Detektion av IT-attacker

Övningsuppställning och insamlad data

KRISTOFFER LUNDHOLM, TEODOR SOMMESTAD,
MATS PERSSON, TOMMY GUSTAFSSON, AMUND HUNSTAD

FOI är en huvudsakligen uppdragsfinansierad myndighet under Försvarsdepartementet. Kärnverksamheten är forskning, metod- och teknikutveckling till nytta för försvar och säkerhet. Organisationen har cirka 1000 anställda varav ungefär 800 är forskare. Detta gör organisationen till Sveriges största forskningsinstitut. FOI ger kunderna tillgång till ledande expertis inom ett stort antal tillämpningsområden såsom säkerhetspolitiska studier och analyser inom försvar och säkerhet, bedömning av olika typer av hot, system för ledning och hantering av kriser, skydd mot och hantering av farliga ämnen, IT-säkerhet och nya sensorers möjligheter.

FOI
Totalförsvarets forskningsinstitut
Informationssystem
Box 1165
581 11 Linköping

Tel: 013-37 80 00
Fax: 013-37 81 00

www.foi.se

FOI-R-3342--SE
ISSN 1650-1942

Underlagsrapport
November 2011

Informationssystem

Kristoffer Lundholm, Teodor Sommestad,
Mats Persson, Tommy Gustafsson,
Amund Hunstad

Detektion av IT-attacker

Övningsuppställning och insamlad data

Titel	Detektion av IT-attacker – Övningsuppställning och insamlad data
Title	Detection of IT attacks – Exercise design and collected data
Rapportnr/Report no	FOI-R--3342--SE
Rapporttyp /Report Type	Underlagsrapport / Base data report
Månad/Month	November/November
Utgivningsår/Year	2011
Antal sidor/Pages	23 p
ISSN	ISSN 1650-1942
Kund/Customer	FM
Projektnr/Project no	E53195
Godkänd av/Approved by	Anders Törne

FOI, Totalförsvarets Forskningsinstitut	FOI, Swedish Defence Research Agency
Avdelningen för Informationssystem	Information Systems
Box 1165	Box 1165
581 11 Linköping	SE-581 11 Linköping

Sammanfattning

Under 2011 genomfördes inom projektet Spaning och motmedel på informationsarenan en övning i detektion av IT-attacker. Övningen syftade till att testa ett övervakningssystem som konstruerats vid FOI under året. Övervakningssystemet skapades genom att koppla samman en uppsättning open source-komponenter till ett enhetligt verktyg.

För att genomföra testet skapades en övningsmiljö bestående av ett stort nät. I detta placerades målsystem, vilka övervakades med detektionsverktyget, såväl som skenmål vars syfte var att generera bakgrundstrafik i nätet.

Nyckelord: Detektionsverktyg, IT-försvarsövning

Summary

An exercise concerning detection of IT attacks was performed within the scope of the project Spaning och motmedel på informationsarenan during 2011. The purpose of the exercise was to test a detection tool developed by FOI during the year. The detection tool was constructed by integrating several open source tools.

In order to perform the test of the tool, an exercise environment was created. The environment consisted of a network into which systems monitored by the detection tool, as well as systems meant to create background traffic was connected.

Keywords: Detection tool, IT-defense exercise

Innehållsförteckning

1 Inledning

Denna rapport beskriver uppställning av och insamlad data från en övning i detektion av IT-attacker som genomfördes under oktober 2011. Rapporten är producerad inom ramen för projektet Spaning och motmedel på informationsarenan. Projektet, som genomförs av Totalförsvarets forskningsinstitut (FOI), löper över 3 år och finansieras inom ramen för Försvarsmaktens Forskning och Teknikutveckling (FoT).

1.1 Syfte

Syftet med denna rapport är att ge läsaren en insikt i hur det inom projektet sammanställda övervakningssystemet fungerar, hur den övningsmiljö som användes för att testa övervakningssystemet är uppbyggd samt vilken data som samlades in under den genomförda övningen.

Rapporten kommer även att användas som underlag för den analys som ska göras av data och erfarenheter från den genomförda övningen. Analysen kommer att presenteras i en FOI-rapport i mars 2012.

1.2 Bidrag

Rapporten har följande tre huvudbidrag:

- En detaljerad beskrivning av den övningsmiljö i vilket övningen genomfördes.
- En detaljerad beskrivning av det av FOI byggda övervakningssystemet och dess ingående komponenter.
- En presentation av den datamängd som samlades in under övningen.

2 Övningsmiljö

Den miljö som övningen genomfördes i är en miniatyrmodell av Internet och bestod av nätinfrastruktur, målnät och skenmål. Nätinfrastrukturen bestod av ett antal routrar som hanterade trafiken mellan de olika näten i övningsmiljön. Målnäten innehöll de maskiner som övervakades under övningen. Det var dessa som angriparna skulle attackera. Skenmålen var till för att skapa realism åt övningen genom att utgöra andra system än målsystem i nätet och användes för att generera bakgrundstrafik.

2.1 Nätinfrastrukturen

Uppbyggnaden av nätinfrastrukturen baserades på ett antal fiktiva internetleverantörer (ISP) vars nät utväxlade trafik i en gemensam punkt. Inom varje ISPs nät användes ett dynamiskt routing-protokoll och mellan näten användes statisk routing. Skenmål och målnät var anslutna till respektive ISPs nät. En övergripande bild av övningsnätets uppbyggnad presenteras i Figur 1.

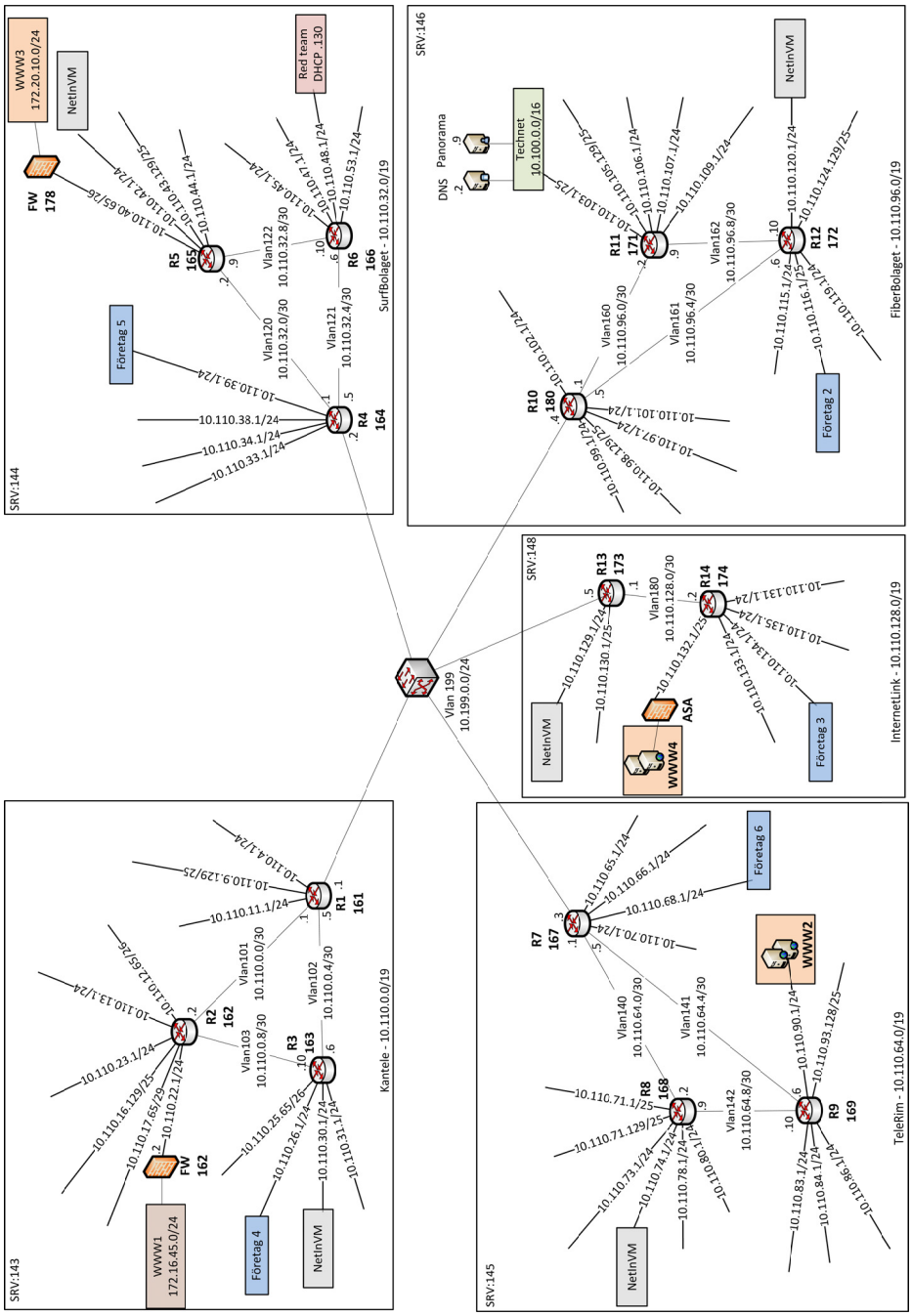
Nätinfrastrukturen byggdes upp av routermjukvaran Quagga på virtuella Ubuntu-servrar. Anledningen till att virtuella servrar valdes var att det möjliggjorde en isolering av övningsmiljön från de fysiska maskiner som övningen genomfördes på samt enklare återställning efter övningen. Inom varje ISPs nät användes routing-protokollet Border Gateway Protocol (BGP)för att dynamiskt utbyta routerinformation, medan det mellan dessa nät användes statisk routing. Detta medför att varje ISP i övningsmiljön har fått ett logiskt avgränsat nätsegment.

Utöver hur routrarna kopplats samman visas i Figur 1 även samtliga entiteter som fanns i nätet under övningen. Entiteterna beskrivs mer ingående i Tabell 1.

Tabell 1: Entiteter i nätet under övningen

Benämning i figuren	Beskrivning
Företag 2 – 6	Målnät som övervakas, se avsnitt 2.2 för beskrivningar av företagsnäten.
www1	Webbhotell med tre servrar varav en övervakades med Zabbix, se avsnitt 2.3 för en utförligare beskrivning.
www2	Webbservrar kopplade direkt till infrastrukturen, se avsnitt 2.3 för en utförligare beskrivning.
www3	Webbhotell, se avsnitt 2.3 för en utförligare beskrivning.
www4	Webbhotell bakom en hårdvarubrandvägg, se avsnitt 2.3 för en utförligare beskrivning.
Read team	Inkopplingspunkten för det röda laget.

Benämning i figuren	Beskrivning
NetInVM	Varje instans benämnd NetInVM är en uppsättning med 10 maskiner som simulerar ett mindre företagsnät. Se avsnitt 2.3 för en utförligare beskrivning.
Technet	Anslutningspunkt för Panorama till övningsnätet. All data från alla klienter ute i de övervakade näten skickade sina loggar hit. Technet innehöll även root DNS servern för övningsmiljön.



Figur 1: Nätinfrastrukturen i övningsnätet

2.2 Målnäten

I övningen fanns fem målnät vilka övervakades av övervakningssystemet (beskrivet i kapitel 3). Dessa var de primära målen för det attackerande laget.

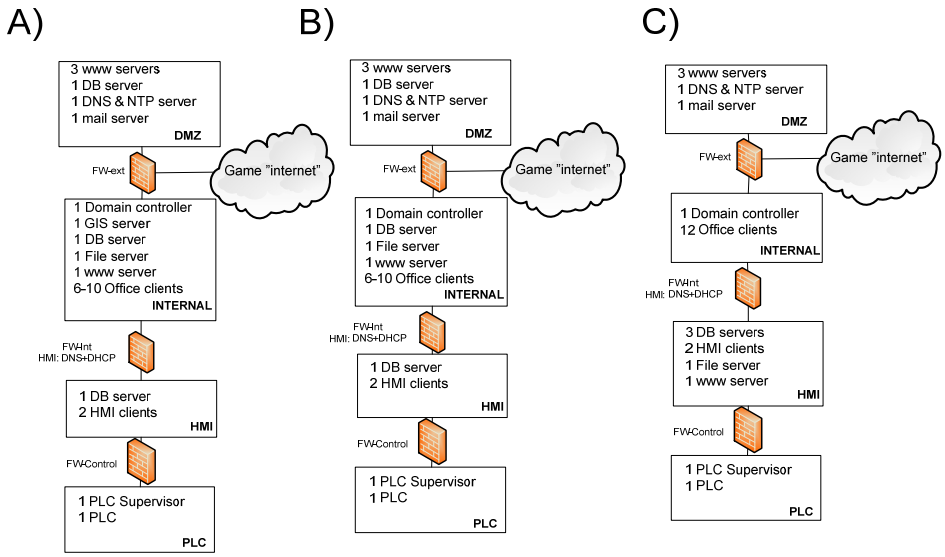
Att skapa en realistisk och omfattande IT-miljö för en övning är tidskrävande, just som det är när IT-miljöer skapas i andra fall. För att begränsa arbetsinsatsen för denna del av projektet användes redan existerande övningsnät som grund för de nät som skapades. De fem övningsnät som användes som grund var ursprungligen identiska i sin konstruktion då de tidigare hade använts vid en annan övning där flera attackerande lag övade parallellt. Det är även av denna anledning som näten är numrerade 2 – 6. För denna övning var dock variation önskvärd för att:

- Den producerade datamängden skulle täcka olika former av attacker och attacker på olika typer av system.
- Det attackerande (övande) laget skulle erbjudas utmaningar och tvingas variera sitt tillvägagångssätt. Om samma attack skulle fungera på samtliga målnät minskar motivationen för att prova andra attackformer.

Med anledning av detta modifierades de fem målnäten för att erbjuda variation. Ändringarna gjordes i nätens arkitektur, installerade mjukvaror samt implementerade skydd. Nedan beskrivs de fem näten utifrån dessa variabler.

2.2.1 Nätarkitektur och maskiner

Nätarkitekturerna ändrades förhållandevis lite. Det upplevdes som svårt att förändra näten i större omfattning utan att minska realismen i övningen – företags nätarkitekturer följer typiska mönster. I Figur 2 visas de arkitekturer som fanns tillsammans med de maskiner som var placerade i respektive nät.



Figur 2: Nätarkitekturer i målnäten. Målnät 2 hade en arkitektur som i A); målnät 5 hade en arkitektur som i C); resterande målnät hade en arkitektur som i B).

Brandväggarna var förhållandevis tillåtande. Från det externa nätet tilläts all trafik till ”DMZ”. Från alla interna nät tilläts trafik till det externa nätet, om den initierades från insidan. Mellan zonen ”DMZ” och zonen ”internal” tilläts all trafik som initierats från ”internal”, medan endast viss trafik tilläts om den initierats från ”DMZ”. I målnät 5 (benämnt C i Figur 2) tilläts all trafik som initierats från zonen ”internal” att gå till zonen ”HMI” medan det i de andra näten endast var viss trafik som tilläts passera brandväggen till ”HMI”. Utöver detta fanns det i målnät 2-5 en kontorsdator som hade dubbla nätverkskort och skapade en brygga mellan zonen ”internal” och ”HMI”.

2.2.2 Mjukvaror

De mjukvaror som var installerade på maskinerna i målnäten varierade, både mellan målnäten och inom målnäten. Variationen inom näten var främst mellan kontorsklienternas mjukvaror. I Tabell 2 ges en översikt över operativsystem och mjukvaror som användes i målnäten.

Tabell 2: Mjukvaror installerade på maskiner i övningen.

	Operativsystem	**Installerad mjukvara/tjänster**
Servrar	Debian 5.0 (lenny/stable) Debian 4.0 (etch/oldstable) Debian 4.0r1 (etch/oldstable) CentOS Windows 2008 Windows 2003, SP0-SP2 Windows 2000 server	Active Directory server, IIS, apache, telnet, SMB, netctat, rdp, vnc, MySQL, PHP, Wordpress, DotNetNuke, icecast, DNS, POP3, SMTP, HTTPS, NTP, IMAP
Klienter	Windows XP, SP0-SP3 Windows 7, SP0-SP1 Windows 2000 SP4	Adobe Acrobat Reader 7.09-9.4 Internet explorer 6-9 Flash 6.0.24-10.3 Java 5.18-6.26 Office 2003

På servrarna var mjukvarorna nästan identiska på de olika näten. Maskinerna som körde dessa mjukvaror hade dock uppdaterats på olika sätt. Det vill säga, det hade installerats olika patchar på alla maskiner. Som en följd fanns det olika sårbarheter i målnäten. Variation mellan klienterna skapades genom att slumpvis välja ut en uppsättning säkerhetsuppdateringar och applicera dessa.

På klientsidan installerades mjukvaror som förväntades vara meningsfulla givet klienternas användarmönster (se avsnitt 2.4 nedan). De mjukvaror som installerades var av olika versioner för att ge olika sårbarheter. Mjukvaran som skulle användas bestämdes genom att slumpa ett datum mellan 2005 och oktober 2011, och sedan installera den mjukvaruversion som släpptes närmast detta datum. I de fall denna procedur inte var tillämplig (exempelvis för Windows 7, som släpptes efter 2005) valdes versionen manuellt, men varierat.

2.2.3 Skydd

Förutom att variera mjukvaror (och därmed sårbarheter) mellan målnätens maskiner varierades också vilka operativsystemsskydd som installerats på dem. Skydden som varierades mellan var: minnesrandomisering (ASLR) och exekveringsskydd (NX-bit) – två motmedel mot utnyttjande av mjukvarusårbarheter. En jämn fördelning skapades mellan maskiner som hade ett av dessa skydd, båda skydden, eller inget. Då många av de fysiska maskiner som användes i övningen var äldre, saknades i de flesta fall hårdvarustöd för dessa skydd. Detta betyder att skydden i övningen i regel är simulerade i mjukvara.

Utöver detta fanns inga direkta skydd mot attacker implementerade. Till exempel fanns inget aktivt antivirusprogram installerat i målnätens maskiner.

2.3 Skenmål

För att skapa realism i övningen lades ett stort antal skenmål in i övningsmiljön. Syftet med dessa var att ha fler maskiner i övningsmiljön än bara målsystemen och för att generera bakgrundstrafik från. Skenmålen var av lite olika typ beroende på det syfte de skulle fylla. Gemensamt för majoriteten av skenmålen var att de innehöll sårbarheter som gjorde dem möjliga att ta över. Detta var avsiktligt för att det röda laget skulle kunna ta över en maskin i nätet och använda denna som utgångspunkt för ytterligare attacker.

2.3.1 Webbhotell

För att användarbotarna (se avsnitt 2.4) som installerats i företagsnäten skulle ha hemsidor att surfa till lades fyra webbhotell in i övningsmiljön. Webbhotellen såg lite olika ut och hade olika mycket skydd. Anledningen till att fyra webhotell lagts in i övningsnätet är för att skapa mer variation för vart i nätet användarbotarna skickade trafik.

Webbhotell 1 (www1) hade tre servrar som låg bakom en brandvägg med adressöversättning men utan filtreringsregler. En av servrarna tillhandahöll en för övningen viktig hemsida varpå det beslutades att denna skulle övervakas med Zabbix trots att servern inte ursprungligen var tänkt som ett målnät. På en annan av servrarna fanns en sökmotor som kunde användas för att göra sökningar i övningsnätet. Den sista servern tillhandahöll hemsidor för användarbotarna att surfa till.

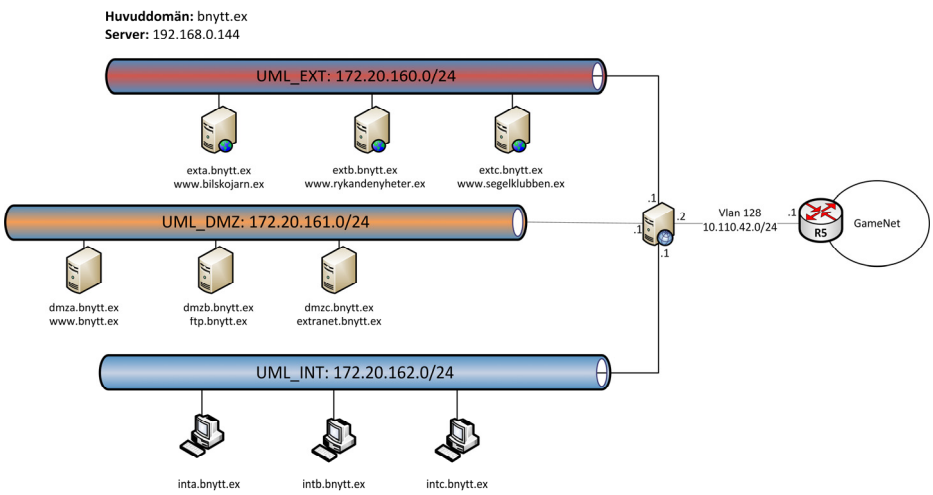
Webhotell 2 (www2) bestod endast av två webbservrar kopplade direkt till infrastrukturen utan något som helst skydd. På dessa fanns en uppsättning webbsidor för användarbotarna att surfa till.

Webbhotell 3 (www3) innehöll en webbserver bakom samma typ av brandvägg som webbhotell 1. Webbservern på detta webhotell tillhandahöll en uppsättning webbsidor för användarbotarna att surfa till.

Webhotell 4 (www4) bestod av två webbservrar som skyddades av en hårdvarubrandvägg (ASA i Figur 1). På en av dessa servrar låg en för övningen central hemsida på en väl skyddad server. Den andra webbservern bakom denna brandvägg var mycket dåligt skyddad och tillhandahöll webbsidor för användarbotarna att surfa till.

2.3.2 NetInVM

NetInVM är ett färdigt, virtualiserat, företagsnät uppbyggt i WMware. Ursprungligen var det tänkt att ett stort antal NetInVM nät skulle kopplas in i övningsmiljön och att dessa skulle ha användarbotar installerade. Då det visade sig att det uppstod upprepade problem med att använda NetInVM, av vilket det största var att klienterna inne i näten inte kunde använda DNS, kopplades endast fem instanser av NetInVM in i övningsmiljön som skenmål. Strukturen för dessa nät var densamma för alla instanser vilken illustreras i Figur 3.



Figur 3: Strukturen för NetInVM

2.4 Användarbotar

När övervakningsverktyg används i praktiken finns det ofarlig bakgrundstrafik som övervakningsverktyget måste kunna hantera. Denna ofarliga bakgrundstrafik kan till exempel vara kontorsanvändares vanliga surfande, e-post-skickande och filhantering. Det kan också vara externa parters aktivitet, som besökare på organisationens webbsida eller extern e-epost som anländer. I de flesta operationella scenarion utgör denna harmlösa trafik den absolut största delen av det totala trafikflödet. För att ett övervakningsverktyg ska vara till nytta måste det kunna sålla bort denna trafik. Även om bara en liten del av denna trafik resulterar i larm kommer falsklarm att bli vanliga, och ett utgöra ett problem.

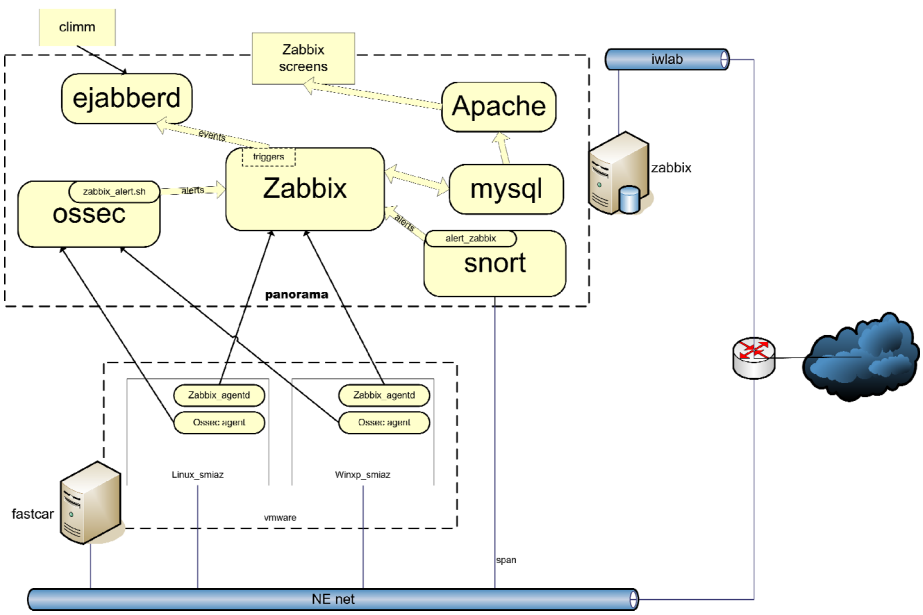
För att skapa en viss mån av realism installerades därför användarbotar på klientdatorer i målnäten. Botarna kördes på Windows-maskiner som var konfigurerade att automatiskt logga in en kontorsanvändare och direkt vid inloggning exekvera ett program som genererade knapptryckningar.

Knapptryckningarna var skapade för att slumpvis göra ett av följande och genomfördes stokastiskt var 1–2 minut:

- 35 % sannolikhet: Öppna Internet Explorer och surfa till en slumpvis vald hemsida i övningsmiljön. Väl på denna hemsida följs länkar slumpvis ett antal gånger.
- 15 % sannolikhet: Öppna en nyligen ändrad fil i mappen ”Mina dokument”.
- 10 % sannolikhet: Använda Outlook för att skicka e-post till en slumpvis vald mottagare övningsmiljön. E postens innehåll bestod av ett slumpat utklipp från en längre text.
- 10 % sannolikhet: Öppna Outlook och läs det senaste e-postmeddelandet som inkommit. Eventuella bilagor sparas i mappen ”Mina dokument”. Då botarna inte skickar bilagor till varandra är denna funktion till för att ge attackerarna en möjlighet att ta över system genom att skicka skadlig kod i bilagor.
- 10 % sannolikhet: Kopiera filer mellan ”Mina dokument” och en delad nätverksmapp.
- 20 % sannolikhet: Gör ingenting.

3 Övervakningssystemet Panorama

Panorama är ett verktyg för detektering av IT-attacker och har sammanställts inom ramen för projektet. Verktyget består av en uppsättning delkomponenter av vilka de mest centrala är Zabbix, Snort och OSSEC. Funktionen hos dessa förklaras mer ingående i avsnitten 3.1–3.3. Utöver de centrala komponenterna används även MySQL och Apache-server för presentation och lagring av data. Dessutom finns möjlighet att skicka ut meddelanden om larm via instant messaging med hjälp av ejabbered – en jabber-server. Som klient för mottagande av dessa meddelanden valdes Clim. Hur komponenterna är kopplade till varandra visas i Figur 4. Som synes är det Zabbix som är den centrala komponenten i Panorama. Zabbix samlar in data från agenter som installeras på alla maskiner som ska övervakas. Vidare skickas även data från OSSEC och Snort till Zabbix.



Figur 4: Översikt av Panorama med kopplingar till externa komponenter

3.1 Zabbix

Zabbix är ett verktyg för att samla in och presentera drift- och säkerhetsrelaterad information för IT-system. Verktyget kan hantera och sammanställa indata från flera olika källor. Zabbix fokuserar i första hand på realtidsövervakning vilket

framgår av hur realtidsdata kan presenteras i förhållande till historisk rapportering.

Den mest centrala komponenten i Zabbix är Zabbix-server. Det är den som tar emot data från alla Zabbix-klienter som finns installerade på de övervakade maskinerna. Zabbix-server tar även emot data från OSSEC och Snort vilket sker genom att data skickas från dessa via hjälpprogram. Servern lagrar de data som tas emot i en SQL-databas. Servern går även igenom alla triggers som finns lagrade i databasen och utför definierade åtgärder för de uttryck vars status anger att de ska utföras.

I Figur 4 finns förutom tidigare nämnda komponenter även Zabbix-screens vilket är ett webbgränssnitt mot Zabbix-server bestående av ett antal PHP-script. Dessa läser ut data ur SQL-databasen och skapar en grafisk bild av statusen för de övervakade systemen. Zabbix-screens använder Apache/PHP för att kommunicera med databasen.

3.2 Snort

Snort är ett Network Intrusion Detection System (NIDS), det vill säga ett nätbaserat intrångsdetekteringssystem. Snort lyssnar på den trafik som går i det nät som övervakas och genererar larm vid detekterade anomalier.

Snort använder sig av ett regelsystem för att matcha IP-paket mot regler. I en standardinstallation av Snort följer det med cirka 10000 regler. Reglerna är indelade i 5 prioritetsnivåer som anger hur allvarligt ett genererat larm ska anses vara. De nivåer som används av Snort är:

- 0 – “None” (prioritering saknas)
- 1 – ”High priority” (allvarliga angrepp)
- 2 – ”Medium priority” (försök till intrång)
- 3 – “Low priority” (portscanningar och dylikt)
- 4 – ”No priority” (informerar om udda paket)

I Panorama användes standardinstallationen av Snort med ett fåtal modifieringar. För det första ignorerades alla larm på nivå 0 och nivå 4 eftersom de inte direkt är säkerhetslarm. För det andra togs ett antal regler bort då de genererade väldigt många larm i övningsmiljön (även när inga angrepp gjordes) Nedan beskrivs vilka larm som togs bort följt av motsvarande Snort-regel:

- varningar om felaktiga tidstämplar på IP-paket:

```
alert ( msg: "STREAM5_BAD_TIMESTAMP";
sid: 4; gid: 129; rev: 1; metadata:
```

```
rule-type preproc ;
classtype:protocol-command-decode; )
```

- varning om för små IP-paket:

```
alert ( msg: "STREAM5_SMALL_SEGMENT";
sid: 12; gid: 129; rev: 1; metadata:
rule-type preproc ; classtype:bad-
unknown; )
```

- larm om Windows-datorer som försöker ansluta till tjänster på Windows-servrar:

```
:alert tcp $EXTERNAL_NET any ->
$HOME_NET [139,445] (msg:"NETBIOS
DCERPC NCACN-IP-TCP winreg OpenKey
overflow attempt";
flow:established,to_server;
dce_iface:338cd001-2244-31f1-aaaa-
900038001003; dce_opnum:15;
dce_stub_data;
byte_test:2,>,1024,20,relative;
content:"|05 00 00|"; metadata:policy
security-ips drop, service netbios-
ssn; reference:bugtraq,1331;
reference:cve,2000-0377;
reference:url,www.microsoft.com/techne
t/security/bulletin/ms00-040.mspx;
classtype:attempted-admin; sid:3218;
rev:12;)
```

- försök att ansluta till icke existerande servrar:

```
alert icmp $EXTERNAL_NET any ->
$HOME_NET any (msg:"ICMP Destination
Unreachable Host Unreachable";
icode:1; itype:3; classtype:misc-
activity; sid:399; rev:6;)
```

Som nämndes i avsnittet om Zabbix skickas alla larm som genereras av Snort vidare till Zabbix-server. För detta ändamål konstruerades en ny programmodul till Snort. När denna modul, kallad ”alert_zabbix” mottar ett larm matchas IP-numret mot ett hostnamn. Därefter skickas hostnamnet och larmmeddelandet från modulen till Zabbix-server.

Eftersom Snort kan generera många larm, vid till exempel portscanning, infördes en begränsning i den trigger som i sin tur genererar ett larm i Zabbix. Ett exempel på sådan trigger är

```
{alert[3].count(300)}>50
```

Vilket ska tolkas som ”har det kommit fler än 50 larm de senaste 300 sekunderna”. Motsvarande trigger för nivå 2 är ”har det kommit fler än 10 larm de senaste 600 sekunderna” och för nivå 1 ”har det kommit något larm de senaste 1200 sekunderna”. Konstanterna i dessa villkor går förstås att justera, och har lagts på en sådan nivå att de kommer en ”lagom” mängd larm. Tidskonstanterna i dessa triggers innebär också att larmen bara syns i 5, 10 respektive 20 minuter och sedan släcks de.

3.3 OSSEC

OSSEC är ett Host-based Intrusion Detection system, det vill säga ett intrångsdetekteringssystem som installeras på varje maskin det ska övervaka. OSSEC består av fyra komponenter som kan aktiveras på varje maskin som det körs på. Komponenterna tillhandahåller funktioner för integritetskontroll av filer, analys av systemloggar, detektion av rootkits/trojaner samt kontroll av att osäkra Windows-policys inte är aktiva. Vad som ska detekteras eller övervakas av dessa funktioner definieras i en konfigurationsfil som läggs in på varje övervakad maskin.

I Panorama används OSSEC genom att en OSSEC-klient installeras på varje maskin som ska övervakas. Denna klient skickar sedan vidare alla larm som genereras till en central OSSEC-server. Från OSSEC-servern skickas sedan larmen vidare till Zabbix-server med hjälp av ett script.

Vid installation av OSSEC följer det med en konfigurationsfil med standardinställningar. I denna konfigurationsfil listas bland annat vilka filkataloger som ska kontrolleras och var systemloggarna finns. I övningen gjordes inga betydande ändringar i konfigurationsfilen. Utöver standardinstallationen tillverkades ett script som OSSEC anropar vid varje larm. Detta script kopplar upp sig mot Zabbix-server och skickar hostnamn och larmmeddelande.

4 Övningen

Detta kapitel beskriver övningens genomförande. Analys av övningen och data från denna kommer att redovisas i en rapport under år 2012.

Övningen genomfördes under en vecka i oktober 2011, från måndag morgon klockan åtta till fredag morgon klockan åtta. Målsystemet befann sig i en avgränsad miljö vid FOI i Linköping och var under övningsveckan tillgängligt dygnet runt via en VPN-tunnel för det attackerande röda laget. Angripande trafik var dock koncentrerad till normal arbetstid. Det attackerande röda laget bestod av personal från IT-förbandet (ITF), en del av Försvarsmaktens telenät- och markteleförband (FMTM).

Målsystemet var inte skyddat av någon form av aktiva defensiva åtgärder; det fanns med andra ord inget försvarande blått lag som agerade mot det röda laget. Det röda laget var – om man bortser från en övervakande observatör – ensamma i sitt agerande relativt målsystemet.

Passiva defensiva åtgärder fanns dock i viss, men varierande, omfattning. Dessa defensiva åtgärder är beskrivna i avsnitt 2.2. Värt att notera är att brandväggskonfigurationerna för flertalet av näten ändrades under övningen som en del av attackerna. Händelseloggarna beskriver dessa förändringar.

Ytterligare realism och komplexitet kunde ha uppnåtts med aktiva defensiva åtgärder men målet med denna övning var avgränsat till att testa övervakningssystemet Panorama. För att kunna bedöma Panoramas förmåga som övervakningssystem behövdes därför tillräckligt realistiska testdata från en attackliknande situation. En sådan attackliknande situation bidrog det röda laget med genom att på olika och varierade sätt angripa målsystemet.

Konstruktören av Panorama observerade trafik och händelser i målsystemet. En del larm bedömdes av observatören som kritiska eller säkerhetsmässigt intressanta och noterades med tidpunkt och händelse för första observation. Från loggarna (se kapitel 5) går det även att ta fram när händelsen första gången registrerades av Panorama.

Utöver målsystemets och övervakningssystemets tekniska uppbyggnad var övningen även förberedd i form av ett scenario. Scenariot beskrev ett land styrd av en extraordinärt IT-positiv statsledning. Ur scenariots perspektiv var det röda lagets utgångspunkt att via sina attacker exempelvis blottlägga svagheter i landets IT-infrastruktur. Scenariot spelade dock en begränsad roll, det var endast en inledning till övningen. Övningens tyngd och fokus låg helt på det tekniska, relativt oberoende av scenariobeskrivningen och vad det röda laget representerade i scenariot.

5 Insamlad data

Detta kapitel beskriver den data som samlades in under övningen. Data som samlades in var:

- Aktivitetsloggar från deltagarna.
- All nättrafik i övningen.
- Operativsystemens loggar.
- Data som producerades av Panorama, (vilket inkluderar data från Zabbix, Snort och OSSEC).

Värt att notera är att ingen ingående kvalitetsgranskning har gjorts av data som samlats in. Till exempel har ingen kontroll gjorts av att fångad nättrafik och operativsystemsloggar är kompletta. Mer ingående kontroller kommer att göras längre fram i projektet, och eventuellt av externa forskargrupper som använder detta dataset.

5.1 Deltagarnas loggar

Den viktigaste och mest lättillgängliga data som samlats in är de loggar som deltagarna i övningen förde. Dessa fördes utifrån förbestämda mallar för att ge en tydlig bild av vad det attackerande laget gjorde vid olika tidpunkter och vad operatören av Panorama såg vid olika tidpunkter. Det attackerande lagets logg var uppdelad på avsökning (scanning) och attack, medan Panorama-operatörens logg bestod av fritext. Ett utklipp från dessa loggar visas i Figur 5.

A)

	Provide stating time (hh:mm and ending time if applicable (hh:mm)		Provide the machine from which the scan is performed	Provide the machine/port/IP-address or IP range which is scanned	Add comments here, for instance, if something goes wrong, flags used etc.
Day (yymmdd)	Time-start	Time-end	Performed from (machnin	Performed against machine/port/IP/IP-range/network	Notes
111011	08:13		.235	172.17.1.5,172.19.1.5	047. http://
111011	08:51		.150	10.100.0.0	048. Traceroute
111011	08:30	08:50	.235	172.21.1.0/24	049. nmap -A
111011	10:02		.235	172.21.1.4	050. Nessus webapp-scan
111011	09:25	10:03	.235	172.18.1.0/24	051. nmap -A
111011	10:20		.141	172.21.1.2,5,6	052. FTP
111011	10.37		.150	10.110.64.0/24	053. nmap -sn
111011	10:43		.150	10.110.64.10	054. nmap -A

B)

	Time of discovery (hh:mm)	Describe observed events. Also not if some something is changed in the network by you, e.g. if machines are rebooted.	Time logged (hh:mm)	Provide further details, e.g. CVE (Common Vulnerability Enumeration) tha is exploited or snort ID (cid)
Day	Time	Event	Log time	CVE or
2011-10-11	08:40	PING mail_6 from 10.110.48.235	08:36	
2011-10-11	08:41	Shellcode attacks vs DMZ 6	08:40	
2011-10-11	08:46	OSSEC reports nmap scans	08:43	
2011-10-11	09:01	nmap vs 172.21.1.0/24	08:43	
2011-10-11	09:02	Snort detect web attack vs office4_4	08:59	
2011-10-11	09:26	possible attack historian_3 from .235		
2011-10-11	09:32	nmap scans vs DMZ 3	08:59	
2011-10-11	10:02	probing TCP vs mail_3 from .235	09:34	
2011-10-11	10:03	traceroute portal_6 from .233	10:00	

Figur 5: Utklipp från deltagarnas loggar. Det attackerande lagets skanning-logg visas i A); det Panorama-operatörens logg visas i B).

Utöver detta har det attackerande laget sparat resultaten från sina aktiviteter i separata filer. Den sista kolumnen i Figur 5 A) pekar till exempel på den fil som producerats vid skanningen. Totalt gjorde det attackerande laget 102 skanningar och 17 attacker. Panorama-observatören noterade 96 händelser, varav ett antal är orelaterade till attacker (t.ex. noterades när observatören lämnade sin post).

5.2 Nättrafik

All trafik i övningsnätet passerade en central punkt. Från denna punkt extraherades paketdumpar på all nättrafik och sparades i form av PCAP-filer. Totalt sparades 238 gigabyte nättrafik. Mängden trafik som sparades skiljer sig signifikant mellan dagarna och varierar från 12 GB den första dagen till 148 GB den tredje dagen.

5.3 Operativsystemsloggar

Efter övningens slut extraherades de mappar som innehåller operativsystemens loggar. På Windowsmaskiner består dessa av evt-filer, på Linuxmaskiner består de av syslog-filer. Ambitionen innan övningen var att maskinerna skulle spara loggar av sådan storlek och tidsrymd att hela övningens aktivitet fångades. Detta har dock inte kontrollerats.

5.4 Nätövervakningsystemens loggar

Nätövervakningssystemen producerade loggar i samband med övningen. Dessa loggar inkluderar Panoramas logg, samt de loggar som producerats av Panoramas delkomponenter. Det vill säga, loggar producerade av Snort, Zabbix och OSSEC.