfiguration*, *4 Systemtestning* och *5 Nätverkstestning*. *Konfiguration* presenterar hur miljön konfigurerades samt drifttestades. Vidare redogörs det i *Systemtestning* och *Nätverkstestning* för hur olika tester genomfördes och vilka resultat som dessa ledde fram till.

Den avslutande delen i rapporten består av kapitel *6 Resultat*, *7 Diskussion*, *8 Slutsats* och *9 Framtida arbeten*. I denna avslutande del presenteras först resultatet av arbetet i sin helhet. Därefter förs en diskussion baserad på de resultat som uppnåddes. De viktigaste resultaten och diskussionerna lyfts fram och presenteras i *Slutsats*. Slutligen ges det i *Framtida arbeten* förslag på komplementerande arbeten som kan vara av intresse att genomföra.

Sist i rapporten finns hårdvaruspecifikationer och konfigurationsfiler som *Appendix A* respektive *Appendix C*. Samtliga programvaror som använts under arbetet

listas i *Appendix B* med länkar till respektive webbsida.

1.8 Terminologi

Genomgående i rapporten används ordet miljö för att referera till hela uppsättningen av hård- och mjukvara. Här ingår två fysiska datorer (system 1 och system 2), switch, router och kablaget däremellan. Ordet system används för att beskriva en dator med värd- och gästoperativsystem. För att referera till de olika värd- och gästoperativsystemen som körs på ett system används termerna röd och grön dator, beroende på vilket virtuellt nätverk som avses.

Rapportens primära språk är svenska, men engelska termer används i de fall det inte finns en motsvarande svensk allmänt känd och använd översättning. Engelska förkortningar som inte skrivs ut i rapporten förklaras i ordlistan.

2 Teoretisk bakgrund

I detta kapitel beskrivs de grundläggande kunskaperna kring virtualisering och VLAN. Här beskrivs olika virtualiseringsmodeller samt vilka virtualiseringstekniker som finns. Därefter kommer ett kort avsnitt om nätverkslagren, samt ett mer ingående avsnitt om hur VLAN fungerar.

2.1 Virtualisering

Virtualisering innebär att man med hjälp av mjukvara simulerar en miljö som motsvarar en viss typ av hårdvara. På detta sätt kan med hjälp av virtuella maskiner köra flera operativsystem parallellt på samma värddator. Hanteringen av de virtuella maskinerna och uppdelningen av hårdvara sköts av en speciell mjukvara [8]. Man delar ofta in virtualiseringsmjukvaror utifrån olika tekniker som används för att skapa och hantera separata virtuella maskiner. I denna uppdelning ingår *full virtualization*, *paravirtualization*, *operating system virtualization* samt *native virtualization*. Det finns även en annan uppdelning där mjukvarorna istället delas in i tre olika virtualiseringsmodeller. Dessa modeller är: *hypervisor typ 1*, *hypervisor typ 2* och *virtuella containrar*. En hypervisor, även kallad *virtual machine monitor* [9], är en mjukvara som används för att kunna virtualisera hårdvara och hantera kommunikation mellan flera virtuella maskiner. Här beskrivs först de två hypervisor-typerna samt virtuella containrar. Därefter beskrivs den andra kategoriseringen med de fyra virtualiseringsteknikerna.

2.1.1 Virtualiseringsmodeller

Det finns tre starka egenskaper man strävar efter vid användning av virtualisering, nämligen likvärdighet, resurskontroll och prestanda [9]. Med likvärdighet menas att det virtualiserade systemet ska avspegla hårdvaran på ett sådant sätt att det inte vore någon skillnad mot om man faktiskt körde direkt på den verkliga hårdvaran [9]. Det är också viktigt att en hypervisor har full resurskontroll eftersom det ska vara omöjligt för en virtuell maskin att styra över systemresurser utan att gå via hypervisor-lagret [9]. Med egenskapen prestanda syftar man på att alla instruktioner från en virtuell maskin, som inte kan påverka andra virtuella maskiner, ska kunna köras direkt på hårdvaran utan att bli modifierade av en hypervisor [9].

2.1.1.1 Hypervisor typ 1

Hypervisor typ 1 installeras direkt på hårdvara istället för att installeras ovanpå ett värdoperativsystem, och fungerar därmed som ett lager mellan underliggande hårdvara och virtuella maskiner [8]. Detta illustreras i figur 1(a). Alla systemanrop måste gå via hypervisor-lagret innan hårdvaran kan utnyttjas [10]. Detta är väsentligt eftersom hanteringen och uppdelning av hårdvarans resurser till virtuella maskiner måste skötas på ett korrekt och säkert sätt. Alla virtuella maskiner verkar därmed helt separerade eftersom de saknar möjlighet att kommunicera med varandra [10].

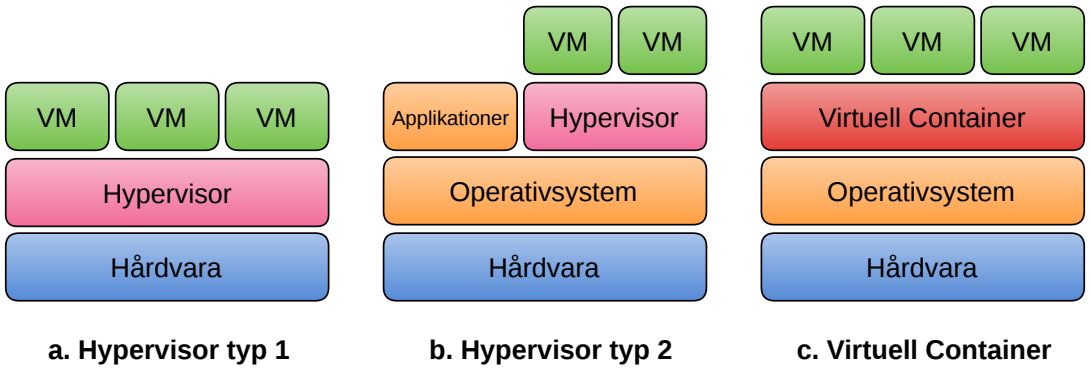
2.1.1.2 Hypervisor typ 2

Den andra virtualiseringsmodellen, hypervisor typ 2, fungerar på ett annorlunda sätt i förhållande till typ 1. Virtualiseringsmjukvaran körs direkt ovanpå ett värdoperativsystem, vilket illustreras i figur 1(b), och emulerar en miljö som tillåter att så kallade gästoperativsystem installeras [8]. Detta innebär att kommunikationen mellan gästoperativsystem och hårdvara, som hanteras av hypervisor-lagret, måste gå via mellanliggande värdoperativsystem. Denna kommunikation medför sämre prestanda i jämförelse med hypervisor typ 1 och kan leda till fler säkerhetsbrister [11]. Största skillnaden är att både värd- och gästoperativsystem potentiellt kan utsättas för attacker [12]. Fördelen med denna virtualiseringsmodell är bland annat att den är lättare att installera och underhålla [10]. Den är dessutom mycket lättare att distribuera eftersom det inte krävs någon omfattande konfiguration för att köra mjukvaran på andra datorer [13].

2.1.1.3 Virtuella containrar

Virtuella containrar bygger på att man isolerar resurser inuti ett redan installerat operativsystem. Detta illustreras i figur 1(c), vilket kan jämföras med hypervisor typ 2 i figur 1(b). Den virtuella containern blir tilldelad sina egna resurser, så som arbetsminne, hårddiskutrymme och processorgaranti [14]. Däremot har en virtuell container oftast ett delat filsystem med värdoperativsystemet, vilket kan uppfattas som både positivt och negativt [14]. Det positiva i jämförelse med hypervisor typ 2 är att virtuella containrar kan få ut bättre prestanda eftersom de i detta fall slipper skapa nya filsystem för alla virtuella maskiner. En stor nackdel kan dock vara att säkerheten i systemet inte blir lika hög eftersom virtuella containrar kan komma åt information tillhörande andra virtuella maskiner i systemet.

Till skillnad från hypervisor typ 1 och typ 2, som skapar en ny virtuell kärna för varje ny virtuell maskin som installeras, så delar virtuella containrar åtkomst till operativsystemkärnan med värdoperativsystemet [14]. Detta innebär att virtuella containrar är begränsade till att köra samma typ av operativsystem, eftersom de är baserade på ett specifikt värdoperativsystem. Sammantaget kan man säga att virtuella containrar erbjuder en flexibel miljö som ger bra prestanda. Dock har man svårt att garantera säkerheten eftersom operativsystemen inte blir lika separerade och isolerade som med hypervisor-modellerna.



Figur 1: *Tre olika typer av virtualiseringsmodeller (a, b och c) och hur dessa är implementerade ovanpå operativsystem respektive hårdvara.*

2.1.2 Virtualiseringstekniker

Utöver kategoriseringen med virtualiseringsmodeller finns det dessutom fyra virtualiseringstekniker som beskriver hur hårdvara simuleras. I en virtualiserad miljö kan vissa instruktioner anses vara kritiska för miljön. Varje teknik anpassar dessa instruktioner på olika sätt för att de ska kunna exekveras felfritt.

2.1.2.1 Full virtualization

I full virtualization simuleras all hårdvara för gästoperativsystem. Operativsystemen är inte medvetna om att de körs i en virtualiserad miljö, och behöver därför inte konfigureras på något speciellt sätt. Full virtualization använder binär översättning (binary translation) [15], vilket innebär att virtualiseringsmjukvaran fångar upp och tar hand om kritiska instruktioner som ett gästoperativsystem försöker exekvera. Instruktionerna byts ut mot säkra instruktioner utan att gästoperativsystemet får kännedom om händelsen. De instruktioner som inte klassas som kritiska exekveras direkt på hårdvaran, vilket bidrar till förbättrad prestanda. Detta resulterar i att ett gästoperativsystem kan köra sina egna applikationer och processer direkt på underliggande hårdvara, vilket fungerar eftersom de virtuella maskinerna är separerade på hårdvarunivå [16]. Denna virtualiseringsteknik ger även ett brett utbud för vilka operativsystem som kan köras på datorn.

2.1.2.2 Paravirtualization

Till skillnad från full virtualization simulerar paravirtualization endast delar av hårdvara för gästoperativsystem [11]. Paravirtualization använder inte binär översättning för att hantera kritiska instruktioner. Istället anpassas gästoperativsystemet för att skicka instruktioner till underliggande hårdvara [15]. Gästoperativsystemet blir därmed medvetet om att det kör på delar av simulerad hårdvara. Denna anpassning ger en del negativa följder för operativsystemet, så som bristande stöd för bakåtkompatibilitet [11]. Dessutom måste gästoperativsystemet konfigureras specifikt för varje enskild dator [11]. Fördelarna med paravirtualization gentemot full virtualization är att tekniken ger en allmänt bättre prestanda och effektivitet [16].

2.1.2.3 Operating system virtualization

Med operating system virtualization fördelas systemresurser på ett sådant sätt att multipla instanser av det underliggande operativsystemet kan köras [15]. På så sätt behöver endast ett operativsystem installeras, underhållas och uppdateras. Alla virtuella maskiner delar åtkomst till filsystem och hårdvara, vilket tenderar att resultera i bättre prestanda och högre användarvänlighet [11]. Däremot blir påföljden sämre isolering och säkerhet jämfört med andra virtualiseringstekniker. Ytterligare nackdelar är att det blir svårt att identifiera ursprunget av resurskrävande processer samt att det är svårt att begränsa enskilda virtuella maskiners resursanvändning [11].

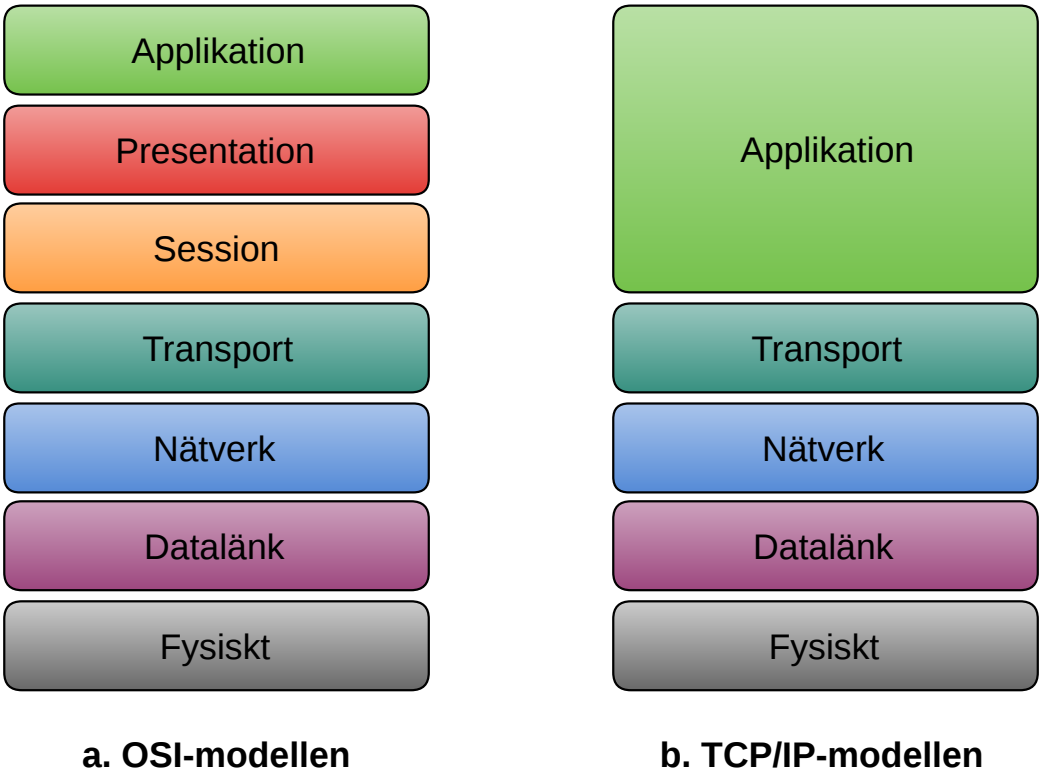
2.1.2.4 Native virtualization

Native virtualization har likt full virtualization möjlighet att simulera all hårdvara eller likt paravirtualization endast delar av hårdvaran. Till skillnad från fullvirtualization och paravirtualization hanterar native virtualization kritiska instruktioner

med hjälp av virtualiseringsstöd i hårdvara [11]. Native virtualization anpassar de kritiska instruktionerna för att de ska fungera felfritt i miljön. Dock kommer fortfarande icke kritiska instruktioner att exekveras direkt av processorn utan att påverkas av hypervisor-lagret. Detta ger i likhet med full virtualization möjlighet för operativsystemets applikationer att utnyttja underliggande hårdvara direkt utan någon modifiering [11]. När native virtualization endast simulerar delar av hårdvara måste gästoperativsystemen anpassas på samma sätt som för paravirtualization.

2.2 Nätverkslagren

Datornätverk består av många olika delar, som tillsammans kan utgöra väldigt komplexa system. Därför har det funnits ett behov av att bryta ner komplexiteten i delar. För att på ett mer överskådligt sätt kunna beskriva och arbeta med datorkommunikation finns därför ett antal olika modeller, som beskriver nätverksarkitektur i termer av lager [17]. Att beskriva arkitekturen med hjälp av olika lager har fungerat som ett sätt att organisera nätverksprotokoll och den hård- och mjukvara som implementerar dessa [17]. Två av de mest kända och tillämpade modellerna är *OSI-modellen* som visas i figur 2(a), och *TCP/IP-modellen* som visas i figur 2(b).



Figur 2: *Konceptuella modeller för datakommunikation. De färgade rutorna representerar olika nätverkslager.*

OSI-modellen, som illustreras i figur 2(a), beskriver nätverk med hjälp av sju olika lager och har sedan slutet av 1970-talet varit en etablerad standardmodell [18]. Modellen standardiserades av *ISO* och efter stegvis utveckling beskrivs den nu i

ISO/IEC 7498-1:1994 [19].

TCP/IP-modellen är en något förenklad modell som saknar de två lagren *Session* och *Presentation*, vilket framgår av figur 2(b). Dessa lager kan istället betraktas som delar av lagret *Applikation*, och det är upp till applikationsutvecklaren att implementera motsvarande funktionalitet [17].

Inom ramen för detta projekt är det fullt tillräckligt att använda TCP/IP-modellen som beskrivning av datakommunikation över Internet, eftersom de övre lagren ej kommer att beröras.

2.2.1 TCP/IP-modellen

Lagerindelningen i TCP/IP-modellen utgörs av de fem lagren *Fysiskt, Datalänk, Nätverk, Transport* och *Applikation* [17]. Härnäst följer en kort beskrivning av respektive nätverkslager baserat på [17, 20]. Varje lagers funktionalitet bygger på de tjänster som tillhandahålls av det underliggande lagret. På motsvarande sätt använder sig varje lager av tjänster från överliggande lager för att kunna utföra sitt arbete.

2.2.1.1 Lager 1: Fysiskt

Det fysiska lagret arbetar nära hårdvaran och ansvarar för den rent fysiska överföringen av data mellan två noder, i form av enskilda bitar. Olika protokoll används för denna överföring beroende på vilken typ av överföringsmedium som används.

2.2.1.2 Lager 2: Datalänk

Datalänklagret tillhandahåller pålitlig överföring av så kallade ramar mellan två sammankopplade noder, till exempel mellan en switch och nätverkskortet i en dator. Datalänklagret, som implementeras av datorer, routrar och switchar, använder sig av hårdvaruadresser (MAC-addresser) för adressering. Exempel på protokoll som implementeras av datalänklagret är *Ethernet* och olika implementationer av *Wi-Fi*.

2.2.1.3 Lager 3: Nätverk

Nätverkslagret syftar till att, med hjälp av protokollet *IP*, erbjuda kommunikation mellan datorer. Nätverkslagret implementeras av datorer och routrar, och de datapaket som skickas kallas datagram. Kommunikationen mellan datorer är på denna nivå förbindelselös, vilket innebär att inga leveransgarantier kan ges.

2.2.1.4 Lager 4: Transport

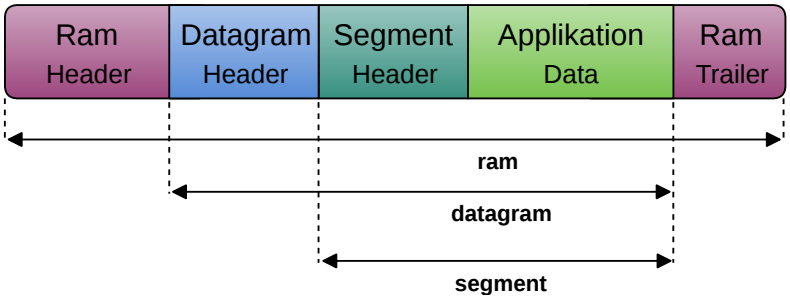
Transportlagrets uppgift är att bygga vidare på nätverkslagrets tjänster för att tillhandahålla uppkopplingsbaserad kommunikation mellan applikationsprocesser som körs på olika datorer. Transportlagret är integrerat i datorer och paketen som skickas benämns segment. Protokoll som används av transportlagret för att erbjuda sådana tjänster på Internet är *TCP* och *UDP*. TCP erbjuder till skillnad från UDP pålitlig överföring av data mellan två datorer i ett nätverk.

2.2.1.5 Lager 5: Applikation

På applikationslagret körs olika internetapplikationer som till exempel e-post och internettelefoni. Olika typer av applikationer nyttjar olika tjänster från transportlagret, beroende på vilken funktionalitet som eftersträvas. Applikationsprocesser kommunicerar med varandra med hjälp av applikationsspecifika protokoll som till exempel *HTTP*.

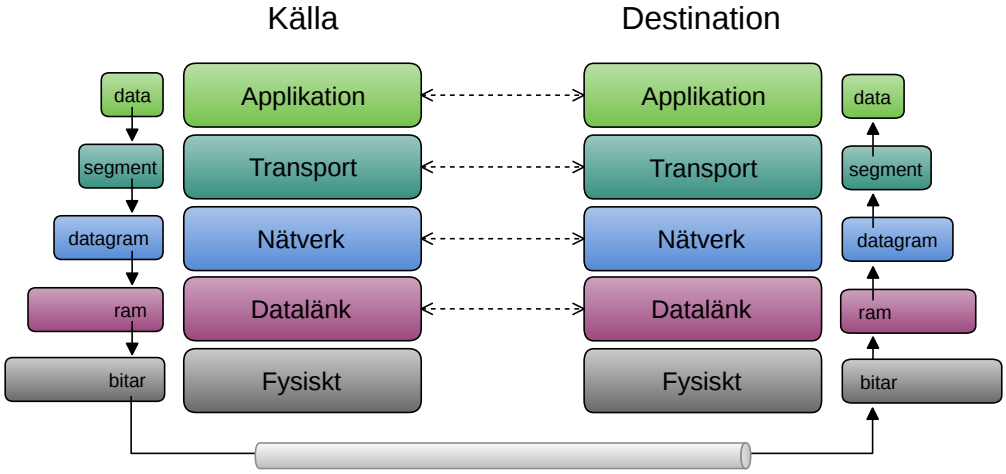
2.2.2 Inkapsling enligt TCP/IP-modellen

Varje lager kapslar in datapaket från föregående lager genom att lägga till en så kallad header och ibland även en trailer, se figur 3. Dessa innehåller information som används av motsvarande lager på mottagarsidan [17].



Figur 3: *Inkapsling av data enligt TCP/IP-modellen.*

Figur 3 visar en ram som är färdig att behandlas av det fysiska lagret, vars uppgift är att dela upp ramen i bitar för att skicka datan genom ett överföringsmedium. Hur bitarna överförs samt hanteras av mottagaren visas i figur 4.

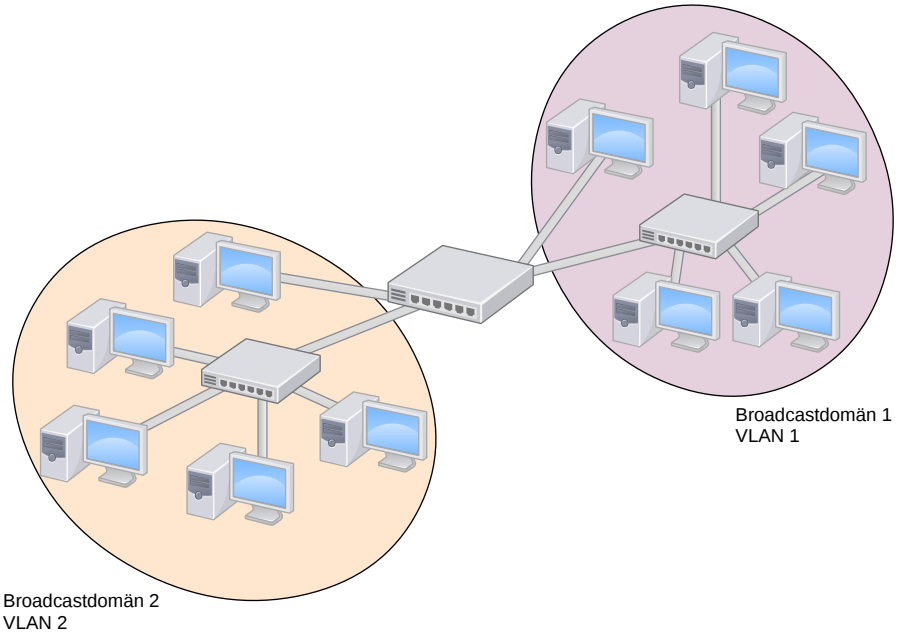


Figur 4: *Data kapslas in och skickas genom ett överföringsmedium till en mottagare som i sin tur levererar datan till en applikationsprocess.*

När applikationslagret uppfattar att data ska skickas påbörjas inkapslingsprocessen som illustreras i figur 4. Datan passerar nedåt i TCP/IP-modellen tills den når det fysiska lagret, som i sin tur skickar bitarna till destinationen. Den mottagande datorn sätter ihop bitarna till en ram som packas upp stegvis för att till slut levereras till en applikationsprocess.

2.3 VLAN

VLAN är en teknik för att dela upp ett lokalt nätverk i flera mindre nätverk [21]. Tekniken implementeras i lager 2 och gör att ett enda nätverk kan utgöras av en samling mindre virtuella nätverk, se figur 5. Fördelen med denna virtuella nätverksuppdelning är att mängden använd hårdvara kan minskas. De virtuella nätverken skapas och skiljs åt enbart med hjälp av mjukvara i switchen, istället för att skapas och skiljas åt genom fysisk separering.



Figur 5: *Ett nätverk bestående av två virtuella nätverk som utgör varsin broadcastdomän.*

Varje virtuellt nätverk är logiskt avskärmat från de övriga virtuella nätverken. Detta medför att alla broadcast-meddelanden som skickas från en enhet endast når de enheter som är anslutna till ett specifikt virtuellt nätverk [22], vilket illustreras i figur 5. När VLAN inte används mottager samtliga enheter i ett nätverk broadcast-meddelanden, vilket kan skapa hög belastning på nätverket.

När tekniken VLAN var alldeles ny och skulle börja användas trodde nätverksadministratörer att denna teknik skulle medföra att man inte längre behövde använda sig av routrar [22]. Detta visade sig dock vara ett felaktigt antagande [22]. Virtuella nätverk kan underlätta vissa uppgifter som att göra enkla accesslistor, men tekniken kan inte ersätta funktionaliteten hos en router [22].

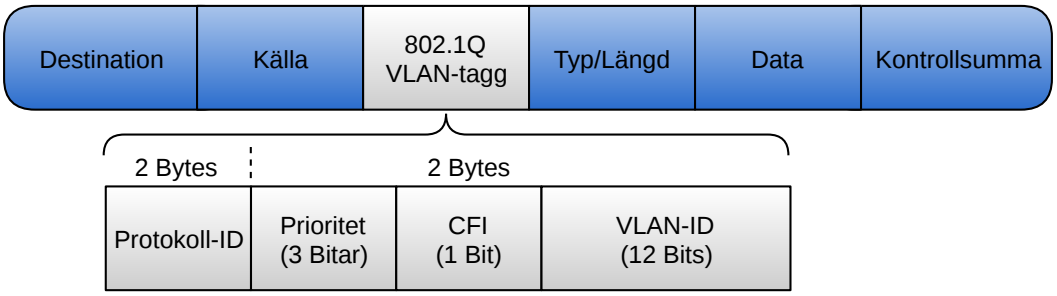
2.3.1 Uppbyggnad

Virtuella nätverk kan konstrueras på olika sätt. En konstruktion är att statiskt konfigurera virtuella nätverk i switchen [22]. Switchens portar delas upp i olika grupperingar, vilka bildar egna separata nätverk [21]. Ett annat sätt är att dynamiskt, med hjälp av speciell mjukvara, bestämma VLAN-tillhörighet på portarna med datorernas MAC-adresser [22]. Dynamiska VLAN använder sig av en databas

som varje inkopplad enhet gör förfrågan mot, och kan på detta sätt tilldelas rätt VLAN-tillhörighet.

2.3.2 Protokollet 802.1Q

När det gäller implementationen av virtuella nätverk fanns det tidigare flera metoder för hur detta kunde göras [22]. Innan en fast standard kom hade olika tillverkare sina egna metoder för att hantera VLAN [22]. När sedan IEEE släppte sin standardisering för VLAN enligt 802.1Q blev den snabbt förstahandsvalet för att implementera virtuella nätverk [22]. Uppbyggnaden av en 802.1Q-tagg samt dess placering i ett paket visas i figur 6.



Figur 6: *Placering av VLAN-tagg i ett paket samt beskrivning av taggens uppbyggnad.*

En VLAN-tagg innehåller bland annat ett VLAN-ID (enligt figur 6), som bestämmer vilket VLAN paketet tillhör [21]. Med hjälp av taggens VLAN-ID kan en switch avgöra vilket VLAN ett inkommande paket tillhör, och kan på så sätt se till att paket endast skickas ut på portar tillhörande samma VLAN [22]. Det är dock även vanligt att paket skickas utan VLAN-tagg, det vill säga otaggade. Källan som paketen skickas från tillhör då fortfarande ett VLAN. Det enda som skiljer detta fall mot det taggade är att paketen som skickas inte förses med ett VLAN-ID och ger därför inte någon information om vilket VLAN källan är ansluten till [22].

För att switchen ska kunna avgöra vilket VLAN ett paket utan VLAN-ID tillhör har den en intern tabell där den kopplar samman MAC-adresser med olika VLAN. I denna så kallade CAM-tabell sparar switchen även information om vilka portar de olika MAC-adresserna befinner sig på [22]. Eftersom switchen känner till varje ports VLAN-tillhörighet kan den genom att titta på källan hos ett paket sammankoppla porten som tar emot paketet och dess VLAN-tillhörighet med källans MAC-adress. Genom att lagra denna information ges switchen möjlighet att enbart skicka ut information på de portar som tillhör samma VLAN, till skillnad från en hubb som skickar ut samma information till alla portar [23]. När ett otaggat paket anländer till en port kan därför switchen avgöra vilket VLAN paketet ska skickas ut inom. Om paketets källa inte finns i tabellen lagras den tillsammans med den mottagande portens VLAN-tillhörighet. Sedan skickas paketet ut till det VLAN som porten är konfigurerad för.

2.3.3 Trunking

Om flera switchar kopplas samman, eller om en dator ska anslutas till flera olika VLAN, kan en trunk-port användas [22]. Med hjälp av en trunk-port kan flera virtuella nätverk överföra information genom samma förbindelse.

Termen *trunking* kan också betyda *link aggregation*. När Hewlett-Packard (HP) använder begreppet trunking syftar de på link aggregation, vilket är en metod för att öka bandbredden genom att slå ihop flera överföringsmedium. När Cisco däremot nämner trunking avses *VLAN multiplexing*, vilket är metoden att samköra flera olika VLAN på samma kabel. I denna rapport är det Ciscos definition av trunking som används.

Fördelarna med att använda trunk-portar är att flera VLAN kan utnyttja samma kabel för informationsöverföring [22]. Detta medför att färre portar behöver användas på en switch [22]. Om användandet av bandbredd kan hållas på en låg nivå för varje VLAN kan oftast flera virtuella nätverk samsas om bandbredden som finns tillgänglig över kabeln utan någon risk för överbelastning.

Om man exempelvis har en 24-portars switch konfigurerad med fem stycken VLAN och vill utöka antalet portar med en extra switch, skulle sammankopplingen av switcharna utan trunking ta upp hela tio portar. Om trunking används skulle endast två portar behövas för att sammankoppla switcharna, en port på vardera switch. Mängden information som behöver gå över en trunk-förbindelse kan minimeras genom att låta merparten av portarna på en switch vara konfigurerade för ett specifikt VLAN, för att på så sätt undvika att broadcast-meddelanden går över trunk-förbindelsen.

3 Konfiguration

Detta kapitel ger en beskrivning av den konfiguration som genomförts för att upprätta en miljö där datorer kan anslutas till två olika virtuella nätverk samtidigt. Här beskrivs konfiguration av datorer och switch samt vilka val av operativsystem och virtualiseringsmjukvara som gjorts. Kapitlet avslutas med en beskrivning av det drifttest som genomfördes för att verifiera att miljön fungerade felfritt.

3.1 Val av mjukvara

I detta delkapitel motiveras valen av operativsystem och virtualiseringsmjukvara. Endast ett Linux-baserat operativsystem och en virtualiseringsmjukvara valdes.

3.1.1 Operativsystem

Operativsystemkärnan i Linux har stöd för konfiguration av både VLAN och virtuella nätverkskort. Utöver detta var inga andra specifika funktioner hos operativsystemet efterfrågade. Fokus låg istället på att välja en distribution som har säker intern struktur. Linux-distributionen Debian valdes eftersom den är inriktad på att ha en säker uppbyggnad genom att paketen testas grundligt innan de inkluderas i Debians repository. Dessutom är Debian brett använt i servermiljöer och har en omfattande dokumentation. Målet var att valet av Linux-distributionen inte skulle påverka resultatet av nätverkstestningen.

3.1.2 Virtualiseringsmjukvara

Det var viktigt att virtualiseringsmjukvaran skulle separera operativsystemen logiskt på hårdvarunivå genom att isolera tilldelade resurser från varandra. Detta för att de skulle agera oberoende av varandra och på så sätt garantera säkerheten. Ett annat krav var att virtualiseringsmjukvaran skulle göra det möjligt att klippa och klistra text mellan operativsystemen.

Virtuella containrar uppfattades som ett attraktivt alternativ, eftersom en sådan implementering tillåter att både värd- och gästoperativsystem kör på samma kärna. Detta innebär att färre resurser används, vilket i sin tur medför mindre prestandaförlust. Däremot hanterar virtuella containrar delningen av diverse filer i filsystemet på ett sådant sätt att operativsystem inte separeras från varandra. Därför ansågs inte virtuella containrar vara lämpliga för testmiljön.

Till skillnad från virtuella containrar hanterar både hypervisor typ 1 och typ 2 resurstilldelning på ett sätt som innebär isolering av operativsystem. Vid en första anblick kan typ 1 ses som den bästa lösningen, då den håller separationen mellan virtuella maskiner och värdoperativsystem bättre. Dock är typ 2 lättare att installera på befintliga operativsystem, vilket är en lämpligare lösning om modellen ska tillämpas på flera datorer i ett större nätverk. Dessutom är förmodligen en bra konfiguration av hypervisor typ 2 tillräcklig för att bibehålla separationen i systemet. Dessa motiveringar ledde till att hypervisor typ 2 användes i testmiljön.

Bland virtualiseringstekniker ansågs inte operating system virtualization vara ett

lämpligt alternativ, med samma anledning som virtuella containrar inte övervägdes. Native virtualization kan endast användas om processorer har virtualiseringsstöd. I övrigt fanns inga andra begränsningar för vilka tekniker som kunde tillämpas.

Virtualiseringsmjukvaran VirtualBox valdes, eftersom den uppfyllde alla kriterier. VirtualBox implementerar full virtualization och är relativt enkel att installera och konfigurera. Dessutom har mjukvaran en bred användarbas samt stöd för att klippa och klistra text mellan virtuella maskiner och värdoperativsystem, vilket var en efterfrågad funktion.

3.2 Konfiguration av miljö

Konfigurationen av switch och arbetsdatorer beskrivs i detta delkapitel. Dessutom beskrivs den övervakningsdator och den attackdator som användes vid utförandet av nätverkstester.

3.2.1 Switch

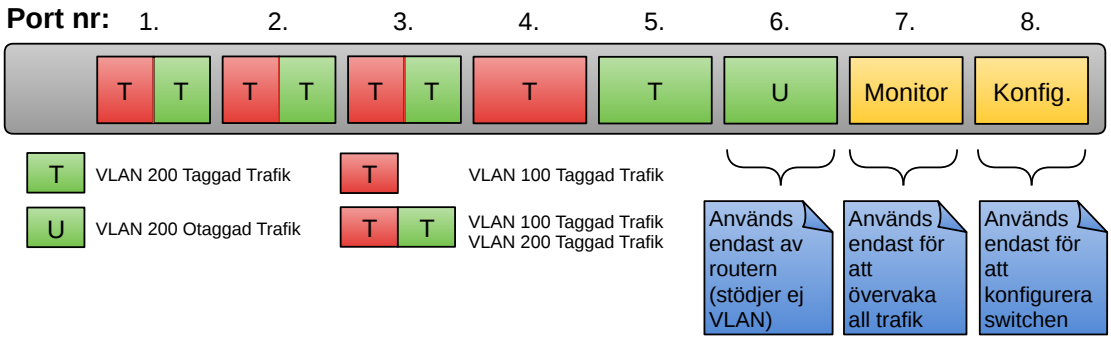
Switchen som användes var en konfigurerbar switch av modellen HP ProCurve 1810G-8 med 8 portar. Den har stöd för protokollet 802.1Q och portarna kan konfigureras för att skicka och ta emot trafik från ett eller flera VLAN.

Följande två virtuella nätverk konfigurerades för att representera ett internt och ett externt nätverk:

- VLAN 100, vilket representerar det röda interna nätverket.
- VLAN 200, vilket representerar det gröna externa nätverket med internetåtkomst.

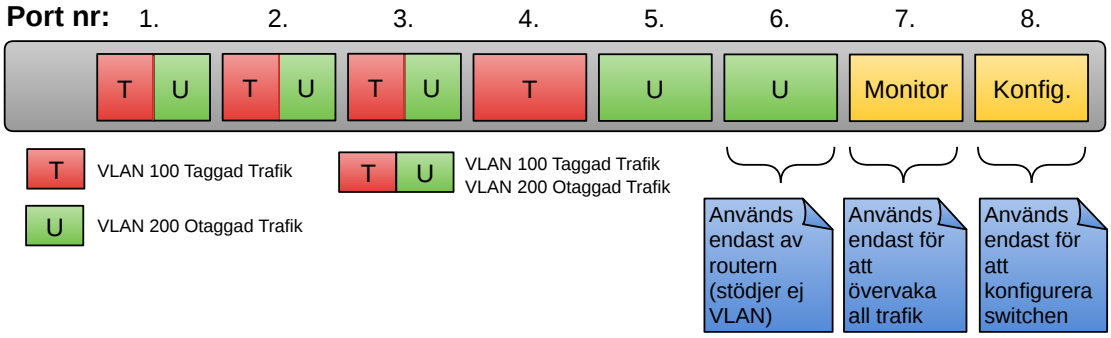
För att möjliggöra parallell trafik med både VLAN 100 och VLAN 200 konfigurerades trunk-portar, vilket krävdes då endast en nätverkskabel skulle anslutas till arbetsdatorerna. Man kan skilja på trafik som går via en trunk-port på två olika sätt. Antingen låter man all trafik vara taggad, eller så tillåter man att trafik tillhörandes maximalt ett VLAN körs otaggad. För att undersöka hur dessa två möjliga alternativ skiljde sig åt säkerhetsmässigt togs följande switchkonfigurationer i beaktande:

- Konfiguration 1 med taggad trafik på VLAN 100 och VLAN 200 (figur 7).
- Konfiguration 2 med taggad trafik på VLAN 100 och otaggad trafik på VLAN 200 (figur 8).



Figur 7: *Portvis beskrivning av konfiguration 1.*

I konfiguration 1 är portarna 1, 2 och 3 trunk-portar som endast hanterar taggad trafik, vilket illustreras i figur 7. Portarna 4 och 5 behandlar endast taggad trafik för VLAN 100 respektive VLAN 200.



Figur 8: *Portvis beskrivning av konfiguration 2.*

I konfiguration 2 är portarna 1, 2 och 3 trunk-portar som hanterar både taggad trafik för VLAN 100 och otaggad trafik för VLAN 200, vilket illustreras i figur 8. Port 4 behandlar endast taggad trafik för VLAN 100 och port 5 behandlar otaggad trafik för VLAN 200.

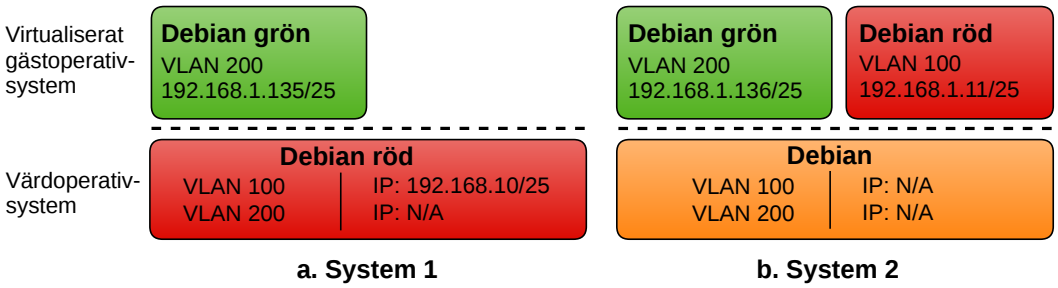
I båda konfigurationerna användes port 5 av attackdatorn, som endast hade åtkomst till VLAN 200. En router kopplades till port 6 för att förse grönt nätverk med internetåtkomst. Switchen hade även en monitor-port (port 7) för att övervaka nätverkstrafik samt en konfigurationsport (port 8), som endast användes för att konfigurera switchen.

3.2.2 Arbetsdatorer

De datorer som ansluts till både ett internt och ett externt nätverk samtidigt kommer hädanefter refereras till som arbetsdatorer. Datorerna som användes var två Dell OptiPlex 960 med Intel Core 2 Duo-processorer med klockfrekvens på 3 GHz. Inbyggt i processorerna finns hårdvarustöd för virtualisering, så kallad *Intel Virtualization Technology*. Vidare har datorerna 4 GB RAM. För övriga specifikationer se *Appendix A*.

Det kan anses utgöra en säkerhetsrisk att låta kommunikation avsedd för ett gästoperativsystem gå genom ett värdoperativsystem i de fall operativsystemen hante-

rar trafik från olika VLAN. Därför konfigurerades två olika system som testades parallellt. De konfigurerade arbetsdatorerna benämns i fortsättningen system 1 respektive system 2, se figur 9.



Figur 9: *Virtualiseringslösningarna för de två arbetsdatorerna.*

För att kunna ansluta datorerna till olika VLAN installerades tilläggspaketet vlan (se *Appendix B*) i Debian. Datorerna konfigurerades på respektive VLAN med kommandot `vconfig` samt genom att ändra i Debians konfigurationsfil (se *Appendix C*) för nätverkskort som ligger placerad på platsen `/etc/network/interfaces` i filsystemet.

System 1 och system 2 var anslutna till varsin trunk-port på switchen med åtkomst till två olika nätverk, ett säkert internt nätverk och ett nätverk med internetåtkomst. Den röda datorn (Debian röd) på vardera system kopplades till det röda nätverket och den gröna datorn (Debian grön) till det gröna nätverket.

I system 1 konfigurerades värdoperativsystemet med två VLAN-gränssnitt enligt figur 9(a). En IP-adress tilldelades till VLAN 100 som därmed användes på Debian röd. För att värdoperativsystemet inte skulle ha någon kännedom om VLAN 200 konfigurerades ingen IP-adress för detta VLAN-gränssnitt. Det virtualiserade gästoperativsystemet, Debian grön, konfigurerades för VLAN 200 och tilldelades en IP-adress enligt figur 9(a).

System 2 bestod av en avskalad Debian-distribution med två VLAN-gränssnitt. Värdoperativsystemet fungerar endast som ett underliggande system för de två virtualiserade gästoperativsystemen, se figur 9(b). Därför är inga IP-adresser tilldelade till värdoperativsystemets VLAN-gränssnitt. De båda gästoperativsystemen, Debian röd och Debian grön, konfigurerades för VLAN 100 respektive VLAN 200 och tilldelades var sin IP-adress enligt figur 9(b).

När system 1 och system 2 användes med switchkonfiguration 2 anpassades de gröna datorerna för att inte köra taggad trafik på VLAN 200. De gröna datorerna var därmed inte medvetna om att de tillhörde något VLAN, se konfiguration i *Appendix C*. Anledningen till att system 1 och system 2 anpassades var för att konfigurationen av systemen och switchen skulle stämma överens med avseende på otaggad trafik för VLAN 200.

3.2.3 Övervakningsdator

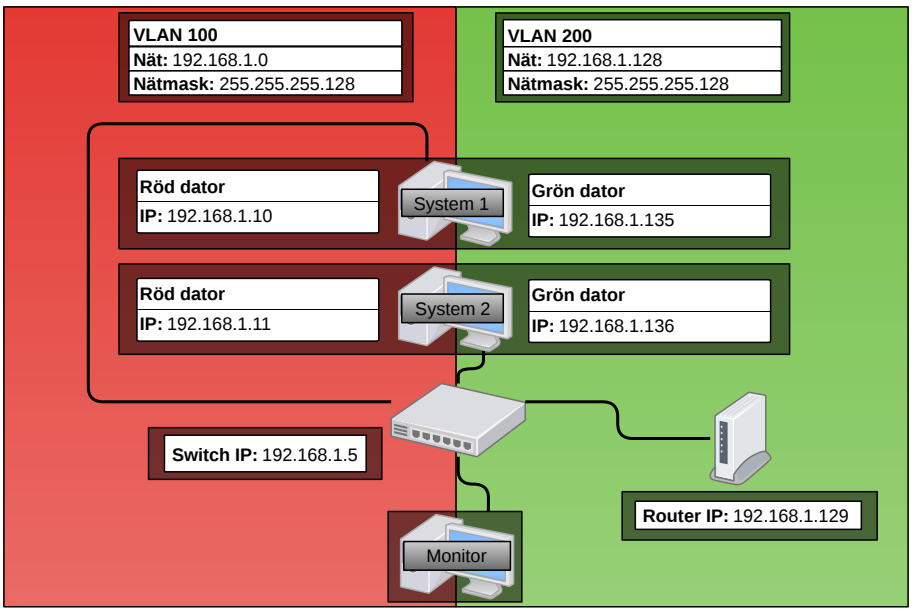
En övervakningsdator kopplades till monitor-porten på switchen, med avsikt att övervaka all trafik som går via switchen. Övervakningsdatorn sände aldrig ut några paket, och paket kunde inte heller vara adresserade till den. Det krävdes att övervakningsdatorn hade stöd för VLAN för att VLAN-taggar skulle visas i programmet Wireshark, som användes för att övervaka trafiken.

3.2.4 Attackdator

Som attackdator användes en dator med samma hårdvaruspecifikation som arbetsdatorerna. För att genomföra alla tester användes Linux-distributionen BackTrack, vilket är en distribution specifikt inriktad mot nätverkstestning. BackTrack kommer förinstallerad med alla nödvändiga verktyg som krävs för att genomföra nätverkstestning, vilket förenklade testningsförloppet.

3.2.5 IP-adresser och subnät

VLAN 100 och VLAN 200 delades in i varsitt subnät. VLAN 100 tilldelades spannet av IP-adresser `192.168.1.0-192.168.1.127` och VLAN 200 tilldelades spannet `192.168.1.128-192.168.1.255`. De olika IP-adresserna beskrivs i figur 10.



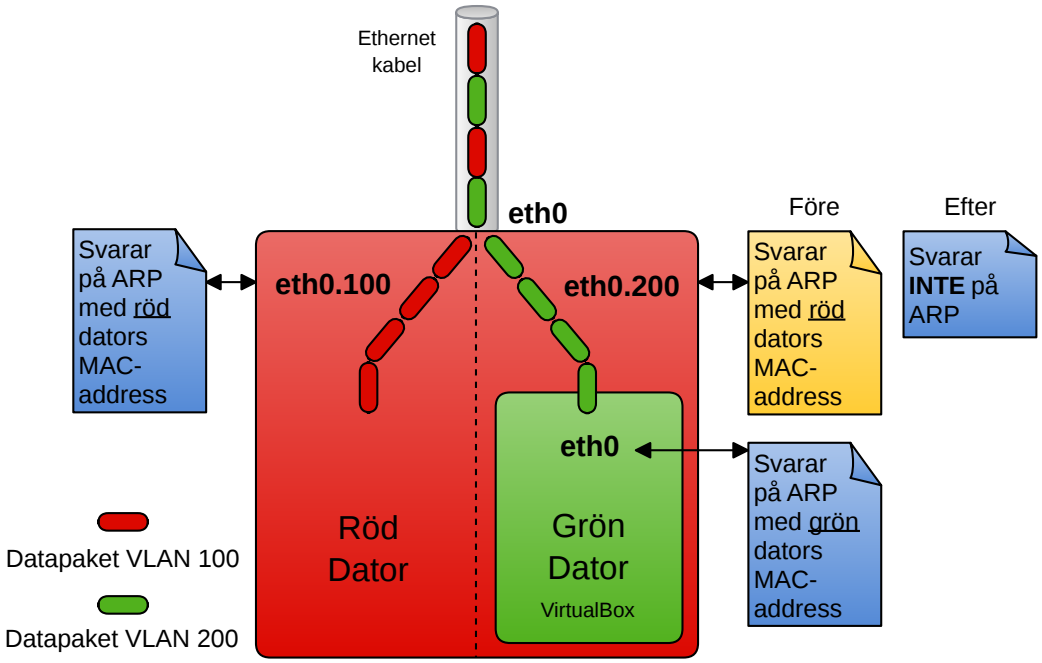
Figur 10: *IP-adresserna som användes i miljön*

I system 1 tilldelades röd och grön dator IP-adresserna 192.168.1.10 respektive 192.168.1.135. Vidare tilldelades röd och grön dator i system 2 IP-adresserna 192.168.1.11 och 192.168.1.136. IP-adresserna var satta så att varje dator tillhörde rätt IP-spann och kunde ansluta till rätt VLAN, vilket illustreras i figur 10. Eftersom övervakningsdatorn inte deltog i några paketutskick tilldelades den ingen IP-adress.

3.3 Drifttest

Ett drifttest genomfördes för varje switchkonfiguration med hjälp av metodiska ICMP-förfrågningar mellan datorer på VLAN 100 och VLAN 200. Detta för att upptäcka eventuella brister i konfigurationen av VLAN på system 1, system 2 och switchen. Drifttestet verifierade att dessa datorer var separerade samt att datorer inom samma VLAN kunde kontakta varandra. Dessutom användes programmet nmap för att skanna av nätverket, vilket gjorde det möjligt att verifiera nätverkstopologin.

Vid ett omfattande skannande från VLAN 200 med ett IP-spann som täckte både VLAN 100 och VLAN 200 svarade värdoperativsystemet i system 1, det vill säga den röda datorn på VLAN 100, på ARP-förfrågan enligt figur 11. Detta gällde för både konfiguration 1 och konfiguration 2 av switchen. Datorn svarade endast på ARP, och inte på ping eller andra förfrågningar. Trots detta kan det vara en säkerhetsrisk då man genom VLAN 200 kan få information om hur många datorer som finns på VLAN 100 samt få reda på deras MAC-adresser, vilket innebär att en attackerare kan få betydande information om potentiella mål.



Figur 11: *Beskrivning av det fysiska och de virtuella nätverkskorten på system 1. Det virtualiserade VLAN-gränssnittet eth0.200 svarade från början på ARP-förfrågan trots att ingen IP-adress var konfigurerad för gränssnittet. Efter ny konfiguration svarar eth0.200 inte längre på ARP-förfrågan, utan ARP-svaret för VLAN 200 kommer som önskat från eth0 i den gröna datorn.*

Problemet med att fel dator svarade på ARP-förfrågningar löstes genom att ändra i konfigurationsfilen `/proc/sys/net/ipv4/conf/eth0.200/arp_ignore` på den röda datorn i system 1. Ändringen gjorde att alla ARP-förfrågningar som kommer in på VLAN 200 ignoreras av värdoperativsystemet i system 1, vilket illustreras i

figur 11. Den virtuella maskinen svarade då istället på ARP-förfrågningar enligt figur 12, och den röda datorn i system 1 var endast synlig på VLAN 100 som tänkt.

Source	Destination	Protocol	Info
Dell_75:60:0d	Broadcast	ARP	Who has 192.168.1.11? Tell 192.168.1.10
CadmusCo_71:58:a5	Dell_75:60:0d	ARP	192.168.1.11 is at 08:00:27:71:58:a5

Figur 12: *Två paket i Wireshark som visar en ARP-förfrågan och ett ARP-svar.*

I figur 12 gör röd dator på system 1 en förfrågan efter IP-adressen 192.168.1.11, och får svar från switchen med tillhörande MAC-adress.

4 Systemtestning

Den enda möjliga interaktionen mellan röd och grön dator skulle vara att klippa och klistra text mellan dem. Systemtestningen syftade till att undersöka huruvida röda och gröna datorer kunde kommunicera med varandra på något annat sätt än via klippa och klistra-funktionalitet.

4.1 Klippa och klistra

I den valda virtualiseringsmjukvaran VirtualBox finns det möjlighet att konfigurera en klippa och klistra-funktion mellan värd- och gästoperativsystemen. Funktionen möjliggör klippa och klistra i en riktning eller i båda riktningarna. Det går endast att kopiera text, filer kan därmed ej flyttas mellan operativsystemen. Arbetsdatorerna konfigurerades till att kunna klippa och klistra text i båda riktningarna mellan värd- och gästoperativsystem.

4.2 VirtualBox delade mappar

I VirtualBox finns det stöd för delade mappar mellan värd- och gästoperativsystem. För att montera en delad mapp krävs root-access. Detta medför att en vanlig användare inte kan sätta upp en delad mapp mellan två operativsystem. Därför krävdes ingen modifikation av systemet för att förhindra möjligheten att dela mappar mellan operativsystemen.

4.3 Externa lagringsenheter

Eftersom olika periferienheter kan virtualiseras direkt av VirtualBox är det en säkerhetsrisk att tillåta användare att montera ett USB-minne i en röd dator, för att sedan montera USB-minnet i en grön dator och föra över information. Detta löstes genom att begränsa användares åtkomst till externa lagringsenheter genom att använda kommandot `chmod`. Kommandot exekverades med följande parametrar: `750 /media`. Katalogen `/media` är monteringsplats för alla inkopplingsbara lagringsenheter och med parametern `750` stryper man rättigheterna för vanliga användare att läsa från katalogen. Kommandot placerades i skriptet `/etc/rc.local` som körs vid varje uppstart av vardera arbetsdator.

En skillnad som uppdagades mellan system 1 och system 2 var möjligheten att montera externa lagringsenheter i den gröna datorn. I system 1, där den röda datorn körs som värdoperativsystem, måste rättigheterna vara strypta så att datorn inte kan läsa eller skriva till inkopplade enheter. Detta medförde att inte heller den gröna datorn kunde få åtkomst till periferienheter, eftersom VirtualBox körs som vanlig användare på den underliggande röda datorn.

På system 2 behövde rättigheterna till periferienheterna inte strypas på värdoperativsystemet eftersom operativsystemet inte var konfigurerat för åtkomst till varken VLAN 100 eller VLAN 200. Rättigheterna kunde därför ändras direkt på den röda datorn och lämnas oförändrade på den gröna datorn. Detta gjorde det

möjligt att montera olika externa lagringsenheter såsom USB-minnen och externa hårddiskar i den gröna datorn men inte i den röda.

4.4 Överbelastning av systemen

Eftersom de olika operativsystemen i system 1 och system 2 delar på samma hårdvara finns det risk för att en överbelastning på det ena operativsystemet kan påverka det andra. Överbelastning kan ske om resursdelningen inte fungerar som den ska och ett operativsystem har möjlighet att få mer resurser än det borde. Skulle ett operativsystem bli överbelastat så att alla tillgängliga resurser används kan det leda till att processer på det andra operativsystemet inte kan fortsätta exekvera.

För att testa överbelastning användes programmet hping3, ett program som genomför en attack där en stor mängd TCP SYN-paket skickas ut. Denna attack gör att den dator som attackeras öppnar upp ett stort antal TCP-anslutningar där varje anslutning väntar på en informationsström [24]. Detta leder till att datorn som är under attack blir överbelastad på grund av den stora mängd anslutningar som skapas. Attacken genomfördes mot grön dator i varje system för att undersöka om de röda datorerna påverkades.

Resultatet av attackerna var att den gröna datorn som mottog paketen fick arbeta hårt för att hålla uppe anslutningarna, vilket bland annat ledde till att mus och tangentbord svarade långsamt på kommando. Dock påverkades endast den gröna datorn som paketen skickades till. Den röda datorn fungerade helt obemärkt eftersom VirtualBox delade upp tillgänglig kapacitet mellan datorerna på ett rättvist sätt.

5 Nätverkstestning

I detta kapitel ges en grundlig genomgång av de säkerhetstester som genomfördes. Testerna gick ut på att utvärdera säkerheten och åtkomsten till det röda nätverket från det gröna. Det röda nätverket är ett internt nätverk och har därför ingen koppling till Internet. Däremot har det gröna nätverket tillgång till Internet och måste därför vara helt avskärmat från det röda nätverket. I idealfallet verkar därför dessa nätverk helt oberoende av varandra. Det är switchen samt respektive dators nätverkskort som hanterar trafiken mellan datorer och nätverk, vilket medför att det var dessa enheters konfigurationer som utvärderades genom testningen för att identifiera eventuella brister eller begränsningar. Inför denna testfas genomfördes en grundlig litteraturstudie kring möjliga säkerhetshot. Litteraturstudien resulterade i ett antal attacker som genomfördes för att undersöka säkerheten i miljön. Följande attacker genomfördes:

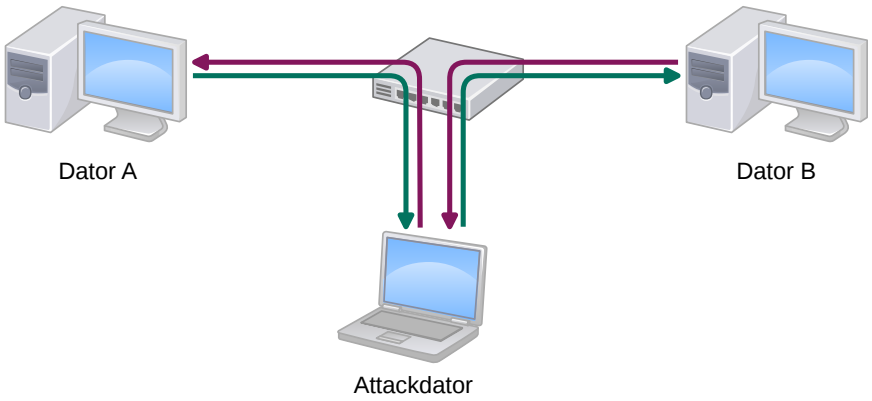
- ARP spoofing
- MAC flooding
- Manipulerad och dubbel VLAN-tagg

5.1 Beskrivning av attacker

För att få en förståelse om de attacker som genomfördes ges en översiktlig beskrivning av dessa. Varje attack beskrivs med syfte, funktion och påverkan i åtanke. Det vill säga målet med attacken, hur den fungerar och vilken påverkan den medför.

5.1.1 ARP spoofing

ARP spoofing, även kallad ARP poisoning, är en attack som utnyttjar en känd säkerhetsbrist i ARP-protokollet [23]. ARP-protokollet saknar verifikationsmekanism för att fastställa korrektheten av identiteten hos avsändaren i en ARP-förfrågan eller ett ARP-svar [23]. Denna säkerhetsbrist gör det möjligt för en attackerare att utge sig för att vara någon annan, inom samma nätverk, genom att med ett ARP-meddelande associera sin MAC-adress med någon annans IP-adress [23]. Detta medför att en attackerare exempelvis kan svara på en ARP-förfrågan som egentligen är ämnad för någon annan. Trafik som skickas till den avsedda mottagaren kommer då istället omdirigeras till attackeraren. Attackeraren kan därmed inspektera skickade paket för att sedan, om den önskar, vidarebefordra dessa till den avsedda mottagaren. Detta utan att avsändaren märker någonting. Den attackerande enheten ges på så sätt möjlighet att agera mellanhand, vilket också brukar benämnas *man-in-the-middle* [23]. Figur 13 visar hur en attackerare fångar upp paket och vidarebefordrar dessa mellan två enheter som kommunicerar med varandra.



Figur 13: *En attackdator agerar man-in-the-middle. Dator A och B har ingen vetskap om att informationen går via attackerarens dator.*

5.1.2 MAC flooding

MAC flooding syftar till att försöka fylla switchens CAM-tabell med påhittade MAC-adresser [23]. När tabellen blir full och nya MAC-adresser ska sparas medför detta att de tidigare MAC-adresserna som låg sparade i tabellen efter en tid ersätts av nya. Detta resulterar i att switchen inte längre vet vilken port informationen ska skickas ut på när olika enheter i nätverket försöker kontakta varandra. Därmed är det möjligt att switchen skickar ut informationen på samtliga portar. Beroende på implementationen av switchens CAM-tabell skickar dock switchen endast ut trafik inom varje enskilt VLAN.

5.1.3 Manipulerad och dubbel VLAN-tagg

Attacken manipulerad och dubbel VLAN-tagg har som mål att skicka iväg ett paket som innehåller modifierade 802.1Q-taggar med syftet att få paketet att hoppa mellan olika VLAN [23]. Olika varianter av attacken kan genomföras. Paket kan manipuleras med antingen en eller två taggar. Om switchen skickar paketet till en port som inte hanterar taggad trafik innebär detta att den yttre 802.1Q-taggen kommer plockas bort av switchen. Denna attack tillämpas främst då flera switchar är sammankopplade via en trunk-förbindelse och använder det fördefinierade standard-VLAN som kör otaggad trafik [23].

5.2 Genomförande av attacker

Samtliga tester genomfördes med de två olika switch-konfigurationerna konfiguration 1 och konfiguration 2, som beskrevs i kapitel *3 Konfiguration*. Varje konfiguration testades dessutom i kombination med både system 1 och system 2, vilket innebar att det för varje attack fanns 4 olika testförfaranden. För varje testförfarande identifierades fyra olika testfall. En attackerare kunde antingen sitta bakom en trunk-port eller en vanlig port, och genomförde attacker mot en dator ansluten till antingen en trunk-port eller en vanlig port.

För att genomföra attackerna användes olika mjukvaror från Linux-distributionen BackTrack av attackdatorn. Tabell 1 visar vilka mjukvaror som användes för respektive attack. Datorn som attackerna genomfördes från var enbart ansluten till det gröna nätverket under hela testfasen.

Tabell 1: *Mjukvara som användes för respektive attack.*

Attack	**Mjukvara**
ARP spoofing	Ettercap
MAC flooding	macof
Manipulerad och dubbel VLAN-tagg	Yersinia

Övervakningsdatorn användes för att övervaka all trafik under genomförandet av samtliga attacker. Detta var nödvändigt för att kunna dra slutsatser om vilken inverkan de olika attackerna hade på nätverket.

5.2.1 ARP spoofing

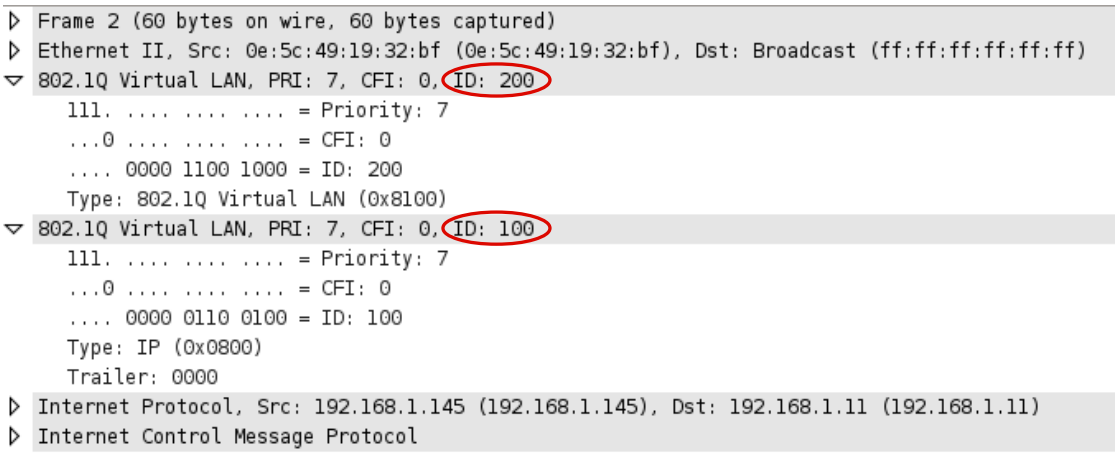
Under genomförandet av testet ARP spoofing användes programmet Ettercap. Detta är ett program som har möjlighet att skanna av ett nätverk efter tillgängliga enheter och utföra en man-in-the-middle-attack [25]. Det första som genomfördes med programmet var en skanning av nätverket efter tillgängliga enheter. Denna skanning gjordes från en attackdator som var ansluten till VLAN 200. Nätverksskanningen i Ettercap gav en lista med IP-adresser över enheter som var tillgängliga och därmed möjliga att attackera. Dessa IP-adresser låg dock alla inom VLAN 200 och därmed var det inte möjligt att genomföra ARP spoofing på datorer inom VLAN 100. För att försäkra sig om att detta verkligen var fallet genomfördes attacken ändå. Ett konstruerat ARP-paket skickades ut för varje testfall. Avsikten var att modifiera en röd dators interna ARP-tabell för att på så sätt agera man-in-the-middle mellan två röda datorer.

5.2.2 MAC flooding

För att genomföra MAC flooding användes programvaran macof på attackdatorn som var ansluten enbart till VLAN 200. Programmet genererade tusentals olika ARP-meddelanden som skickades ut på nätverket. Detta för att stresstesta switchen genom att försöka överbelasta dess CAM-tabell. De olika ARP-meddelandena som skickades ut ledde till att switchen började lagra deras källor i CAM-tabellen. Tanken var att switchen skulle tappa kontrollen över hanteringen av paket när tabellen var fylld, vilket skulle kunna innebära att paket hoppar från VLAN 200 till VLAN 100. Under genomförandet visade det sig att valet av port att genomföra attacken från inte hade någon betydelse. De beskrivna testfallen med olika portar för attacken var därmed inte aktuella för denna attack. Attacken genomfördes från attackdatorn på VLAN 200 eftersom avsikten var att få paket att nå VLAN 100.

5.2.3 Manipulerad och dubbel VLAN-tagg

Attacken manipulerad och dubbel VLAN-tagg utfördes med hjälp av programmet Yersinia som gör det möjligt att skicka paket med manipulerade VLAN-taggar. Figur 14 visar hur ett paket som manipulerats med två VLAN-taggar kan se ut.



Figur 14: *Wireshark visar hur dubbel VLAN-tagg ser ut. VLAN-ID 200 ligger som första tagg och VLAN-ID 100 som andra.*

Attacken utfördes genom att för varje testfall konstruera antingen en VLAN 100-tagg eller en inre VLAN 100-tagg följd av en yttre VLAN 200-tagg. Dessa taggningar valdes eftersom avsikten var att få paket att hoppa från VLAN 200 till VLAN 100. Eftersom en VLAN 100-tagg indikerar att paketet tillhör VLAN 100 skulle ett sådant paket potentiellt sett kunna nå VLAN 100 även om det skickas från VLAN 200. Anledningen till att även paket med två taggar testades var för att switchen, beroende på implementation och konfiguration, kan skala av den yttre VLAN 200-taggen och därmed låta den inre VLAN 100-taggen bestämma VLAN-tillhörighet.

När testfallen utfördes med konfiguration 1 var samtliga datorer konfigurerade för att använda taggad trafik. I de testfall då konfiguration 2 användes var endast de röda datorerna i system 1 och system 2 konfigurerade för att hantera taggad trafik. De gröna datorerna på båda systemen var istället konfigurerade för att endast använda otaggad trafik.

5.3 Resultat av attacker

I detta delkapitel beskrivs resultaten av de utförda attackerna. Avsnittet redovisar resultaten i tabeller för de två olika switchkonfigurationer som tagits i beaktande.

5.3.1 ARP spoofing

Resultatet av testet ARP spoofing blev lyckat ur säkerhetssynpunkt. De manipulerade ARP-paketen som skickades ut kom endast fram till de enheter som var anslutna till samma VLAN som attackdatorn, det vill säga VLAN 200. Därför var det inte möjligt att genomföra en man-in-the-middle-attack med syfte att attackera en enhet inom VLAN 100. Resultaten för konfiguration 1 och 2 beskrivs i tabellerna 2 och 3.

Tabell 2: *Resultat för varje testfall när konfiguration 1 användes. Varje testfall beskriver de modifierade ARP-paketens källa och destination samt resultatet av varje testfall.*

Konfiguration 1

	Källa		Destination		Resultat
	VLAN	Port	VLAN	Port	
1	200	T T	100	T T	Lyckades ej
2	200	T T	100	T	Lyckades ej
3	200	T	100	T T	Lyckades ej
4	200	T	100	T	Lyckades ej

T VLAN 200 Taggad trafik

T VLAN 100 Taggad trafik

T T VLAN 100 Taggad trafik
VLAN 200 Taggad trafik

Tabell 3: *Resultat för varje testfall när konfiguration 2 användes. Varje testfall beskriver de modifierade ARP-paketens källa och destination samt resultatet av varje testfall.*

Konfiguration 2

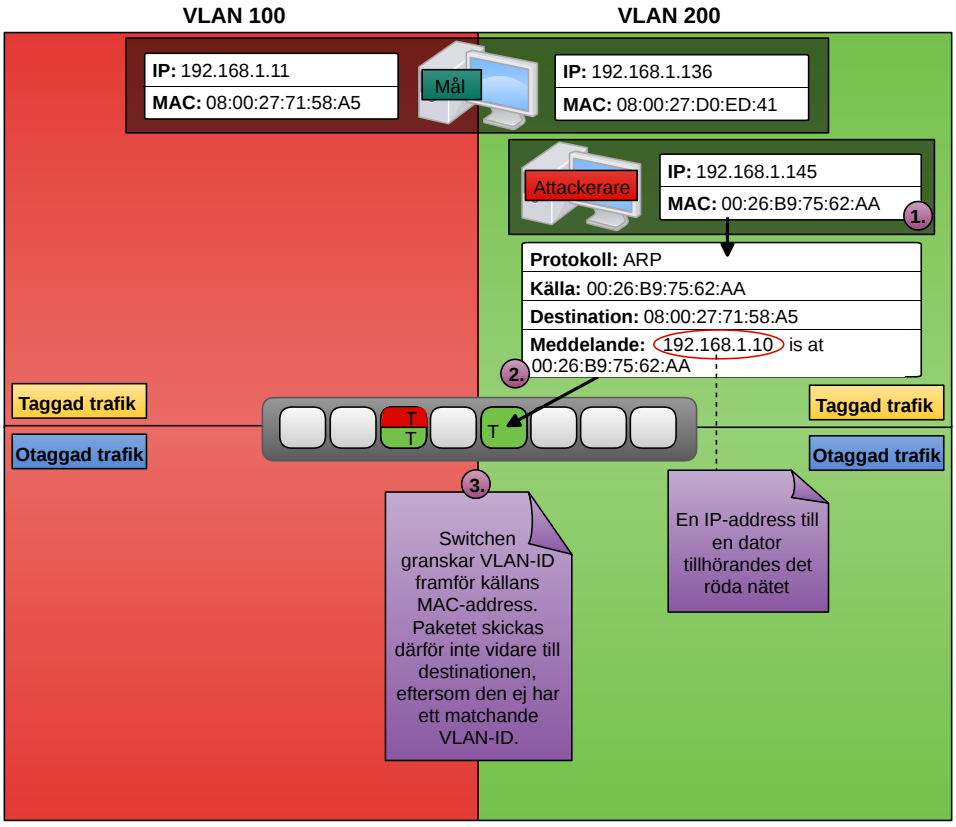
	Källa		Destination		Resultat
	VLAN	Port	VLAN	Port	
1	200	T U	100	T U	Lyckades ej
2	200	T U	100	T	Lyckades ej
3	200	U	100	T U	Lyckades ej
4	200	U	100	T	Lyckades ej

U VLAN 200 Otaggad trafik

T VLAN 100 Taggad trafik

T U VLAN 100 Taggad trafik
VLAN 200 Otaggad trafik

Samtliga testfall gav identiska resultat, vilket framgår av jämförelse mellan tabell 2 och 3. Ingen skillnad noterades mellan system 1 och system 2. Resultaten kan förklaras med att switchen hanterar de anslutna enheterna på ett sätt som gör det möjligt att gruppera dem efter VLAN-tillhörighet. Detta medför att enheter inom olika VLAN inte kan kommunicera med varandra. För att demonstrera vad som händer när en enhet inom ett VLAN försöker kontakta en enhet inom ett annat VLAN ges i figur 15 en beskrivning av testfallet på rad tre i tabell 2.



Figur 15: *En attackerare skickar ett ARP-paket från VLAN 200 som annonserar en IP-adress tillhörande en dator på VLAN 100.*

ARP-paketet som skickades från VLAN 200 kom inte fram till datorn som befann sig på VLAN 100, vilket framgår av händelseförloppet i figur 15. Detta hände på grund av att switchens CAM-tabell lagrade VLAN-ID tillsammans med en MAC-adress i varje post, se figur 16. Eftersom attackdatorn var ansluten till VLAN 200 var det detta ID som lades till MAC-adressen. Det innebär att switchen alltid kommer känna till vilket VLAN anslutna enheter tillhör. Därför uppdaterades inte måldatorns interna ARP-tabell som därmed förblev opåverkad av attacken.

MAC Address	Source Port	MAC Type
00:01:00:13:77:FB:A5:C0	8	Learned
00:01:F0:62:81:49:29:30	CPU	Management
00:64:00:26:B9:75:60:0D	1	Learned
00:64:08:00:27:71:58:A5	3	Learned
00:C8:00:1F:33:EF:D0:39	5	Learned
00:C8:00:26:B9:75:60:0D	1	Learned
00:C8:00:26:B9:75:62:AA	6	Learned
00:C8:08:00:27:5A:8C:C3	1	Learned
00:C8:08:00:27:D0:ED:41	3	Learned

Figur 16: *Skärmdump på switchens CAM-tabell. På den markerade raden visas hur MAC-addressen från attacken läggs till i CAM-tabellen med två extra bytes för VLAN-ID.*

5.3.2 MAC flooding

Switchen mottog ständigt ARP-meddelanden från attackdatorn på VLAN 200 och registrerade de påhittade MAC-adresserna i CAM-tabellen. Detta medförde att alla MAC-adresser tillhörande både VLAN 100 och VLAN 200 som sparats tidigare ersattes av nya efter en viss tid. Totalt registrerade tabellen 8192 stycken adresser innan den var fylld.

Eftersom attackdatorn tillhörde VLAN 200 fylldes switchens CAM-tabell med MAC-adresser som fick ID 200. Därmed kunde det konstateras att denna tabell är gemensam för alla VLAN. Om tabellen inte hade delats av alla VLAN skulle det fortfarande funnits kvar poster för datorer tillhörandes VLAN 100. Den gemensamma CAM-tabellen kan ses som en säkerhetsbrist eftersom resultatet av attacken blev att switchen började fungera som en hubb i båda nätverken och all trafik skickades ut till samtliga noder inom samma VLAN. Paket hoppade inte mellan VLAN eftersom switchen vet vilket VLAN paket skickades från och skickade därmed enbart ut paketen på detta VLAN. Resultatet blev detsamma för både konfiguration 1 och konfiguration 2, och var helt oberoende av vilka datorer som var anslutna till switchen. Attacken visade att det är möjligt att påverka VLAN 100 genom att skicka ARP-paket från en dator på VLAN 200, både med konfiguration 1 och konfiguration 2.

Switchen skulle kunna erbjuda högre säkerhet genom att ha separata CAM-tabeller för varje VLAN. Detta skulle resultera i att MAC flooding på VLAN 200 aldrig skulle skriva över MAC-adresser tillhörande enheter på VLAN 100.

5.3.3 Manipulerad och dubbel VLAN-tagg

När konfiguration 1 testades lyckades inte något paket som skickades från VLAN 200 till VLAN 100 komma fram. I tabell 4 visas en översikt över de resultat som registrerades. Paketen som skickades ut från VLAN 200 modifierades enligt kolumnen *VLAN-taggning* medan kolumnen *Leverans* indikerar om paketet levererades till destinationen.

Tabell 4: *Resultat för varje testfall när konfiguration 1 användes. Varje testfall beskriver de modifierade alternativt dubbeltaggade paketens källa och destination samt resultatet av testfallet.*

Konfiguration 1						
	Källa		Destination		VLAN-taggning	Leverans
	VLAN	Port	VLAN	Port	Från källa	
1	200	T T	100	T T	200 + 100	Nej
2	200	T T	100	T T	100	Nej
3	200	T T	100	T	200 + 100	Nej
4	200	T T	100	T	100	Nej
5	200	T	100	T T	200 + 100	Nej
6	200	T	100	T T	100	Nej
7	200	T	100	T	200 + 100	Nej
8	200	T	100	T	100	Nej

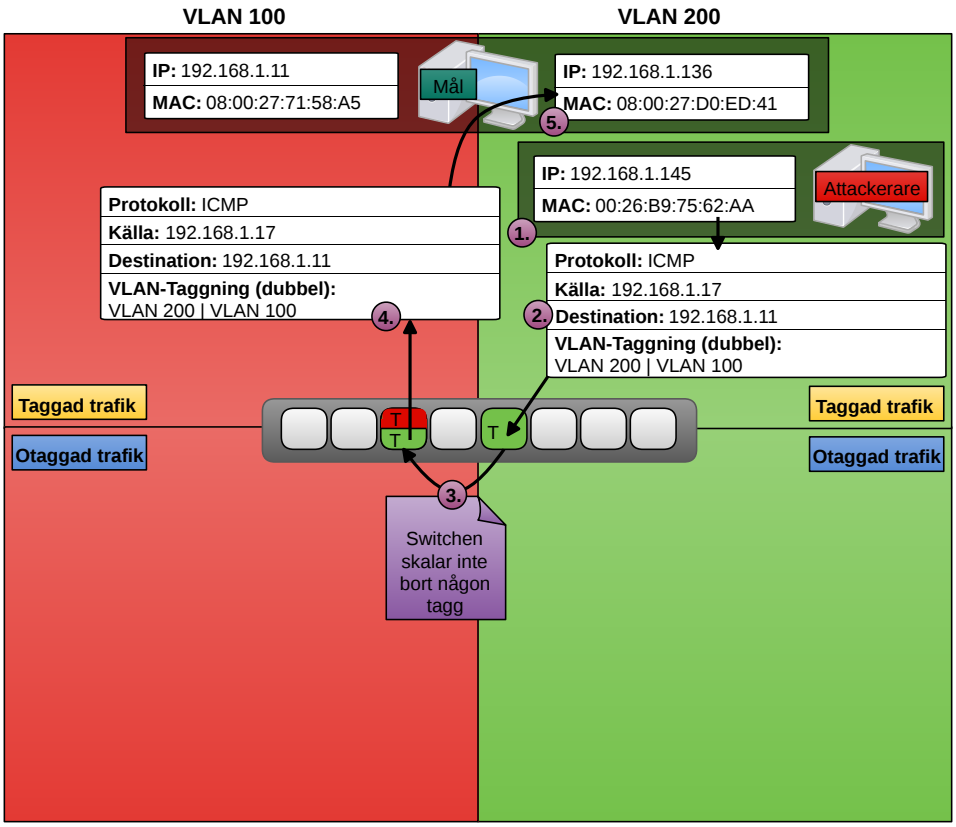
T VLAN 200 Taggad trafik

T VLAN 100 Taggad trafik

T T VLAN 100 Taggad trafik
VLAN 200 Taggad trafik

Inga paket lyckades ta sig från VLAN 200 till VLAN 100 i något av testfallen, vilket framgår av tabell 4. Resultatet blev detsamma för både system 1 och system 2. När alla portar hanterar taggad trafik begränsas möjligheten att försöka få switchen att behandla paket som om det skickades från ett annat VLAN. I samtliga testfall lade operativsystemet på den gröna datorn till en yttre VLAN-tagg med ID 200 på paketet som skickades, även i de fall då paketet redan hade en sådan tagg. Både källan och destinationen satt på taggade portar tillhörande antingen VLAN 100 eller VLAN 200, vilket medförde att switchen aldrig skalade av någon VLAN-tagg på paket som skickades. Detta innebar att yttre taggen alltid utgjordes av ID 200. Eftersom switchen inte hade kännedom om destinationens MAC-adress på VLAN 200, agerade switchen som en hubb och skickade ut paketet till alla anslutna enheter på VLAN 200 istället. Det är först när den kommit fram till gröna datorer som den yttre VLAN-taggen skalas bort av operativsystemet, vilket påvisar att paketet aldrig nådde någon röd dator. Detta skedde endast i testfall 1, 2, 5 och 6 i tabell 4 då destinationen är ansluten till en trunk-port. I övriga fall anlände paketet aldrig till destinationen eftersom den inte hade åtkomst till VLAN 200.

I figur 17 illustreras testfall 6, i vilket ett paket med dubbel VLAN-tagg skickades från en taggad VLAN 200-port. Eftersom destinationen var en taggad trunk-port skalade inte switchen av någon tagg från paketet, vilket medförde att paketets VLAN-ID förblev 200.



Figur 17: *Paket som skickas med taggarna VLAN 200 och VLAN 100. Paketet skickas från en taggad VLAN 200-port och når fram till gröna datorn på VLAN 200.*

Switchen tog emot paketet och läste av den första VLAN-taggen med ID 200. Därefter skickades paketet ut oförändrat på VLAN 200 och levererades till den gröna datorn med dubbel VLAN-tagg enligt figur 17. Vad som testades var huruvida switchen skulle skala bort enbart den första VLAN-taggen och därefter vidarebefordra paketet ut på VLAN 100. Detta skedde dock inte och försöket kan anses lyckat ur säkerhetssynpunkt.

Testet av konfiguration 2 resulterade i flera intressanta resultat om hur switchen hanterade dubbla VLAN-taggar då data skickades från en port som var konfigurerad för otaggad trafik. Resultatet blev att paketen i flera testfall levererades till den röda datorn på VLAN 100, vilket beskrivs i tabell 5. Att paketen kom fram berodde på att trunk-portarna kombinerade taggad och otaggad trafik. Samma resultat uppnåddes vid test av både system 1 och system 2.

Tabell 5: *Resultat för varje testfall när konfiguration 2 användes. Varje testfall beskriver de modifierade alternativt dubbeltaggade paketens källa och destination samt resultatet av testfallet.*

Konfiguration 2						
	Källa		Destination		VLAN-taggning	Leverans
	VLAN	Port	VLAN	Port	Från källa	
1	200	T U	100	T U	200 + 100	**Ja**
2	200	T U	100	T U	100	**Ja**
3	200	T U	100	T	200 + 100	**Nej**
4	200	T U	100	T	100	**Ja**
5	200	U	100	T U	200 + 100	**Ja**
6	200	U	100	T U	100	**Nej**
7	200	U	100	T	200 + 100	**Nej**
8	200	U	100	T	100	**Nej**

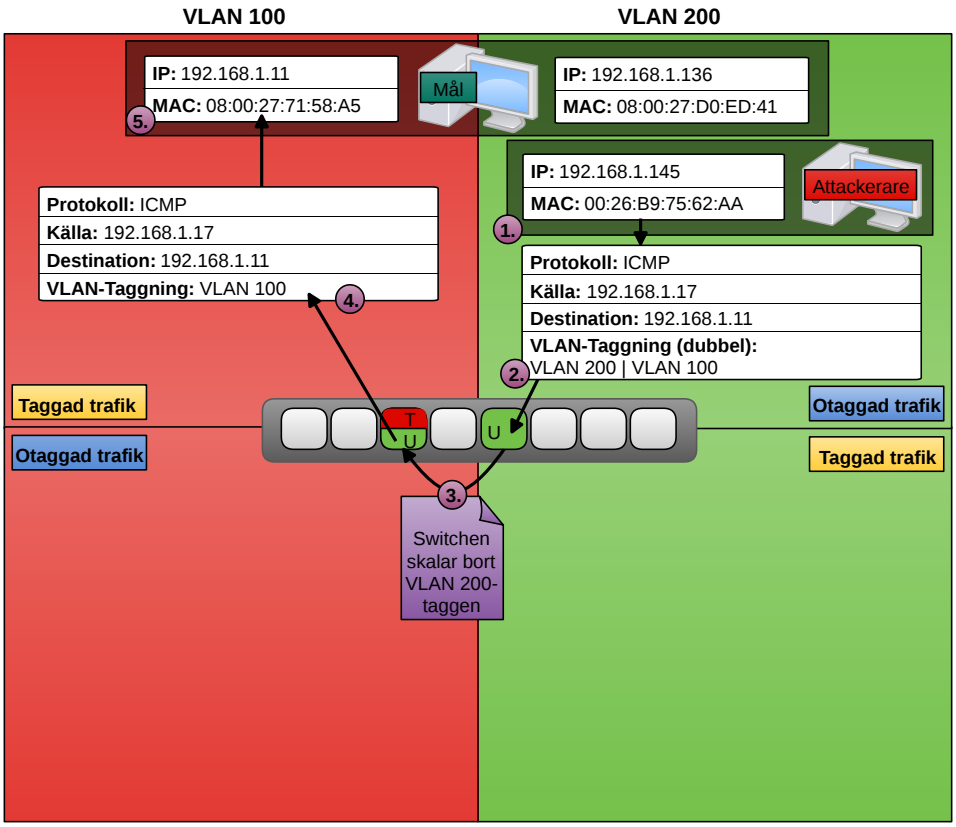
U VLAN 200 Otaggad trafik

T VLAN 100 Taggad trafik

T U VLAN 100 Taggad trafik
VLAN 200 Otaggad trafik

I testfall 1, 2, 4 och 5 (se tabell 5) levererades paketen som skickades från VLAN 200 till röd dator på VLAN 100. Detta innebär att paket hoppade från VLAN 200 till VLAN 100, vilket är en stor säkerhetsbrist. Testfall 2 och 4 hade gemensamt att attackeraren var ansluten till en trunk-port och att enbart en VLAN-tagg med ID 100 skickades. Eftersom paketen skickades från en otaggad port lades det aldrig på någon extra VLAN-tagg, vilket medförde att switchen fick paketen med VLAN-ID 100. Detta resulterade i att paketen skickades vidare ut på VLAN 100 och levererades till den röda datorn.

I testfall 1 och 5 skickades dubbla VLAN-taggar, en med VLAN-ID 200 och en med VLAN-ID 100. I testfall 1 var attackdatorn ansluten till en trunk-port och i testfall 5 var den ansluten till en vanlig port inställd för otaggad trafik på VLAN 200. Vilken port attackeraren tillhörde hade dock ingen inverkan på resultatet, som blev detsamma i båda testfallen. Eftersom attackeraren satt bakom en port som tillät otaggad trafik ändrades inte VLAN-taggningen och paketen levererades till switchen med dubbla VLAN-taggar. Då destinationen också hanterade otaggad trafik plockade switchen bort den yttersta taggen med VLAN-ID 200 och lämnade kvar den inre taggen med VLAN-ID 100. Detta resulterade i att paketen skickades ut på VLAN 100 och levererades till den röda datorn. I figur 18 beskrivs testfall 5 i detalj.



Figur 18: *Dubbel VLAN-tagg i ordningen VLAN 200, VLAN 100 skickad från en otaggad VLAN 200-port till en trunk-port med taggad trafik på VLAN 100 och otaggad trafik på VLAN 200.*

Steg 1 i figur 18 representerar attackdatorn som var inkopplad på en port konfigurerad för otaggad trafik på VLAN 200. Ett paket med dubbla VLAN-taggar skapades och skickades med en röd dator på VLAN 100 som destination. Eftersom den röda datorn var ansluten till en trunk-port, som var konfigurerad för otaggad trafik på VLAN 200, skalade switchen av den yttre taggen. Kvar blir den inre taggen med VLAN-ID 100, vilket visas i steg 4. I och med att taggen hade VLAN-ID 100 levererades paketet till den röda datorn. När den röda datorn försökte svara på anropet skickade den ett paket taggat med VLAN 100, vilket switchen inte vidarebefordrade till attackdatorn eftersom den röda datorn tillhörde ett annat VLAN. Attacken fungerade alltså endast åt ett håll, vilket dock är tillräckligt för att på något sätt kunna komma åt VLAN 100. Det visade sig att kombinationen av taggad och otaggad trafik på trunk-portar ledde till att VLAN 100 och VLAN 200 inte förblev helt separerade.

6 Resultat

I detta kapitel ges en sammanställning av de resultat som presenterats i rapporten under kapitel *3 Konfiguration*, *4 Systemtestning* och *5 Nätverkstestning*. Resultatsammanställningen följer samma struktur som används i rapporten. Det första som sammanställs är resultatet för *Konfiguration*. Därefter följer resultatet för *Systemtestning* och kapitlet avslutas med en resultatsammanställning av *Nätverkstestning*.

6.1 Konfiguration

Arbetet resulterade i en fullt fungerande miljö, som visade sig kunna ersätta två fysiska datorer kopplade till två fysiskt separerade nätverk genom användandet av VLAN och virtualisering. Vid normal användning fungerade de två virtuella nätverken som tänkt, och all trafik bevarades inom respektive VLAN. Både konfiguration 1 och konfiguration 2 för switchen kunde användas för att åstadkomma det ovan nämnda resultatet. En viktig skillnad uppdagades dock vid jämförelsen av system 1 och system 2. Under drifttestet visade det sig att den röda datorn i system 1 felaktigt svarade på ARP-förfrågan från det gröna nätverket. Detta kunde dock åtgärdas genom att den röda datorn i system 1 konfigurerades om.

6.2 Systemtestning

En brist som upptäcktes vid systemtestningen var att användare hade tillgång till externa lagringsenheter. Problemet löstes genom att användares rättigheter begränsades. Lösningen resulterade i att inga externa lagringsenheter kunde användas på varken den röda eller den gröna datorn i system 1. Med samma lösning för system 2 kunde den gröna datorn fortfarande nyttja de externa lagringsenheterna eftersom rättigheterna kunde begränsas på endast den röda datorn.

Den metod som används av VirtualBox för att dela ut mappar mellan värd- och gästoperativsystem ansågs inte utgöra ett säkerhetsproblem, eftersom det krävs att en användare aktivt delar ut en mapp på värdoperativsystemet samtidigt som den har root-access på gästoperativsystemet. Däremot möjliggjordes klippa och klippa text i VirtualBox mellan operativsystem, vilket inte medförde några säkerhetsrisker om användaren inte medvetet för över intern information.

Överbelastningsattacken som genomfördes under systemtestningen resulterade i att den gröna datorns funktionalitet blev kraftigt försämrad medans den röda datorn förblev opåverkad. Detta visade att delningen av resurser i VirtualBox hanterades på ett tillfredsställande sätt.

6.3 Nätverkstestning

För konfiguration 1 resulterade nätverkstestningen i att system 1 och system 2 klarade samtliga nätverkstester. Varken ARP spoofing, MAC flooding eller manipulerad och dubbel VLAN-tagg lyckades få trafik att hoppa mellan rött och grönt nätverk. Däremot lyckades testet MAC flooding få switchen att fungera som en

hubb och all trafik skickades därmed ut till samtliga noder inom respektive VLAN. Det var alltså möjligt att påverka hur switchen hanterade paketen i det röda nätverket genom att utföra en MAC flooding-attack från det gröna nätverket. Dock var det fortfarande ingen trafik som hoppade mellan nätverken.

För konfiguration 2 klarade varken system 1 eller system 2 nätverkstestningen. Det var attacken modifierad och dubbel VLAN-tagg som utgjorde en säkerhetsbrist. Med modifierad VLAN-taggning var det möjligt att få trafiken att hoppa från det gröna till det röda nätverket. Problemet berodde på att konfiguration 2 kombinerade taggad och otaggad trafik till olika VLAN på en och samma trunkport. Testerna ARP spoofing och MAC flooding gav samma resultat för både konfiguration 1 och konfiguration 2.

7 Diskussion

I detta kapitel förs en diskussion som behandlar den miljö som konfigurerats och resultaten från de genomförda testerna. Diskussionen består av två delar. Den första delen är en evaluering av miljön, som tar upp viktiga resultat och säkerhetsrisker i miljön. I den andra delen diskuteras den konfigurerade miljön i ett större sammanhang. Det diskuteras kring hur den framtagna miljön lämpar sig för implementation inom företag och organisationer. Frågan som behandlas är om det är möjligt och även relevant att skala upp den framtagna miljön för att kunna implementeras och användas i större skala.

7.1 Evaluering av miljön

Resultatet visade att det går att konfigurera en miljö för att säkert ansluta en enda dator till två olika virtuella nätverk. Däremot är säkerheten i miljön helt beroende av hur switchar och datorer konfigureras. Till exempel konstaterades det att konfiguration 2 inte bör användas på grund av att konfigurationen inte klarade testet modifierad och dubbel VLAN-tagg. Taggad och otaggad trafik bör därför aldrig blandas på samma kabel, vilket är något som även [23] har konstaterat. En enda felkonfiguration av en switch eller dator är tillräcklig för att det ska skapas en koppling mellan det interna och det externa nätverket, vilket innebär att säkerheten i det interna nätverket äventyras.

En annan stor säkerhetsrisk med den miljö som konfigurerats är om en attackerare får åtkomst att konfigurera switchen. Därför bör tillräckliga medel vidtas för att säkerställa att switchens konfiguration inte kan ändras av någon annan än en administratör. Detta kan åstadkommas genom att tillåta konfiguration på endast en av switchens portar samt att ett säkert lösenord sätts. Dessutom är det en fördel att begränsa den fysiska åtkomsten till switchen för att förhindra möjligheten att koppla om nätverkskablar eller att göra en total återställning.

På samma sätt är det viktigt att användare inte har möjlighet att ändra inställningar för nätverkskortet. Om inställningarna ändras på något sätt är risken stor att trafik från de två nätverken inte separeras helt och hållet, vilket potentiellt sett gör det möjligt för en attackerare att lyssna på trafik tillhörandes det interna nätverket. I Linux-baserade operativsystem är separeringen av rättigheter mellan administratör och vanlig användare stor. Vanliga användare har väldigt få rättigheter, till exempel har de inte möjlighet att ändra nätverksinställningar eller andra systeminställningar. Vid användande av andra operativsystem kan det dock vara nödvändigt att begränsa användarens rättigheter.

När det kommer till möjliga attacker som kan genomföras mot den konfigurerade miljön var MAC flooding den enda attacken som påvisade en viss svaghet även för konfiguration 1. Den upptäckta svagheten möjliggör för en attackerare att genom MAC flooding från ett VLAN åstadkomma att switchen skickar ut paket på alla portar tillhörande ett annat VLAN. Detta leder till en onödigt stor bandbreddsanvändning vilket kan göra att nätverket blir överbelastat. Enda sättet att undvika detta är att manuellt koppla ur datorn som utför attacken. Genom att använda

switchens SNMP-funktion kan CAM-tabellen övervakas av en SNMP-server som meddelar en ansvarig om en attack utförs. Den som är ansvarig kan då koppla bort den dator som attacken genomförs från. På andra typer av switchar kan det dock vara möjligt att begränsa antalet unika MAC-adresser som får höra till en specifik port på switchen, vilket skulle kunna lösa problemet utan att involvera manuellt arbete.

Med tanke på att nätverket kan utsättas för MAC flooding samt att varje dator utgör en potentiell koppling mellan de två virtuella nätverken, så kan säkerheten ej sägas motsvara säkerheten som uppnås med två fysiska datorer kopplade till två fysiskt separerade nätverk. Två fysiskt separerade nätverk har inga som helst kopplingar mellan varandra, och är därför per definition två helt isolerade nätverk. Däremot är de två virtuella nätverken inte helt isolerade, och det kan därför vara en bra idé att komplettera med exempelvis brandväggar och kryptering. Speciellt i de fall där det är kritiskt att säkerheten är hög.

7.2 Lämpar sig miljön för implementation i större skala?

Eftersom olika switchar och system sannolikt kräver olika konfigurationer kan det för företag och organisationer bli betydligt svårare att sätta upp en säker miljö. I detta fall kan det vara fördelaktigt att först finna en fungerande lösning för ett mindre nätverk, och sedan använda samma hård- och mjukvara för att bygga upp ett nätverk i större skala.

För företag och organisationer kan det vara en fördel att implementera system 1 istället för system 2. Om Debian eller någon annan Linux-distribution redan finns installerad på datorn krävs endast att VirtualBox och ett gästoperativsystem installeras. För att implementera system 2 krävs en ominstallation av hela systemet eftersom det underliggande operativsystemet endast är tänk att användas för att separera de virtualiserade systemen.

När en fungerande konfiguration har gjorts på en dator är det enkelt att utföra samma konfiguration på fler datorer. Förmodligen har de flesta företag någon form av distributionslösning för att uppdatera och underhålla sina datorer och kan därmed utföra samma konfiguration på samtliga enheter. Detta gör det relativt enkelt att utföra konfigurationen även om det rör sig om en stor organisation med många enheter. Värt att nämna är dock att nya säkerhetsbrister potentiellt kan introduceras i storskaliga nätverk, vilket inte har tagits i beaktande i denna rapport. Sådana säkerhetsbrister kan introduceras på grund av att företagsanpassade switchar vanligtvis stödjer fler protokoll, som i sin tur kan exploateras med andra uppsättningar av attacker.

På det stora hela måste man också ställa sig frågan om det är rimligt att företag och organisationer ska behöva förändra hela nätverkstopologin och använda sig av virtualisering på varje enskild dator, enbart för att separera intern och extern datakommunikation. En annan utgångspunkt skulle kunna vara att införa en server-baserad virtualiseringslösning. Servern skulle kunna konfigureras för internetåtkomst och anställda skulle bara behöva tunna klienter. De anställda skulle

kunna ges åtkomst till virtuella maskiner, som körs på servern, för att få tillgång till både intranät och Internet. På så sätt skulle man kunna undvika större förändringar i nätverkstopologin. Man behöver inte heller installera en virtualiseringsmjukvara och ett gästoperativsystem på varje dator.

Även ur säkerhetssynpunkt skulle en sådan lösning kunna anses vara bättre, eftersom antalet potentiella kopplingar mellan de två nätverken minskas drastiskt. I den miljö som behandlas i denna rapport kan den logiska separeringen av nätverken äventyras genom en felaktig konfiguration på vilken dator som helst, eftersom varje arbetsdator behandlar trafik från båda nätverken. Med en server-baserad virtualiseringslösning kan kopplingen mellan nätverken åtminstone isoleras till en eller några enstaka punkter, nämligen i servern eller servrarna som används för virtualisering. Poängen med att separera nätverk är att eliminera kopplingarna mellan dem. På samma sätt som en dator tillhörande två fysiska nätverk kan agera brygga mellan dem kan det också anses vara olämpligt att låta datorer indirekt agera brygga mellan två virtuella nätverk. Ju fler potentiella kopplingar det finns mellan två nätverk desto fler potentiella felkällor finns det.

8 Slutsats

Resultatet från testerna visar att VLAN i kombination med virtualisering ger en tillräckligt hög säkerhet i den konfigurerade miljön. Det säkra interna nätverket förblir dock endast separerat från det externa nätverket då uteslutande taggad trafik används i kombination med välfungerande switch- och datorkonfigurationer. Trunk-portar bör således aldrig konfigureras med både otaggad och taggad trafik. Oavsett konfiguration går det dessutom att påverka prestandan i det interna nätverket genom att utföra en MAC flooding-attack från det externa nätverket.

Trots att miljön ansågs vara tillräckligt säker är lösningen inte nödvändigtvis väl anpassad för företag och organisationer. Detta på grund av att både virtualiseringsmjukvara och gästoperativsystem måste installeras på varje enskild dator. Dessutom måste varje dator ha stöd för VLAN och konfigureras för att tillhöra trunk-portar. Varje dator utgör därför en potentiell koppling mellan två VLAN, eftersom de behandlar trafik från dem båda. Därmed kan en felaktig konfiguration av en enda dator eller switch leda till att intern och extern datakommunikation inte separeras.

Eftersom nätverket i den uppsatta miljön påverkas negativt av MAC flooding samt att varje dator utgör en potentiell koppling mellan de två virtuella nätverken, så kan säkerheten ej sägas motsvara säkerheten som uppnås med två fysiska datorer kopplade till två fysiskt separerade nätverk.

9 Framtida arbeten

Eftersom VLAN ofta används i större nätverk, med betydligt fler noder än i den miljö som behandlades i denna rapport, är det troligt att en verklig implementation skulle kunna ge upphov till andra säkerhetsbrister. Här finns utrymme för att utföra flera intressanta nätverkstester på en miljö som mer liknar en implementation som kan tänkas användas av företag och organisationer. Eftersom endast en switch användes i detta arbete och denna switch saknade många av de mer avancerade funktionerna som används i större nätverk kunde inga tester utföras på sådana routing- och nätverksprotokoll. Det kan därför vara av intresse att undersöka hur väl sådana routing- och nätverksprotokoll fungerar i kombination med VLAN.

Det kan även vara av intresse att undersöka om syftet kan uppnås med någon annan teknologi än VLAN. En möjlighet skulle vara att använda kryptering och en proxyserver med någon form av verifiering. På datorn skulle olika applikationer kunna separeras från resten av systemet med hjälp av virtuella containrar. Om behovet enbart är att ett fåtal protokoll såsom HTTP och FTP behöver användas så skulle detta kunna vara en bra lösning. Ytterligare en lösning skulle kunna vara att köra en server-baserad virtualiseringslösning, till exempel XenDesktop [26]. Användare skulle då kunna köra virtualiserade system genom att ansluta till en server, vilket skulle innebära att användarnas datorer varken behöver konfigureras för VLAN eller köra flera operativsystem.

Referenser

[1] Y.-P. Lai och R.-H. Dai, "The implementation guidance for practicing network isolation by referring to ISO-17799 standard," *Computer Standards and Interfaces*, vol. 31, nr. 4, juni 2009, ss. 748–756.

[2] G. Leischner och C. Tews, "Security Through VLAN Segmentation: Isolating and Securing Critical Assets Without Loss of Usability," i *9th Annual Western Power Delivery Automation Conference*, Spokane, WA, 2007.

[3] *Media access control (MAC) bridges and virtual bridged local area networks - amendment 19: PBB-TE infrastructure segment protection*, IEEE Standard 802.1Qbf-2011.

[4] R. Farrow, "VLANs: Virtually insecure?" *IT Architect*, vol. 18, nr. 3, 2003, ss. 62–63.

[5] S. A. Rouiller, "Virtual LAN Security: weaknesses and countermeasures," GIAC Security Essentials Practical Assignment - Version 1.4b.

[6] J. Liu och W. Lai, "Security Analysis of VLAN-based Virtual Desktop Infrastructure," i *2010 International Conference on Educational and Network Technology*, 2010, ss. 301–304.

[7] N. Alströmer, L. Book, F. Carlsson, *et al.*, "Dubbla nät i samma dator - (Blanda rött/grönt) Kandidatarbete inom Data-och informationsteknik," 2011.

[8] J. Daniels, "Server virtualization architecture and implementation," *Crossroads*, vol. 16, nr. 1, september 2009, ss. 8–12.

[9] G. J. Popek och R. P. Goldberg, "Formal Requirements for Virtualizable Third Generation Architecture," *Communications of the ACM*, vol. 17, nr. 7, 1974, ss. 412–421.

[10] J. Brodkin, "A look at bare-metal hypervisor," *Network World*, vol. 27, nr. 14, 2010, ss. 1,11.

[11] J. Hoopes, *Virtualization for Security: Including Sandboxing, Disaster Recovery, High Availability, Forensic Analysis, and Honeypotting.* Burlington, MA: Syngress Publishing, INC, 2009.

[12] P. Lindstrom, "5 LAWS OF VIRTUALIZATION SECURITY," *Baseline*, vol. 1, nr. 84, 2008, ss. 54–57.

[13] R. Day, "Virtualization Makes Better use of Open-Source OSes and apps," *Electronic Engineering Times*, vol. 1, nr. 1558, 2009, ss. 39–40.

[14] S. Soltesz, H. Pötzl, M. E. Fiuczynski, *et al.*, "Container-based operating system virtualization: a scalable, high-performance alternative to hypervisors," i *Proceedings of the 2nd ACM SIGOPS/EuroSys European Conference on Computer Systems 2007*, Lisbon, mars 2007, ss. 275–287.

[15] G. Harrison, "Virtualization Architectures Do Make a Difference," *Database Trends and Applications*, vol. 24, nr. 2, 2010, s. 32.

[16] VMware, "Understanding Full Virtualization, Paravirtualization and Hardware Assist," http://www.vmware.com/files/pdf/VMware_paravirtualization.pdf, 2007, tillgänglig: 2012-02-20.

[17] J. F. Kurose och K. W. Ross, *Computer Networking: A Top-Down Approach (international Edition)*, 5e upplagan. Boston, MA: Pearson, 2010.

[18] H. Zimmermann, "OSI Reference Model–The ISO Model of Architecture for Open Systems Interconnection," *IEEE Transactions on communications*, vol. 28, nr. 4, 1980, ss. 425–432.

[19] "Information technology - Open Systems Interconnection - Basic Reference Model: The Basic Model," Geneva, 2003, ISO/IEC 7498-1:1994.

[20] R. Braden, "Requirements for Internet Hosts - Communication Layers," http://tools.ietf.org/html/rfc1122, 1989, tillgänglig: 2012-05-14.

[21] *AlliedWare Plus OS overview of VLANs (Virtual LANs)*, Allied Telesis, 2008, tillgänglig: 2012-02-14.

[22] J. Kane, Ed., *Cisco networking academy program CCNA 3 and 4 companion guide*, 3e upplagan. Indianapolis, IN: Cisco Press, 2004.

[23] "Virtual LAN Security Best Practices," Cisco Systems, Inc, 2002.

[24] W. Eddy, "TCP SYN Flooding Attacks and Common Mitigations," http://tools.ietf.org/html/draft-ietf-tcpm-syn-flood-05, 2007, tillgänglig: 2012-05-14.

[25] D. Norton, "An Ettercap Primer," april 2004, GIAC Security Essentials Practical Assignment - Version 1.4b.

[26] Citrix systems, "Xendesktop," http://www.citrix.com/xendesktop, tillänglig: 2012-05-14.

Appendix

A Hårdvara

System 1 och System 2

Dator	DELL OptiPlex 960
CPU	Intel® Core™2 Duo CPU E8400 @ 3.00GHz Intel® Virtualization Technology Intel® Trusted Execution Technology
Minne	4GB (2x2GB) DDR2 @ 800MHz
Chipset	Intel® Q45 Express
Nätverkskort	Integrated 10/100/1000 Ethernet (Intel WG82567LM LOM)
Grafikkort	GeForce 8600 GT 512MB RAM
Hårddisk	Western Digital WD3200AAKS-75L9A0 320GB

`http://www.dell.com/se/foretag/p/optiplex-960/pd`

Switch

HP Procurve 1810G-8
8 Portar

`http://h10010.www1.hp.com/wwpc/il/en/sm/WF06b/`
`12883-12883-3445275-3445282-3445282-3963985-3963989.html?dnr=1`

B Programvaror

Debian 6.0 “Squeeze”
http://www.debian.org

vlan 1.9-3
http://packages.debian.org/squeeze/vlan

VirtualBox
https://www.virtualbox.org/

virtualbox-guest-additions 3.2.10-1
http://packages.debian.org/sv/squeeze/virtualbox-guest-additions

Backtrack 5 R2
http://www.backtrack-linux.org/

Nmap 5.51
http://nmap.org/

Yersinia 0.7.1
http://www.yersinia.net/

hping3 20051105
http://www.hping.org/

macof
http://www.irongeek.com/i.php?page=backtrack-3-man/macof (manual)

Ettercap 0.7.4.1-Lazarus
http://ettercap.sourceforge.net/

C Konfigurationsfiler

Hur filen /etc/network/interfaces har konfigurerats för de olika operativsystemen i varje system.

Switchkonfiguration 1

Grön dator system 1

```
################################################
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug eth0
iface eth0 inet static
   address 192.168.1.135
   gateway 192.168.1.129
   netmask 255.255.255.128
   network 192.168.1.128
   broadcast 192.168.1.255
################################################
```

Röd dator system 1

```
################################################
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug eth0

auto eth0.100
iface eth0.100 inet static
   address 192.168.1.10
   netmask 255.255.255.128
   network 192.168.1.0
   broadcast 192.168.1.127
   vlan_raw_device eth0

auto eht0.200
iface eth0.200 inet manual
   up ifconfig $IFACE 0.0.0.0 up
   vlan_raw_device eth0
################################################
```

Grön dator system 2

```
################################################
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug eth0
auto eth0
iface eth0 inet static
   address 192.168.1.136
   gateway 192.168.1.129
   netmask 255.255.255.128
   network 192.168.1.128
   broadcast 192.168.1.255
################################################
```

Röd dator system 2

```
################################################
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug eth0

auto eth0
iface eth0 inet static
   address 192.168.1.11
   gateway 192.168.1.1
   netmask 255.255.255.128
   network 192.168.1.0
   broadcast 192.168.1.127
################################################
```

Underliggande värdoperativsystem system 2

```
################################################
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug eth0

auto eth0.100
iface eth0.100 inet static
   up ifconfig $IFACE 0.0.0.0 up
   vlan_raw_device eth0

auto eth0.200
iface eth0.200 inet manual
   up ifconfig $IFACE 0.0.0.0 up
   vlan_raw_device eth0
################################################
```

Switchkonfiguration 2

Grön dator system 1

```
################################################
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug eth0
iface eth0 inet static
   address 192.168.1.135
   gateway 192.168.1.129
   netmask 255.255.255.128
   network 192.168.1.128
   broadcast 192.168.1.255
################################################
```

Röd dator system 1

```
################################################
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug eth0

auto eth0.100
iface eth0.100 inet static
   address 192.168.1.10
   netmask 255.255.255.128
   network 192.168.1.0
   broadcast 192.168.1.127
   vlan_raw_device eth0
################################################
```

Grön dator system 2

```
################################################
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug eth0
auto eth0
iface eth0 inet static
   address 192.168.1.136
   gateway 192.168.1.129
   netmask 255.255.255.128
   network 192.168.1.128
   broadcast 192.168.1.255
################################################
```

Röd dator system 2

```
################################################
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug eth0

auto eth0
iface eth0 inet static
   address 192.168.1.11
   gateway 192.168.1.1
   netmask 255.255.255.128
   network 192.168.1.0
   broadcast 192.168.1.127
################################################
```

Underliggande värdoperativsystem system 2

```
################################################
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug eth0

auto eth0.100
iface eth0.100 inet static
   up ifconfig $IFACE 0.0.0.0 up
   vlan_raw_device eth0
################################################
```