Kandidatuppsats
IT-Forensik och
Informationssäkerhet

RAM-minnets kontaminering vid tillämpning av forensiska verktyg

Christian Johansson, Robin Nilsson

Högskolan i Halmstad, IDE – Sektionen för Informationsvetenskap, Data- och Elektroteknik

Innehållsförteckning

Förkortningslista

ASCII	American Standard Code for Information Interchange - Används som teckenkodning för att förtydliga svårtolkad text.
Binärkod	Även kallat maskinkod som en dator klarar av att tolka och exekvera instruktioner med hjälp av ettor och nollor.
BIOS	Basic Input/Output System - Den första koden som körs på det körande systemet där primärmålet är att ladda och starta upp ett operativsystem.
DRAM	Dynamic Random-Access Memory - Långsammare minne som behövs uppdateras för att inte förlora sin information.
DMA	Direct Memory Access – Ett system som gör det möjligt att kontrollera minnet utan att processorn är involverad.
EEPROM	Electrically Erasable Programmable Read-Only Memory - För att omprogrammera minnet används högre elektriska spänningar.
EPROM	Erasable Programmable Read-Only Memory - Med hjälp av UV-ljus kan man radera innehållet på minneskretsen.
Flash-minne	Innehåller inga rörliga komponenter utan allt är i elektronisk form istället för mekaniskt.
Footprints	Spår som efterlämnas i datorsystemet som är unika och värdefulla i ett forensiskt syfte.
FSB	Front Side Bus – Kommunikation mellan nordbryggan och processorn.
Hibernera	Försätter systemet i icke-strömsatt läge och RAM-minnet skrivs ut till en fil som lagras på hårddisken.
MCC	Memory Controller Chip - Ett chip som fungerar som ett hjälpmedel för kommunikationen mellan processorn och RAM-minnet.

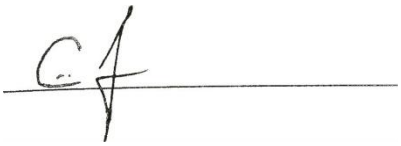
POST	Power-On-Self-Test - Ett test som genomförs när uppstart sker för att kontrollera så att hårdvaran fungerar på systemet.
PROM	Programmable Read-Only Memory - Minnet går endast att skrivas till en enda gång.
RAM	Random-Access Memory - Varje minnescell kan nås utan att söka igenom alla andra celler i minnesstrukturen.
ROM	Read-Only Memory - Minne som endast går att läsa data ifrån men tillåter inte skrivning.
SRAM	Static Random-Access Memory - Används som cacheminne i processorer då minnet utför instruktioner snabbt.
UNICODE	Universal standard som är designat för att datorer ska kunna behandla de flesta av världens skriftsystem.
.vmem	En .vmem-fil är en spegelbild av den virtuella maskinens RAM-minne. Det kommer bara att visas om den virtuella maskinen är igång, eller om den har kraschat.

Förord

Denna rapport är baserad på ett kandidatarbete inom IT-forensik och informationssäkerhet vid Halmstad Högskola, vårterminen 2011.

Vi vill ge vår handledare Mattias Weckstén ett stort tack för kontinuerligt stöd och engagemang för vårt arbete. På samma sätt vill vi även passa på att tacka våra klasskamrater som har visat intresse inom området och bidragit med diverse inlägg.

Halmstad, 2011-06-13



Christian Johansson

Robin Nilsson

Abstract

This report deals with a specific field of computer forensic and information security. As an involved part in an emergency situation, the knowledge of volatile memory can be decisive. As computer crime has increased dramatically, it has also contributed to a huge development in the forensic context. As a computer forensic it is important to recognize the impact on the incident system, therefore it is always good to have in mind that the memory always constantly changes, therefore, sought minimal changes to the system.

A computer system is equipped with a physical memory, whose purpose is to temporarily store information when it is active. This memory can be a rich source of information from a forensic perspective. Volatile memory is constructed of binary code that can be analyzed using tools. Today there are a wide variety of tools to the public and authorities, therefore, this report are focusing on commercial practices.

In order to streamline the practical performance, implementation of virtualization, in order to facilitates the detection of consistent data extraction. The experimentation of the volatile memory was obtained successful results in which it was possible to determine changes in memory allocation. When a program is executed, it results in noticeable changes that can be seen in the physical memory. When the tool has completed their given instructions, it is possible to interpret the percentage difference between the memory dump. This contributed to the predetermined goals.

The physical memory is a relatively new and unexplored territory of technology. The option for a memory extraction can include everything from hardware to software solutions, where the emphasis in this report is based on the software solution. All computer forensics should open their eyes to this information resource that can be the key to success in live-response.

Keywords: Forensic, live-response, virtualization, RAM-memory, extraction.

Sammanfattning

Denna rapport behandlar ett specifikt område inom IT-forensik och informationssäkerhet. Då en berörd part behöver agera i ett skarpt läge, kan kunskaperna om volatilt minne vara avgörande. I takt med att IT-brott har ökat dramatiskt, har det också bidragit till en enorm utveckling inom de forensiska ramarna. IT-forensikerns handlingar är av avgörande karaktär då minnet förändras kontinuerligt, därför eftersträvas minimala förändringar på systemet.

Ett datorsystem är utrustat med ett fysiskt minne vars syfte är att temporärt lagra information då det är aktivt. Detta minne kan vara en rik informationskälla ur ett forensiskt perspektiv. Volatilt minne är uppbyggt av binärkod som går att analysera med hjälp av verktyg. Idag finns det ett stort utbud av verktyg för allmänheten och myndigheter, därför begränsas rapporten med inriktning mot kommersiella metoder.

För att effektivisera det praktiska utförandet, tillämpas virtualisering som underlättar påvisning av konsistent datautvinning. Vid experimenterande av volatilt minne erhölls framgångsrika resultat där det gick att fastställa förändringar i minnesallokeringen. Då ett program exekveras, resulterar det i märkbara förändringar som går att urskilja i det fysiska minnet. När verktyget sedan har utfört sina givna instruktioner, går det att tolka den procentuella skillnaden mellan minnes-dumparna. Detta bidrog till de förutbestämda målen.

Det fysiska minnet är ett relativt nytt och outforskat tekniskt område. Valmöjligheten för en minnesutvinning kan innefatta allt från hårdvaru- till mjukvarulösningar där tyngden på denna rapport baseras på mjukvarulösning. Alla IT-forensiker bör få upp ögonen för denna informationstillgång som kan vara nyckeln till framgång under live-respons.

Nyckelord: IT-forensik, live-respons, virtualisering, RAM-minne, utvinning.

1 Inledning

Idag ser man allt fler system som måste vara tillgängliga dygnet runt, inte minst för företag. Detta medför att en IT-forensiker bör agera på plats, då man inte kan ta med datorn och samla in informationen i en sluten miljö.

För att kunna åstadkomma minnes-dumpar från volatilt minnet, krävs det olika typer av verktyg. Det första steget i vårt arbete är att studera vilka aktuella verktyg det finns ute på marknaden att tillgå. När vi har fått en övergripande bild över de olika verktygen, sätter vi igång med vår analys där vi studerar hur mycket minne programvaran använder när processen är aktiv. I ett tidigt skede [4.1 Noll-test] kommer det att utföras ett noll-test som fastställer hur RAM-minnet kontamineras över en given tidsperiod.

Vid varje tillfälle då vi använder processen vill vi fastställa om det är ett konsekvent resultat vi får fram för minnesdumpen eller om det visar sig att minneskapaciteten varierar. Det finns olika tillvägagångssätt att få ut minnesinformationen från systemet. Man kan antingen dumpa hela minnet eller inrikta utvinningen mot en specifik process. Delar av informationen som samlas in vid en hel minnes-dump kan vara överflödig, därför kan det vid vissa tillfällen vara lämpligt med inriktning mot en process för att begränsa informationen.

1.1 Syfte

Syftet med vårt examensarbete är att analysera RAM-dumpar från ett körande datorsystem. RAM-dumparna kommer att användas för att se hur RAM-minnet kontamineras vid användning av forensiska verktyg.

Arbetet ska ge inblick i hur viktigt det är att lagra relaterad data till korrekt lagringsmedia. Därför utförs en jämförelse mellan att lagra data på det interna mot det externa lagrings-mediet.

Vi vill även kunna skapa en verktygslåda med specialiserade verktyg som fungerar vid live-respons. Den ska vara enkel att tillämpa för att få ut den viktiga informationen från volatilt minne.

1.2 Mål

För att förtydliga målen har vi valt att dela in projektet i följande delmål:

- Hur mycket RAM-minne kontamineras vid exekvering av forensiska verktyg?

- Går det att fastställa minnesallokeringen av forensiska verktyg i RAM-dumpen?

- Vilken strategi lämpar sig bäst för olika typer av incidenter (lokalt- eller nätverksutvinning)?

1.3 Avgränsningar

Arbetet är inriktat mot Windows operativsystem. Inga undersökningar eller slutsatser kommer att utföras med avseende på operativsystemen Linux och Mac OS X.

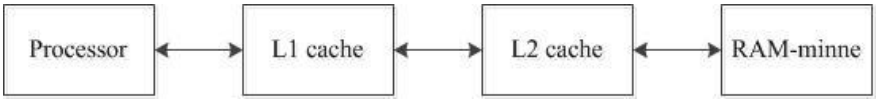
2 Bakgrund

För att få en översikt av projektet presenteras här den bakgrundsinformation som ligger till grund för att få en djupare förståelse hur datautvinningen genomförs av volatilt minne. Eftersom denna uppsats fokuserar på systemets primära minne, kommer vi att förklara skillnaden mellan ett volatilt och icke-volatilt minne. Då information finns kvar i minnet, använder Windows en temporär fil för att underlätta arbetet för minnet. Denna fil kan innehålla viktig data för att skapa en uppfattning av en incident. Vi kommer därför att skriva en kort introduktion om filens betydelse. För att få en tydligare bild över hur ett minne är konstruerat, kommer vi även att förtydliga detta och beskriva innebörden av kall- respektive varmstart. Slutligen behandlar vi när det lämpar sig att bruka live-respons vid en utvinning, då det idag används i större utsträckning för en IT-forensiker.

Vid en närmare inblick i tidigare fallstudier går det att relatera till en specifik rapport [1] där de tar upp liknande scenarion.

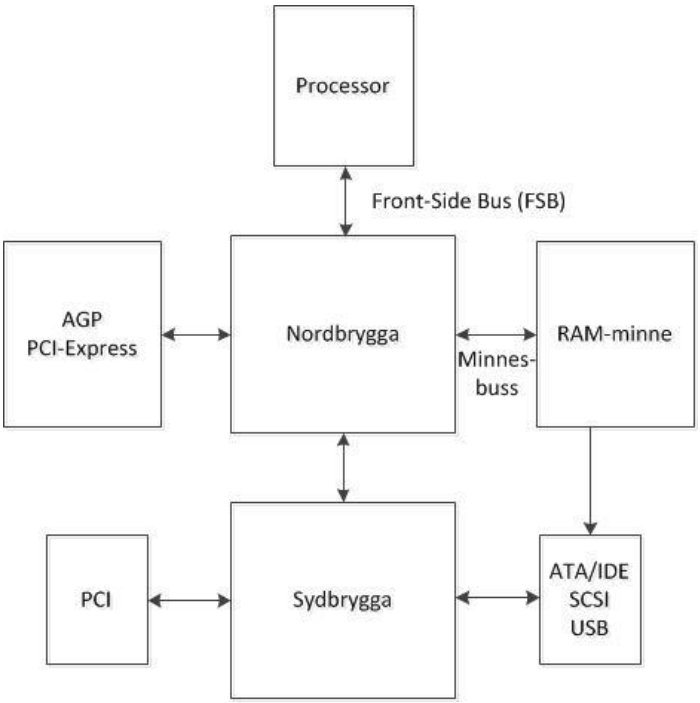
2.1 Översikt av minnets konstruktion

Nord- och sydbryggan är integrerade kretsar som sitter på moderkortet. Nordbryggan arbetar närmare processorn, därför är hastigheten snabbare. Den är dedikerad till RAM-minnet och grafikkortsanslutningar (AGP och PCI-express). Sydbryggan är däremot inriktad mot långsammare enheter i systemet vilket innefattar PCI-platser, SATA/IDE/SCSI och USB-portar. Minnesbussen används för att sammankoppla arbetsminnet till processorn. För att processorn inte ska bli överbelastad finns DMA som underlättar vid överföring av information. FSB är bussen mellan nordbryggan och processorn där den styr hur snabbt processorn kan kommunicera med systemet. I dagens processorer används en blandning av två arkitekturer: Harvard- och Von Neumann modellen. Harvard modellen användas närmast processorn som L1 cache och Von Neumann är de cacheminne L2 som samarbetar tillsammans med RAM-minnet, enligt följande:



Figur 2. Arkitektur mix av Harvard- och Von Neumann modellerna.

När processorn behöver specifik data, begär den informationen från minnesbussen. Processorn har inte kontroll på den fysiska platsen där RAM-minnet lagras. För att hjälpa processorn att hitta informationen, finns MCC som har koll på var data befinner sig i RAM-minnet.



Figur 3. Nord- och sydbryggans funktioner.

2.2 Icke-volatilt minne

Ett icke-volatilt minne är data som kan lagras utan strömförsörjning. För att kunna lagra information på ett system under en längre tidsperiod krävs det att man har ett icke-volatilt minne. Denna typ av minne används oftast som sekundärt lagringsmedia och de vanligaste formerna av icke-volatilt minne är ROM, PROM, EPROM, EEPROM och Flash-minne [2]. Enheter som beträffar icke-volatilt minne är hårddiskar, floppy diskar, CD-ROM och USB-minnen.

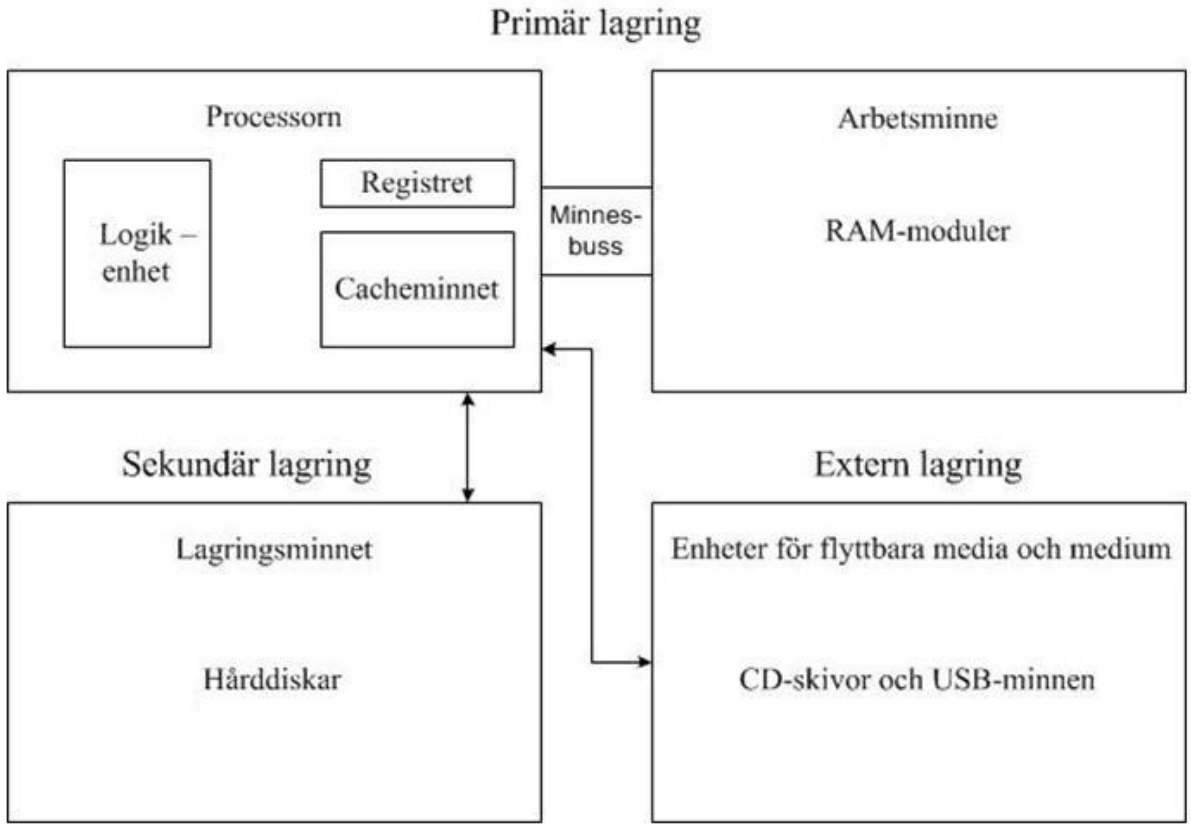
2.3 Volatilt minne

RAM betyder “Random Access Memory” som är ett primärt volatilt minne. Det används främst för att allokera ledigt utrymme till körande processer. Innebörden för “Random Access” är att slumpmässig data kan hämtas under tiden som systemet är aktivt. Volatilt minne är ett så kallat “flyktigt” minne då information upprätthålls med hjälp av en strömkälla. Detta innebär att data som lagrats i minnet endast kan hämtas från ett körande system.

RAM-minnet är systemets temporära arbetsutrymme som lagrar data, kod och inställningar. Volatilt minne lagrar information som ett elektroniskt kalkylblad bestående av rader och kolumner, där varje cell innehåller antingen en etta eller nolla. RAM-minnet sparar data i maskinkod som är ett lågnivå språk vilket systemet kan tolka.

För att skilja på volatilt minne i systemet finns det statiskt- och dynamiskt minne. Det statiska minnet förkortas som SRAM och dynamiskt minne för DRAM. Den största skillnaden mellan minnestyperna är informationshanteringen. För att DRAM ska kunna behålla sin information måste den uppdateras med regelbundna intervaller. Denna egenskap är inte behövlig av SRAM, då minnet förvaras tills strömkällan försvinner. Användningsområdet för typerna varierar då SRAM används som mellanlagring av information (cacheminne) och DRAM som det fysiska minnet i systemet (arbetsminnet). Som vi tidigare har nämnt, avtar minnet i RAM-cellerna med tiden. Kondensatorerna bidrar till att informationen kan hållas kvar i minnet under en kortare tidsperiod. Det som skiljer SRAM från DRAM är antalet transistorer som finns inuti minnet. DRAM består endast av en transistor som är sammankopplad till en kondensator. Kondensatorn kan anta två lägen: ”från” eller ”till” och detta motsvarar en etta eller nolla i cellen. När ett DRAM inte blir frekvent uppdaterat, läcker kondensatorn ut information. Det leder tillslut till att minnet återgår till ursprungsläget och tidsintervallet kan variera mellan olika minnen. Nyare minnen tenderar att avta betydligt snabbare än äldre minneskretsar [4].

Eftersom att cacheminnet arbetar nära systemets kärna (processorn) är minnet dyrare än arbetsminnet. Det är användbart att ha olika hastigheter för volatilt minne i systemet för att kunna påskynda processer. För att RAM-minnen ska fungera på ett korrekt sätt är det lämpligt att använda samma RAM-moduler. Genom att granska kapacitet, hastighet, tillverkare och den underliggande teknologin bidrar det till ett stabilare system [2].



Figur 1: Struktur över systemets lagringskomponenter.

2.4 Swap-filens betydelse

En swap-fil kallas också för “pagefile” på Windows operativsystem. Filen har i syfte att lagra temporär information på hårddisken [3]. Minnet är begränsat för hur mycket kod som det kan innehålla. När Windows transporterar data från RAM-minnet till ”pagefile”, kallas denna process för “disk thrashing” [2]. Det är ett tydligt tecken på att systemet har för lite RAM-minne installerat. Filen är dold i Windows och den används för virtuellt minne. Swap-filen brukar vanligtvis vara dubbelt så stor som det installerade RAM-minnet i systemet och den är dynamiskt växande. Standardinställningen för swap-filen är att den inte ska skrivas över när systemet stängs ner, dock går det att modifiera detta i registret. För en IT-forensiker kan det vara relevant att samla in data från swap-filen, då användaren inte tänker på att information som funnits i RAM-minnet kan återfinnas där.

2.5 Skillnaden mellan kall- och varmstart

Då utredaren utvinner data från ett volatilt minne, har kall- och varmstart en väsentlig roll. En kallstart är när systemet startar upp då det har varit avstängt [4]. Då systemet startas kan den skriva över delar av minnet då BIOS laddas in. Det finns också en inbyggd funktion i POST som kan nollställa minnet. Därför måste utredaren ha god insikt i vad som laddas in i RAM-minnet för att åstadkomma så lite åverkan som möjligt.

När systemet stängs ner är det ett flertal program som automatiskt tar bort känslig information ur volatilt minne. Utredaren kan då istället tillämpa en varmstart av systemet vilket ger ett litet utrymme för att minnet ska kunna förändras. Varmstart innebär att datorn är påslagen och utredaren startar endast om systemet [4].

2.6 Definition av live-respons

Dagens system utvecklas i en rasande takt och blir allt mer komplexa [5]. Det påverkar samhället, då människan blir beroende av tekniken för att infrastrukturen ska fungera. För företag och i den privata sektorn är det viktigt att kunna lita på datorsystemen. Live-respons har därför en betydelsefull roll då agerandet på körande system kan vara avgörande vid en incident [6]. För att kunna agera på ett strukturerat sätt när det gäller aktiva system måste utredaren ha en fungerande strategi.

Med hjälp av live-respons blir det möjligt att samla in volatilt minne som annars skulle gå förlorad [1]. Det kan vara avgörande bevismaterial för utredningen. Olika verktyg finns tillgängliga som gör det möjligt att utföra utvinning på systemet.

Det utredaren bör ha i åtanke är att allt som utförs på systemet påverkar volatilt minne. Denna princip kallas för “Locard’s Exchange Principal” och innebär att system alltid förändras med tiden.

När utredaren har samlat in volatilt minne, skapas en överblick av systemet vid den tidpunkt då utvinningen utfördes.

Det som kan vara av avgörande karaktär är om utredaren har kontrollen över verktygets inverkan på volatilt minne. Har utredaren den kunskapen går det att lätt urskilja på relaterade- och suspekta filer gällande fallet. Spår som lämnas av programvaran i minnet kallas för “footprints”. Dessa avtryck kan skriva över relevant information som ligger allokerat i minnet. Utredaren har alltid som mål att utföra så lite åverkan som möjligt på systemet.

Då en utredare ska utföra en live-respons finns det tre olika sätt att utföra det på. Dessa strategier baseras på hur fallet ser ut och vad som tillämpas i situationen. Det är en avvägning som utredaren får avgöra från fall till fall [7]:

- **Inledande live-respons**
  Innebär att utredaren endast plockar ut volatilt minne från mål-systemet.

- **Djupgående respons**
  Här tar utredaren så mycket data som krävs för att kunna välja en genomtänkt respons-strategi. Det kan gälla icke-volatilt minne som extraherats ut från systemet för att få en djupare förståelse om incidenten.

- **Full live-respons**
  Med hjälp av den här metoden tar utredaren ut fullständig data från live-systemet och utför en forensisk avbild av hårddisken. Denna metod kräver att systemet är avstängt.

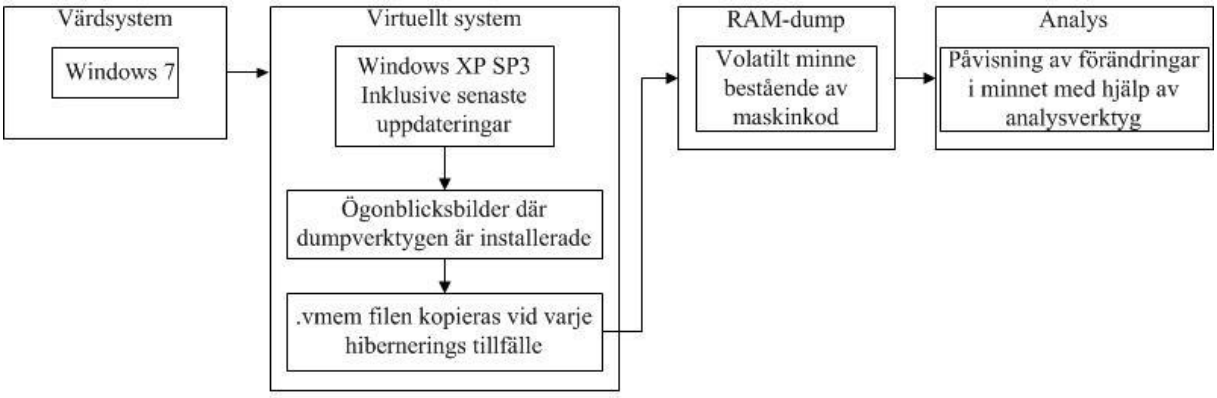
3 Metod

Arbetet är främst riktat mot primärmålet som är datautvinning av volatilt minne. Metoden som används är av undersökande karaktär då vi vill kunna påvisa förändringar i minnet. För att kunna analysera volatilt minne behöver utredaren veta vilken information som har en betydelsefull roll för fallet. Har utredaren den kunskapen, underlättar det tillämpningsmetoden då man filtrerar bort kända filer för att begränsa sökområdet. För att hitta relevant information i volatilt minne går det att utföra sträng-sökningar och att kontrollera signaturerna [8]. Det går att utföra grep-uttryck för att hitta specifik information i dump-filen såsom E-mail och IP-adresser [6]. För att hitta text i den binära strukturen som används för dump-filen kan en utredare använda verktyg som konverterar koden till ASCII eller UNICODE. Idag finns det programvaror som underlättar analyseringen då många funktioner är inbyggda i verktyget. Har utredaren tillräckligt avancerade kunskaper kan det vara lämpligt med automatiserade skript som påskyndar processen [9]. Vi har valt att utföra de praktiska momenten i en virtualiserad miljö. Metoden består av följande moment:

- Installation
- Utvinning
- Analysering
- Fastställning

3.1 Struktur för tillämpad metod

För att få en översikt av metoden vi använder oss av visas här en illustration av hur systemet är uppbyggt. Projektet går att dela in i moment där vi definierar de olika stegen under metodens gång.



Figur 4: Illustration över experiment-utförandet.

För att kunna utföra metoden på ett fulländat sätt [Figur 4] använder vi oss av ett värdsystem som klarar av att virtualisera ett operativsystem.

På värdsystemet applicerar vi Windows 7 som är det fysiska operativsystemet. I den virtuella miljön installeras Windows XP inklusive service pack 3 med de senaste uppdateringarna. Därefter skapas en ögonblicksbild som är utgångspunkten för alla experiment. Ögonblicksbilder gör det möjligt att återskapa systemet till ursprungsläget och man kan då på ett effektivt sätt arbeta inom varje sparad session.

För att få en uppfattning om hur ett system påverkas under en tidsperiod tillämpar vi ett noll-test. Sedan installeras varje programvara i varsin ögonblicksbild för att inte påverka varandras resultat. Metoden som vi eftersträvar är att generera tre stycken RAM-dumpar för varje verktyg som finns med i experimentdelen. För att åstadkomma detta hibernerar vi systemet vid tre tillfällen: direkt (dump I), startat (dump II) och använt (dump III). Är det inte möjligt att genomföra tre stycken RAM-dumpar, tillämpar vi en metod där vi jämför de två dumpar direkt (dump I) mot använt (dump III). Vid tillfället "direkt" menar vi att systemet endast har startats upp och inga förändringar har utförts. Innebörden av "startat" är när programvaran har exekverats och är redo för användning. Till sist skapar vi ett tillstånd som vi kallar "använt" som baseras på att programvaran har använts för sitt syfte. Vid de tre tillfällena skapar vi varsin dump (dump I, dump II och dump III) genom att hibernera det virtuella systemet. I hiberneringstillståndet skapar virtualiseringsprogramvaran en fil som innehåller volatilt minne. Filen .vmem kopieras vid varje hiberneringsläge och läggs på ett extern media.

Det flertalet RAM-dumpar som har skapats vill vi kunna analysera med hjälp av enkla metoder. Därför tillämpas en Linux Ubuntu (version 11.04) distribution i vårt syfte att jämföra filerna. I operativsystemet finns det integrerade hjälpmedel som är optimerade och har en välstrukturerad implementation.

Analysgenomförandet består av två integrerade verktyg: cmp och wc. Deras funktion är att jämföra filernas uppbyggnad mot varandra och ge ett mätvärde i antalet Bytes. Verktyget cmp har flera parametrar att tillgå men vi väljer att endast använda -l vilket ger oss värden för alla skiljande Bytes [10]. Det andra hjälpmedlet wc har också parametrar att tillämpa där vi väljer att applicera -l som skriver ut antalet rader [11]. Kommandot som används under Linux distributionen är enligt följande:

```
cmp -l X Y | wc -l >> Z
```

I formeln är konstanterna X och Y de dump-filer som ska jämföras. För att kunna se resultatet, vidarebefordrar vi utdatan till en tom textfil som här betecknas med konstanten Z.

Händelseförloppet för att få fram skillnader mellan dumparna kan beskrivas med den binära operationen XOR [12]. Denna operation innebär att den binära strukturen för filerna jämförs "bit för bit". Är mönstret som består av ettor och nollor samma för filerna så är de likvärdiga vilket ger en nolla [Tabell 1]. Däremot om binärkoden skiljer sig åt enligt tabellen nedan så ger detta en etta som indikerar på bit-skillnad.

X	Y	X ⊕ Y
0	0	0
0	1	1
1	0	1
1	1	0

Tabell 1. ⊕ =XOR "antingen eller" operation, där 0 =ingen förändring och 1 = förändring.

För att framkalla den procentuella skillnaden mellan dumparna använder vi en matematisk formel enligt:

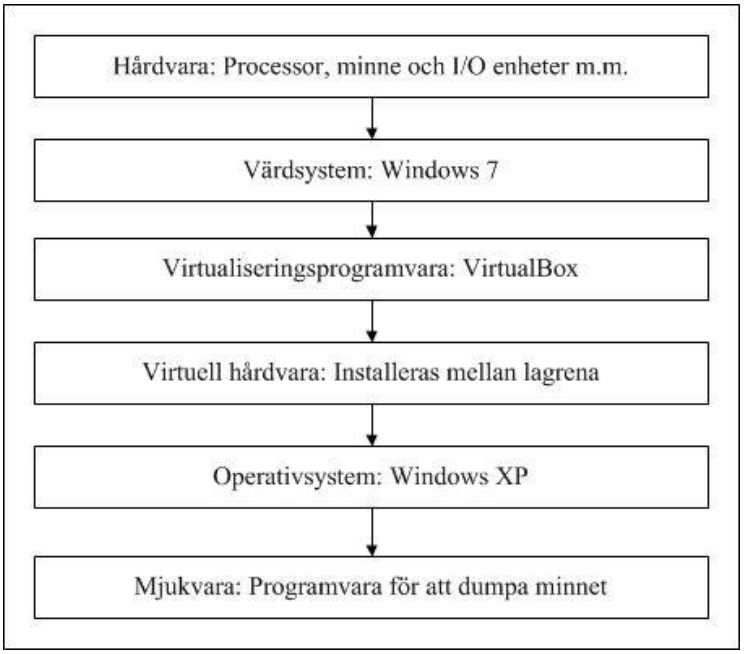
$$Z = \frac{(X \oplus Y)}{536870912} * 100$$

Uträkningen består av två dump-filer X och Y, där X innefattar det största värdet. Eftersom att RAM-minnet är avsatt till 512 MB i den virtuella miljön konverterar vi Megabyte till Bytes som blir 536870912. Detta utförs för att förenkla metoden då alla värden antar samma enhetsbeteckning. Då vi sedan vill åstadkomma en formatering till procent, multiplicerar vi beloppet med 100. Den procentuella skillnaden som har beräknats fram med hjälp av formeln kallar vi Z.

3.2 Virtuell miljö

Då dagens teknologi har blivit allt mer utvecklad, står vi nu i ett skede där virtualisering har blivit allt mer aktuellt. Många företag idag använder inte mer än 10-15 % av den datorkapacitet som finns inom företaget [13]. Därför bör flertalet företag få upp ögonen för tekniken och inom en snar framtid ser man säkert en större utsträckning inom området. Syftet med virtualisering är att skapa en miljö där en dator kan agera som ett flertal separata datorer. För att möjliggöra denna teknologi krävs det en fysisk dator bestående av hårdvarukomponenter som klarar av att driva ett virtuellt system. Ur ett ekonomiskt perspektiv blir det för företag mindre hårdvarukostnader, energiförbrukningen minskar, bättre översikt och utrymme frigörs i arbetslokalerna. Det som administratören ska tänka på när ett företag övergår till en virtualiserad miljö är att det fysiska systemet blir mer sårbart då det driver fler applikationer på ett enda system. "En virtuell infrastruktur förenklar hanteringen så att du får maximal nytta av dina IT-investeringar samtidigt som kostnaderna minskar." enligt [14].

För att göra det mer översiktligt med tillämpning av virtuell miljö har vi konstruerat en överskådlig illustration över de lager som virtualiseringen kräver.



Figur 5. Överblick av Virtualiseringslager som är baserad på Marshall et al.

Då vi kommer att använda oss av en virtualiseringsmiljö som är baserad på hårdvaran enligt vår [Bilaga A] är det ett system som klarar av att utföra våra krav. Värdsystemet är installerat med Windows 7 eftersom att det är det nyaste operativsystemet på marknaden och ska enligt referens [25] visa presterande resultat i testsammanhang mot andra Windows operativsystem.

Virtualiseringsprogramvaran vi använder oss av heter VMware Workstation som har möjlighet att skapa ögonblicksbilder där vi kan återskapa systemet till ett tidigare skede. Vi jämförde VMware Player mot VMware Workstation då vi insåg att vi var tvungna att kunna skapa ögonblicksbilder vilket endast Workstation kunde.

Då man vill tillämpa virtualisering måste ett virtualiseringslager installeras på värdsystemet. Infrastrukturen för det virtuella lagret är uppbyggt så att det befinner sig mellan hårdvaran och applikationslagret. Virtualiseringslagret använder hårdvaran hos värdsystemet till den virtuella miljön, därför måste de dela på resurserna.

Den virtuella hårdvaran som går att ställa in i programmet, utförde vi inga förändringar i, utan vi använde standardinställningar enligt [Bilaga A]. Vi valde att inte justera minnesstorleken som var avsatt till 512 Megabyte [Bilaga C] för Windows XP för att underlätta arbetet vid bearbetningen av RAM-dumpen. Det virtuella operativsystemet som användes är Windows XP då det är mest frekvent använt av alla Microsofts operativsystem[15].

Det som vi vill åstadkomma är att lämna ett så litet "footprint" som möjligt i minnet. Därför utför vi experimentet med ett flertal programvaror. Det ger oss en bättre överblick då vi i efterhand kan tolka information och besluta om vilket program som lämpar sig för live-respons.

3.3 Forensiska verktyg

De forensiska verktygen som har valts ut, är med hjälp av kurslitteratur [6] och andra programvaror som vi har varit i kontakt med under utbildningens gång. Att dumpa volatilt minne innebär att man tar ut minnesallokeringen till en fil som sparas på ett lämpligt lagringsmedium. Enligt [6] finns det ett flertal program som fungerar som verktyg vid insamling av volatilt minne. Som vi har nämnt tidigare så är det viktigt för en IT-forensiker att kunna fastställa hur verktyget påverkar minnesdumpen då man vill åstadkomma så lite åverkan som möjligt.

Vi har inriktat oss mot speciella verktyg för att dumpa minnet. Här följer därför en överskådlig förklaring av programvarorna vi tänker använda oss av:

DD

DD är en förkortning för "Data Dumper" och programvaran är till för att samla in en avbild av det fysiska minnet. Verktyget var främst utvecklat för UNIX-miljö men det har modifierats för att kunna köras på Windows 2000 och XP-system och kallas därför Win32dd. Det är ett kraftfullt verktyg som kan användas för att kopiera filer och skapa hela hårddiskavbildningar. Verktyget kräver fysisk tillgång till det lokala systemet för att kunna utföra RAM-dumpen. DD sparar informationen från minnet i .img format som kan lagras över nätverket eller på ett USB-minne [9].

Helix

Detta verktyg har en stor fördel för en IT-utredare, då det både kan samla in data från en systemavbild samt att den har inbyggda verktyg för att kunna analysera insamlad data. Det finns också tillgängligt som en forensisk Live-CD för Windows [16]. Denna CD effektiviserar utredningen då en applikation lägger sig på systemet och samlar in data från det körande Windows systemet.

Memoryze

Den här programvaran är gratis och det finns ett flertal exempel för hur man använder verktyget [17]. För att samla in minnet på enklaste sätt kan man exekvera en batch-fil så gör den det automatiskt. RAM-dumpen genereras då på den plats programvara ligger. Det finns många möjligheter för att anpassa hur Memoryze ska skapa dump-filen, dock görs detta via terminalen.

Nigilant32

Detta verktyg är likt DD och används främst för att samla in informationen från RAM-minnet. Programmet brukar ett mer användarvänligt gränssnitt och ett grafiskt utseende för att underlätta hanteringen för användaren [18]. För att utredaren lättare ska kunna ha med sig programvaran kan man välja att placera den på ett extern lagringsmedium. Det är möjligt att göra utvinning lokalt eller dumpa minnet via nätverket alternativt till ett USB-minne. I verktyget finns det även möjlighet till fjärranslutning till system och på så sätt utföra dumpningen.

OS-Forensics

Den här programvaran är fortfarande under utvecklingsstadiet men det verkar vara ett användbart verktyg. Det som en utredare kan ha nytta av med programvaran är att allt är samlat på ett och samma ställe. Med OS-Forensics går det att utforska, fastställa material och hantera information på ett snabbt och effektivt sätt. Då det inte krävs några externa verktyg utan allt är sammansatt i ett enda program. Det finns även en inbyggd funktion som gör det möjligt att kontrollera minnet direkt i verktyg [19].

FTK Imager Lite

Det här verktyget kan användas för att generera fram RAM-avbilder. Det sparas i .mem format och kan bearbetas i programvaran. Verktyget är enkelt och har ett lätt användargränssnitt. Det behöver inte installeras på värdsystemet utan det går att lagra programmet på en extern enhet och exekvera det därifrån.

För att granska information i programmet finns det även olika vyer vilket kan underlätta för utredaren [20]. En fördel med FTK Imager Lite är att den kan importera hiberneringsfilen från ett virtuellt system direkt in i programvaran.

Nedanför följer en tabell över verktygens fysiska storlek i Bytes:

Programvaror	Exekveringsfilen (innan installation)	Programvarans mapp (efter installation)
Win32dd från Helix live-CD	106312	*
Helix live-CD v.2.0	735844352	*
Memoryze v.1.4.41.0	8059904	8485136
Nigliant32 beta v.0.1	774144	*

OS-Forensics beta v.0.93	37426488	56072987
FTK Imager Lite v.2.9.0.1385	20339408	59945736

Tabell 2. Tabell över verktygens fysiska storlek. ** = installeras ej.*

3.4 Incident respons

En incident har aldrig samma typ av karaktär som ett tidigare fall. När ett datorsystem är igång så påverkas det alltid av den princip [6] (Locard's Exchange Principle) som vi tidigare tog upp.

Brottsplatsen kan variera mellan olika incidenten därför är det lämpligt att ha flera strategier för utredaren. I särskilda fall kan det vara passande att agera kvickt och snabbtänkt för att spår inte ska ödeläggas. Många faktorer kan påverka resultatet som i slutändan ska mynna ut i en rapport för IT-forensikern. Utredaren måste ha understöd för det som hittas och påvisning om vart informationen härstammar ifrån ska kunna kopplas till fallet. Många typer av brott innefattar en husrannsakan där datorn är inblandad [21] och den beslagtas med all sannolikhet. I detta fall har utredaren mer tid på sig då informationen kan behandlas på ett annat sätt. Här utgår man från ett avstängt system och datorsystemet kopplas upp i en forensisk labbmiljö. I detta sammanhang har volatilt minne en mindre betydelse för fallet då information försvinner vid avstängning av datorn.

Lokal-utvinning kan komma till nytta då incidenten har en sådan karaktär att utredaren kan få fysisk tillgång till det berörda datorsystemet. Återigen måste utredaren ha en grundlig struktur för de material som efterfrågas vid det enskilda fallet. Det kan röra sig om att identifiera eller verifiera källan till ett visst brott då man vill skapa "chain of custody" som innebär hela beviskedjans fastställande. Vid en lokal-utvinning måste systemet vara aktivt och utredaren bör nödvändigtvis ha kännedom om användarens inloggningsuppgifter. När IT-forensikern väl har kommit in i systemet är utgångspunkten att lämna så lite spår som möjligt efter sig. Verktygslådan ska vara utformad på ett sådant sätt att den går att tillämpa på ett effektivt sätt för den situationen. Då systemet är igång, vill utredaren dumpa minnet [7] för att få ut lösenord som finns sparat i klartext. Det kan även vara filer som har varit öppnade vid ett tidigare tillfälle, information om körande processer och nätverksrelaterade anslutningar [Bilaga B].

Nätverksutvinning kan tillämpas då en utredare behöver göra datautvinningen på ett system i det lokala nätverket. För att åstadkomma utvinningen krävs det ett system som är sammankopplat till det specifika värdsystemet. På samma sätt som vid en lokal-utvinning krävs det inloggningsuppgifter till målsystemet. Det kan vara lämpligt att använda metoden då utredaren vill övervaka nätverkets aktiviteter [7].

Processen kan bli något långsammare än att använda en lokal-utvinning eftersom att det sker via nätverket och det kan påverka tiden det tar för utvinningen. När en nätverksutvinning utförs är det viktigt att alla konfigurationer är rätt inställda så att informationen skickas till korrekt destination inom nätverket.

När en berörd part tillämpar virtualisering av ett operativsystem kan system försättas i hibernering. Detta medför att man kan återuppta systemläget från det vilande tillståndet. Det som inträffar när systemet hiberneras är att volatilt minne komprimeras och skrivs till en fil på hårddisken som kallas hiberfil.sys [6]. Den fil som skapas vid hiberneringstillfället innehåller inte det exakta tillståndet för det körande systemet. Hiberneringsläge finns även på stationära datorer såväl som i bärbara datorer där det ofta används som standardinställning. Det aktiveras efter en tid då användaren inte har brukat systemet. Denna metod kan vara användbar för företag där implementation av virtualisering dominerar. Då deras system måste vara tillgängliga för användaren, anses det vara oacceptabelt att avbryta processen under en längre tidsperiod.

Efter att live-utvinningen är färdigställd gör utredaren en bedömning om hur systemet ska hanteras utifrån det operativsystem som är installerat. I många fall drar man ur strömkontakten från chassit för att undvika skrivning till hårddisken [7]. Att kunna skapa en tidslinje över fallet är ett mål som eftersträvas då det kan hjälpa utredaren att fokusera på det som är väsentligt. Under hela utredningsförloppet är det ett måste för IT-forensikern att dokumentera allt som tillhör brottsutredningen.

Fotografering av brottsplatsen är en detalj som inte bör glömmas bort då man vill kunna återskapa platsen igen till det skick som var från början. Markeringar över de datorkontakter som varit berörda med chassit måste utmärkas för att återställningen ska kunna verkställas. För att bevismaterial inte ska skadas är det lämpligt att placera hårdvara i anti-stat påsar för att bevara dess innehåll.

Då ett IT-relaterat brott har inträffat måste en husrannsakan fastställas för att IT-forensikern ska få tillträde till det berörda systemet. Ur ett juridiskt perspektiv finns det endast två renodlade databrott. I Sveriges lagbok är brotten med under brottsbalken som innefattar dataintrång BrB 4:9c och bedrägeri BrB 9:1. Internet är en källa av information som är tillgänglig över hela världen. Kommunikationen sker i digital form och datoranvändare kan åstadkomma att en effekt inträffar på ett ställe (effektort) då personen i själva verket befinner sig på en annan plats (händelseort). Jurisdiktion är därför ofta inblandat i databrott då brotten kan begås vart som helst på jordklotet. Alla världens länder har egna föreskrivna lagar, därför har effektort och händelseort en avgörande betydelse för brottet [21].

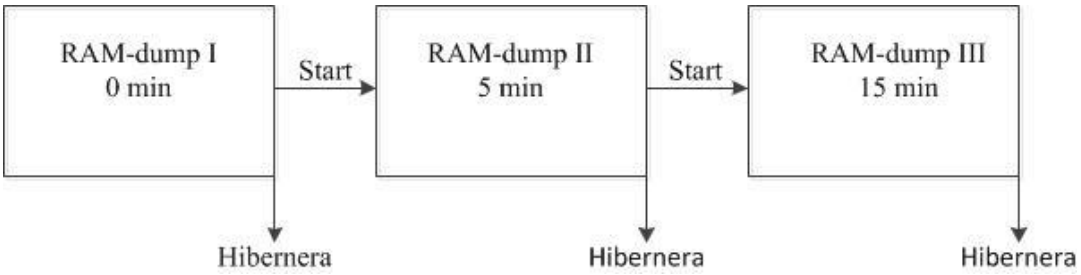
4 Experiment

Det tillvägagångssätt som valts är att använda mjukvara för att konsekvent kunna påvisa förändringar i volatilt minne. Därför tillämpas virtualisering med hjälp av VMware workstation där det finns möjlighet att skapa ögonblicksbilder. Virtualiseringsverktyget gör det möjligt att hibernera systemet där volatilt minne automatiskt skrivs ut till en specificerad fil (.vmem). Operativsystemet som används är Windows XP inklusive service pack 3 och de senaste uppdateringarna. Anslutningen till Internet är delat mellan värdsystemet och den virtuella miljön. Vi har gjort detta val då det oftast förekommer vid en incident. RAM-minnets storlek är alltid avsatt till 512 MB (536870912 Bytes) för det virtuella systemet.

4.1 Noll-test

Då vi vill kunna fastställa hur volatilt minne påverkas under en viss tidsperiod krävs det att systemet inte används, men ändå är aktivt. Detta medför kunskaper om hur ”Locard´s Exchange Principle” påverkar vårt experimentsystem. Därför kallar vi detta experiment för noll-test som är ett tidstest, vilket ligger till grund för att bilda en uppfattning om systemets förändringar över en tidsperiod.

Experimentet som utförs är indelat i tre moment där vi skapar tre RAM-dumpar för varje tidsram. [Figur 6] visar tillvägagångssättet för hur vårt experiment är indelat. Det första steget är att generera den första RAM-dumpen genom att hibernera systemet direkt och kopiera filen .vmem till en extern hårddisk. Därefter startas det virtuella systemet upp igen. För att få fram den andra RAM-dumpen väntar vi i fem minuter och hibernerar systemet ytterligare en gång. När vi har väntat på den angivna tiden förändras .vmem-filen och innehåller RAM-minnets nuvarande tillstånd, därför kopierar vi den ytterligare en gång. Då vi vill ha flera mätvärden, skapar vi ännu en dump på samma sätt som föregående men väntar i tio minuter innan vi hibernerar systemet. Därefter sparar vi undan RAM-minnesfilen till ett externt medie.



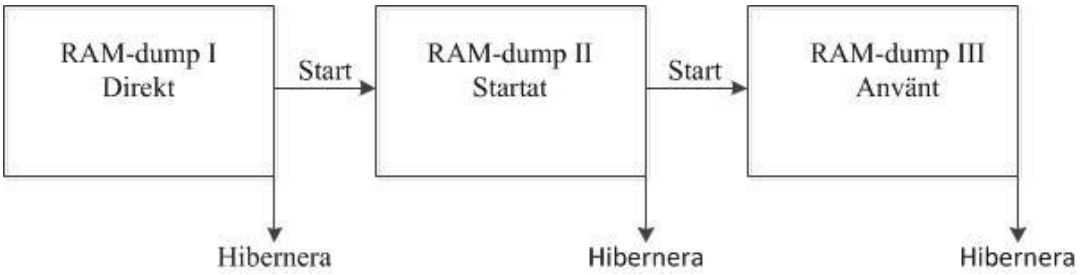
Figur 6. Illustration över RAM-dumpens tidsram.

Efter att ha utfört hiberneringarna och sparat undan de tre genererade .vmem filerna, har vi då fått fram materialet för analysering av noll-testet.

4.2 Utvinning av volatilt minne med forensiska verktyg

Ur ett forensiskt perspektiv kan det vara fördelaktigt att veta hur mycket minne ett specifikt verktyg använder vid en live-respons. För att klargöra detta kommer vi att utföra experiment på forensiska verktyg. De programvaror som kommer att ingå i vårt experiment är FTK Imager Lite, OS-Forensics, Helix, Win32dd, Memoryze och Nigilant32. Den gemensamma faktorn som de forensiska verktygen har, är möjligheten att utvinna volatilt minne med understöd av programmet. Därför vill vi kontrollera hur minnet kontamineras vid användning av programmet. Genom att specificera var RAM-dumpen ska placeras (fysiskt eller externt), går det att dra en parallell för hur mycket volatilt minne som påverkas när dumpen placeras på de två lagringsenheterna. Vi kommer därför att följa upp ett verktyg som vi anser påverkar systemet i markant bemärkelse då RAM-dumpen placeras på olika lagringsmedium. Därefter jämförs .vmem-filen mot den genererade RAM-dumpen som återfinns i den virtualiserade miljön vilket möjliggörs genom att kopiera dumpen till värdsystemet.

Följande figur ger en lättöverskådlig överblick i hur RAM-dumparna skapas:



Figur 7. *Illustration över de tre stegen (Direkt, Startat och Använt) där RAM-dumparna skapas.*

Fristående verktyg

De fristående verktyg som vi har valt att fokusera närmare på är FTK Imager Lite, Nigilant32, Win32dd och Helix. Programvarorna behöver inte vara installerade på systemet utan kan exekveras från ett externt media.

När vi ska genomföra experimenten använder vi den metod [Figur 7] som vi tidigare har nämnt i rapporten. Helix, Win32dd och Nigilant32 exekveras från en live-CD. FTK Imager Lites installationsmapp kopieras från en extern enhet (USB-minne) och läggs in på det virtuella systemet.

För att genomföra det första verktygsexperimentet aktiverar vi ögonblicksbilden som innehåller FTK Imager Lite. Därefter startar vi upp den virtuella miljön och hibernerar systemet (.vmem). Det bidrar till den första dump-filen (Direkt) för programmet som vi kopierar från VMware katalogen till en extern enhet. Efter det startas den virtuella maskinen och FTK Imager Lite exekveras. Sedan hiberneras (.vmem) det virtuella systemet som ger den andra dumpen (Startat). Den kopieras och sparas innan vi återigen startar upp det virtuella systemet. Därefter exekveras programvaran och används genom att klicka på: File → Capture Memory (placerar dump-filen på skrivbordet i den virtuella miljön). När processen är färdig hiberneras systemet (.vmem) ytterligare en gång och RAM-dumpen (Använt) kopieras.

I Nigilant32 börjar vi med att öppna upp ögonblicksbilden i VMware. När det virtuella systemet har laddat klart hiberneras systemet (.vmem). Därefter kopierar vi dump-filen (Direkt) för den första RAM-dumpen och lägger den på ett externt media. Sedan återupptas det virtuella systemet igen och verktyget Nigilant32 exekveras. När programmet har laddats klart, hiberneras systemet (.vmem) och den andra dump-filen (Startat) skapas. Vi kopierar dump-filen och lägger den på ett externt media. Återigen återupptas det virtuella systemet och programvaran används genom att klicka på: Tools → Image Physical Memory → Programmet startar och filen genereras till skrivbordet. Vi hibernerar systemet (.vmem) och kopierar den sista dump-filen (Använt). Därefter placeras filen på ett externt media.

När vi ska experimentera med verktyget Win32dd tillämpar vi ögonblicksbilden för programmet. Därefter monteras CD-skivan in i den virtuella maskinen. Nästa steg är att starta upp VMware och sedan hibernera systemet (.vmem). Detta medför att en första dump-fil (Direkt) skapas och vi kopierar denna till vår externa enhet. Steg två är att starta upp den virtuella maskinen och exekvera Win32dd. I verktyget väljer att klicka på: Live Acquisition → Pekar mot skrivbordet med namnet ”image.dd” → Trycker på Require. Slutligen hiberneras (.vmem) den virtuella maskinen och dumpen kopieras (Använt) för att sedan läggas på det externa mediet.

Helix har ett liknande utförande som Win32dd där det genomförs i två steg. Det första steget är att montera skivan som innehåller Helix programvara. Sedan startar vi upp det virtuella operativsystemet för att sedan hibernera det (.vmem). Detta ger möjlighet till den första dumpen (Direkt) som ska användas vid analyseringen. Andra steget är att starta VMware igen och aktivera live-skivan för att sedan använda det inbyggda verktyget i Helix för att se alla aktiva processer som körs på systemet. Därefter hiberneras systemet (.vmem) och den slutgiltiga dump-filen (Använt) kopieras för vårt experiment med Helix.

Installationsverktyg

De programvaror som ingår som installationsverktyg i experimentdelen är OS-Forensics och Memoryze. För att kunna använda dessa verktyg krävs en installation på värdsystemet [Bilaga A] innan det går att fastställa kontamineringen av RAM-minnet.

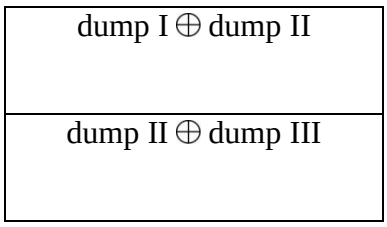
När vi ska experimentera med OS-Forensics går vi in i programmets ögonblicksbild. När systemet laddat färdigt hibernerar (.vmem) vi den virtuella miljön. Detta bidrar till den första dump-filen (Direkt) som vi kopierar och lägger till ett extern media. Efter detta återupptar vi processen för det virtuella systemet och exekverar OS-Forensics. Sedan hiberneras systemet (.vmem) och filen kopieras till ett externt lagringsmedia. Det ger den andra dump-filen (Startat) för experimentet med verktyget. Därefter startas maskinen upp igen och programmet används genom att klicka på: Memory Viewer → Väljer ”Dump Physical Memory Contents” → Därefter klickar vi på ”Dump” → Dump-filen döps till ”memory” i formatet .bin och placeras på skrivbordet. När allt detta har genomförts hiberneras systemet (.vmem) en sista gång. Det ger den sista dump-filen (Använt) och kopieras till ett externt media.

Det andra verktyget som vi använder som installationsverktyg är Memoryze. För att använda programmet går vi in i ögonblicksbilden för det angivna verktyget. Det första som genomförs är att vänta tills det virtuella systemet har laddat färdigt för att sedan hibernera (.vmem) miljön. Det skapar en dump-fil (Direkt) för hur systemet ser ut från början innan något har exekverats på systemet. Återigen återupptas den virtuella miljön. När man använder Memoryze finns det inget start-läge för programmet då det körs automatiskt vid exekvering av en .bat-fil. För att köra programvaran tillämpas följande tillvägagångssätt: Kör igång filen MemoryDD → Programvaran skapar en fil i ”Audit” mappen. Detta bidrar till den sista dumpen (Använt) då systemet väl har försatts i hiberneringsläge (.vmem).

4.3 Analys

Under detta avsnitt analyseras de experiment som har utförts rent praktiskt.

I tre av sex fall med programvarorna FTK Imager Lite, Nigilant32 och OS-Forensics har vi lyckats skapa tre stycken dump-filer för varje experiment. Dessa jämför vi med den metod som vi tidigare angett. Förändringarna i de tre dumparna konstateras genom att ställa upp filer mot varandra på följande vis:



Figur 8. Illustration över hur de tre dumparna jämförs mot varandra.

I de tre återstående experimenten som innefattade Helix, Win32dd och Memoryze bestod analysen av två dump-filer. Orsaken till att det blev färre RAM-dumpar berodde på att programvarorna inte behövdes startas upp innan de exekverades. Helix och Win32dd kördes via en live-CD och Memoryze exekverades från en .bat-fil där den automatiskt skapade en avbild.

Avbilderna som framkallas med live-CD:n sparades lokalt på skrivbordet och Memoryze lagrade sin dump i den folder som programvaran befann sig i. Jämförelsen för de två dumparna blev då på följande sätt:

dump I ⊕ dump III

Figur 9. Illustration över hur de två dumparna jämförs mot varandra.

För att få ut skillnaderna i dumparna använder vi den metod som tidigare angivits under metod-delen [3.1 Struktur för tillämpad metod].

Med det populära forensiska verktyget Win32dd valde vi att ytterligare experimentera för att se skillnader i minnesstrukturen vid lagring av dump-filen. Vid de sex genomförandena placerade vi dump-filen av programvarorna på den lokala hårddisken. Därför ansåg vi det passande att se förändringar i minnesstrukturen då IT-utredaren väljer att placera utdatat på ett extern lagringsmedia. Vi antydde också att en jämförelse mellan den genererade RAM-dumpen som programvaran själv skapat skulle sättas i relation till de andra dump-filerna.

4.4 Resultat

Det som går att fastställa vid den första RAM-dumpen för noll-testet är att inga förändringar har inträffat på systemet. Efter 5 minuter kan man se en förändring (5225027 Bytes) i RAM-minnesstrukturen då den procentuella enheten är 0,97 %, vilket i sin tur påvisar att minnet har förändrats. I tabellen [Tabell 3] går det också att utläsa hur minnet har påverkats efter 15 minuter. Procentenheten har då blivit 1,18 % där 6332999 Bytes förändras av de totala 536870912 Bytes i RAM-minnet.

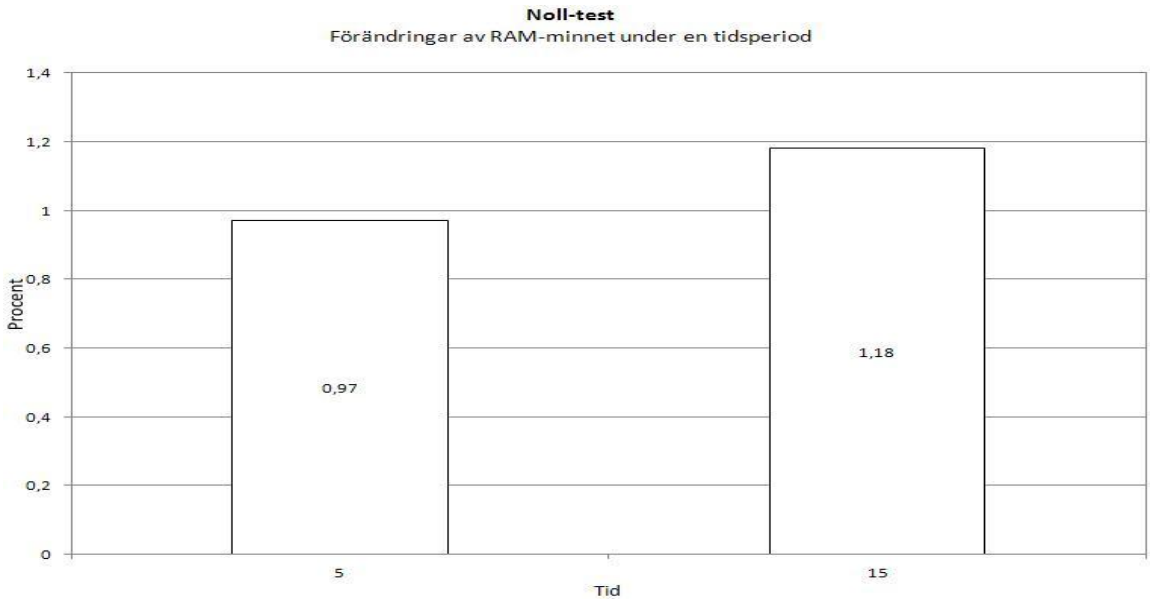


Diagram 1. Diagram över noll-testet med utgångspunkt från 0 - 1,4 %.

Noll-test har utförts under tre tidsintervaller som illustreras i första kolumnen enligt [Tabell 3]. Sista kolumnen anger procentuell förändring i Bytes som förblir oförändrade i RAM-dumparna.

Tid (min)	Förändring (Bytes)	Procent (%)
0	0	0
5	5225027	0,97
15	6332999	1,18

Tabell 3. Tabell över noll- testets förändringar vid angivna tidsperioder.

Det första forensiska verktyget som testades var FTK Imager Lite som vid uppstart gav ett relativt litet "footprint" (4,26 %). Efter att utfört det händelseförlopp som programmet skulle utföra enligt [4.2 Utvinning av volatilt minne med forensiska verktyg], insåg vi att spår som kvarstod i minnet var förvånansvärt litet (0,68 %).

Nigilant32 (0,85 %) var programmet som utgav det absolut minsta avtrycket i minnet efter att ha exekverats. Däremot när verktyget hade utfört sin funktion gav det ett helt annat utslag på det berörda minnet (38,82 %).

Beta versionen av OS-Forensics visade på resultat som indikerade på att programvaran använder relativt stor minnes-resurs vid en uppstart (17,67 %). Det mest häpnadsväckande var att programvaran efter sin användning enligt [4.2 Utvinning av volatilt minne med forensiska verktyg] gav höga resulterande värden (44,77 %).

I de fall då vi beslutade oss för att endast genomföra två dump-filer visade Helix (8.42 %) ett resultat där det inte påverkade systemet i någon märkbar utsträckning. Däremot visade både Win32dd (45,66 %) och Memoryze (51,39 %) en förändring av minnet som inte gick att undvika. Därför tog vi ett beslut att fokusera på Win32dd för en vidare undersökning.

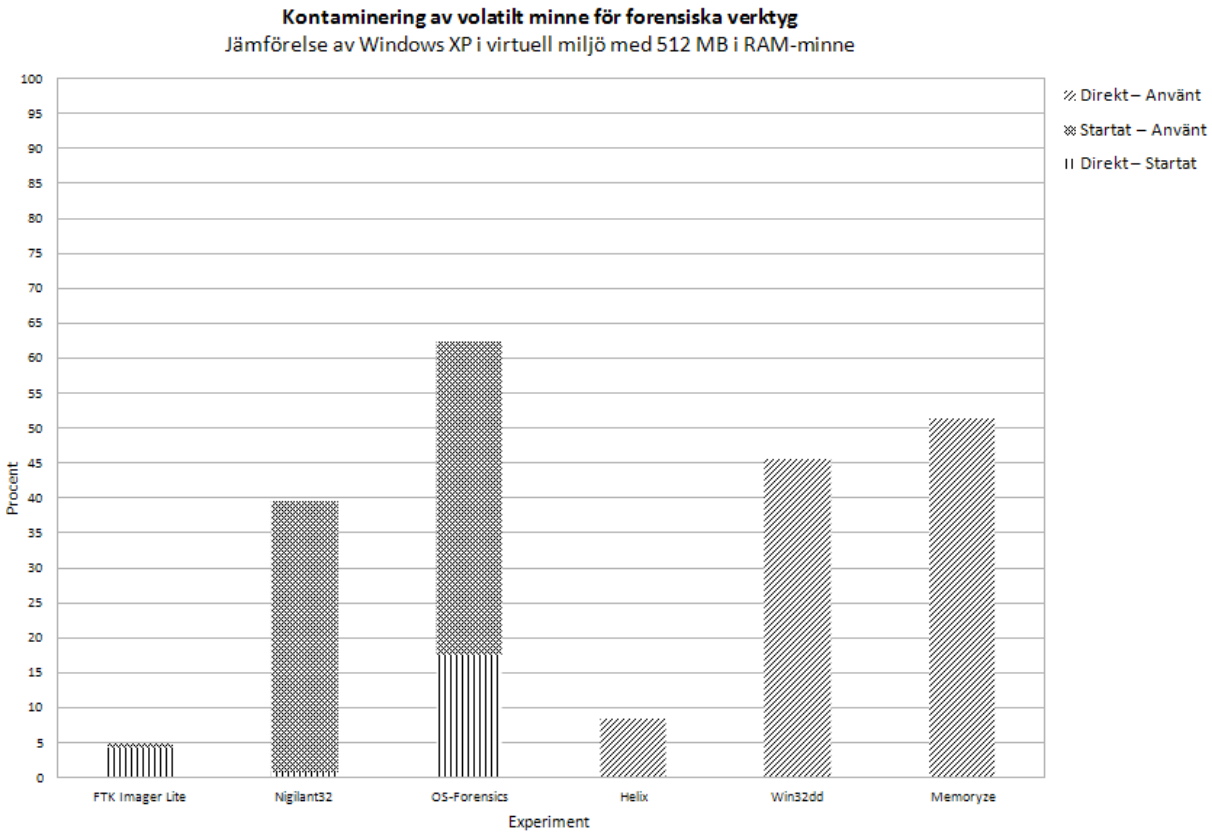


Diagram 2. Diagram över de programvaror som ingår i experimentet.

Programvaror	Dump I - Dump II Förändring (Bytes)	Procent (%)	Dump II - Dump III Förändring (Bytes)	Procent (%)
FTK Imager Lite	22865625	4,26	3641132	0,68
Nigilant32	4553686	0,85	208393715	38,82
OS-Forensics	94884578	17,67	240334722	44,77

Tabell 4. Resultat med tre dump-filer.

Programvaror	Dump I - Dump III Förändring (Bytes)	Procent (%)
Helix	45185988	8,42
Win32dd	245134008	45,66
Memoryze	275888316	51,39

Tabell 5. Resultat med två dump-filer.

Eftersom vi blev överraskade av det resultat som framgick av det första experimentet med Win32dd utfördes en vidare analys av fallet. Som vi tidigare nämnt i rapport är eftersträvan att alltid minimera påverkan av systemet vid en live-respons. Därför genomförs experimentet ytterligare en gång enligt [4.3 Analys] där en jämförelse återkopplas till det föregående resultatet.

Det framgick att när lagringsmediet är specificerat på den lokala hårddisken (45,66 %) blir påverkan av volatilt minne betydligt högre än när man lagrar minnes-dumpen på ett externt lagringsmedia (20,69 %).

För att få en uppfattning över hur RAM-dumpen (DD-dump) som programvaran Win32dd genererat (18,43 %) kopieras den utifrån den virtuella miljön till värdsystemet. Därefter jämförs DD-dumpen mot .vmem-filen (Använt) som skapas då systemet har satts i hiberneringsläge (20,69 %), vilket bidrar till att filerna har en differens på (12154214 Bytes) 2,26 %.

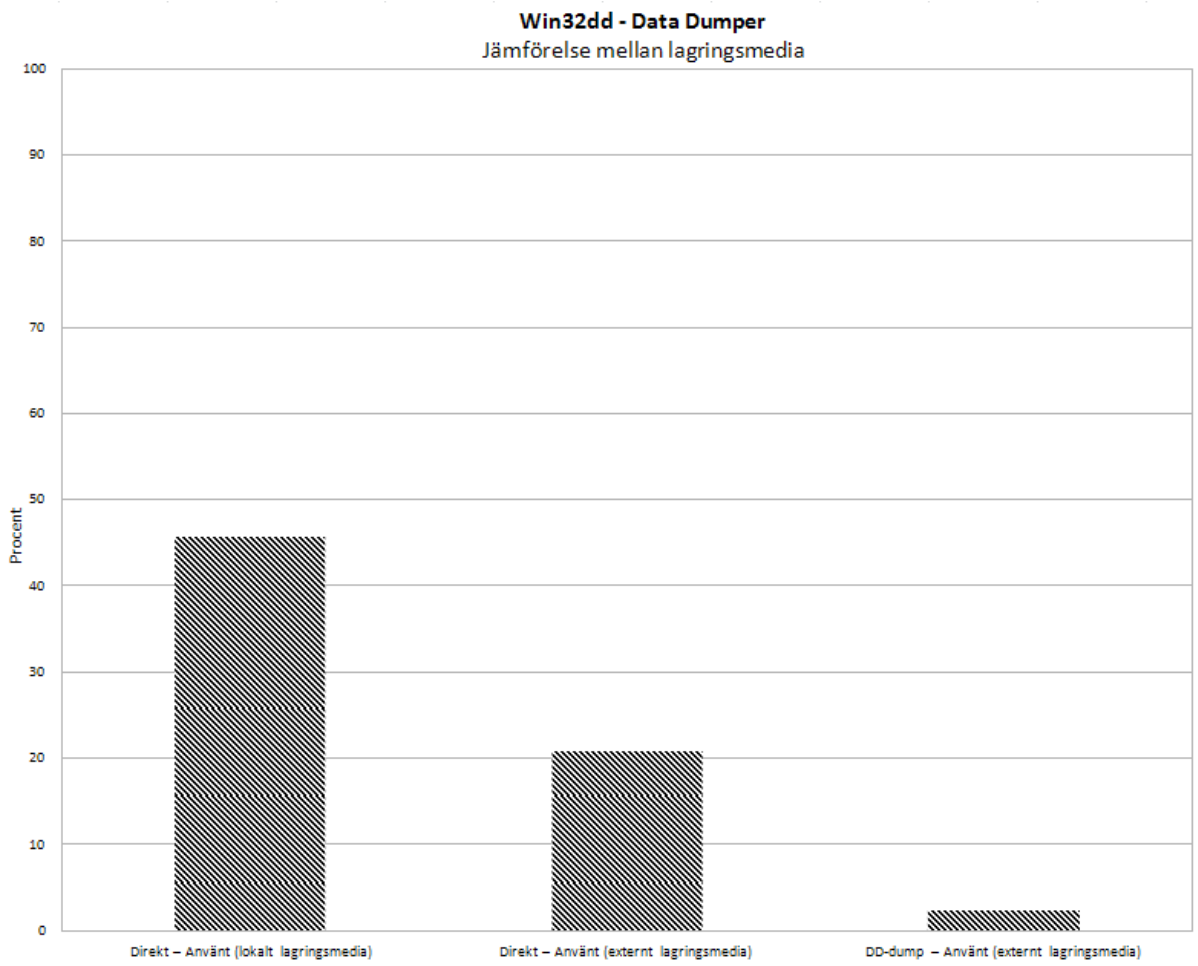


Diagram 3. Jämförelse av datautvinning med lagring på lokalt mot externt media.

Programvara	Dump I - Dump III (Bytes)	Procent (%)
Win32dd Internt	245134008	45,66
Win32dd Externt	111080502	20,69

Tabell 6. Sammanställning av Win32dd *av intern och extern lagring.*

5 Diskussion

I diskussionen behandlas de för- respektive nackdelar som har uppstått under arbetets gång. Förhoppningar om att kunna sätta arbetet i relation till tidigare studier inom detta område är ett önskvärt mål för rapporten.

Med hjälp av att använda en tillämpad strategi för hur datautvinning av volatilt minne skulle utföras, blev analyseringen av materialet lätthanterligt. Genom att implementera denna lösning i en virtuell miljö undvek vi eventuella felmarginaler som kunde påverka systemet. Detta bidrog till att området blev mer fokuserat mot ett virtualiserat system. Då rapporten blev inriktad mot en mjukvarulösning bidrog det till att andra metoder inte belystes i den bemärkelsen. Volatilt minne innehåller informationsrik data om det körande systemet och vi valde att endast arbeta med 512 MB i RAM-minne i det virtuella systemet [Bilaga A]. Detta gjorde vi endast för att enklare kunna urskilja på olika processer i den binära koden som bygger upp minnet. Idag ser vi en marknad där ett standard-utförande av ett system innefattar minst 4 GB i RAM-minne, vilket innebär ännu mer information för IT-forensiker att undersöka. Kunskap, strategi och rätt prioriteringar gör datautvinningen mer överskådlig och man fokuserar inte på fel ändamål vid en incident.

Metoden vi valde att jämföra dump-filerna på, var en tidskrävande process eftersom alla kommandon manuellt matades in i terminalen. I det här fallet kunde man ha automatiserat processen genom att skriva ett skript som utför den manuella processen åt en. Vi tittade på olika typer av automatiseringsskript som kunde tillämpas till vårt experimenterande men ansåg att den manuella metoden lämpades bättre för våra fåtal experiment. Är syftet att konstatera den exakta signifikanta minnesallokeringen kan det vara angeläget att utföra alla experiment upprepade gånger, där skript har en betydelsefull roll.

Ur ett forensiskt perspektiv lyder den allmänna regeln att man aldrig ska lagra berörda filer på det lokala lagringsmediet. Det insåg vi inte till en början när vårt experiment skulle utföras. Därför blev vi tvungna att i efterhand undersöka om förändringen mellan att lagra minnes-dumpen lokalt mot externt hade någon större inverkan på volatilt minne. Alla experiment gjordes inte om på detta tillvägagångssätt utan endast på en programvara Win32dd, då verktyget visade på en oväntad minnesförändring. Vi kom fram till att placeringen av dump-filen har en stor inverkan på testet med Win32dd. När man placerar dump-filen på ett externt media tillskillnad från att lagra det internt blir minnesförändring betydligt mindre, enligt [Diagram 3]. Detta kan ses som en påvisning av betydelsen att inte påverka systemets struktur vid en live-respons. Efter lite forskning om programvaran DD insåg vi att det fanns en distribution DCFLDD som ska vara mer anpassad för en IT-forensiker [23].

Då vi vill dra en parallell mellan de experiment som har utförts, jämförs de mot en annan fallstudie [1]. Testet vi jämför är vårt noll-test mot fallstudiens ”Baseline test”. Det som skiljer sig åt mellan dessa tester är att ett antivirus program (McAfee Antivirus) är installerat på deras system men också att deras ”Baseline test” pågår under en längre tidsram: 0-900minuter.

I denna jämförelse är dock bara tidsramen från 0 till 10 minuter relevant eftersom fallstudiens tidsram är indelat på ett sätt då vår tidsram sträcker sig mellan 0-15 minuter. När vi jämförde hur mycket den första noll-test dumpen förändrats efter 5 minuter, var skillnaden 0,9 %. Det som även uppmärksammades var versionen av Windows XP där vi använde service pack 3 och de tillämpade en tidigare version (service pack 2).

I många fall kan det ses som en komplicerad tolkningsförmåga att förtydliga vad som står i volatilt minne. Det som vi har påvisat är att vissa delar i minnet står i klartext som går att utläsa med hjälp av enkla analysverktyg. Däremot finns det binärkod i minnet som är svårtolkad och sett ur ett anti-forensiskt sammanhang finns det ett flertal möjligheter att gömma eller förvränga koden.

Swap-filen som i många fall kan ha en betydelsefull roll vid live-respons utvecklades inte till ett område av intresse ur vår synvinkel. Det var främst med avseende på att händelseförloppet var under en kontrollerad miljö och vi betingade den kunskap som krävdes för alla aktiva processer. Vid en verklig incident förekommer säkerligen flera sessioner och processer simultant vilket kan göra det betydligt svårare för en IT-forensiker att undersöka volatilt minne.

De problem som har inträffat under examensarbetets gång är att vi har fått utveckla metoden för hur vi ska kunna påvisa kontaminering av forensiska verktyg ur volatilt minne. Till en början hade vi en plan att använda de olika verktygen som vi tidigare har nämnt och kunna peka på vart exakt programvaran lägger sig i minnesstrukturen. När vi sedan påbörjade experimenterandet insåg vi efter en kort tid då vi använt programvarorna att det inte var särskilt relevant att förstå vart det befinner sig i minnet. Detta kom vi fram till efter lite forskning kring RAM-minnets uppbyggnad och struktur då vi förstod att det inte fanns något filsystem för minnet. Det som vi istället insåg var att det är en mer betydelsefull uppgift att förstå hur mycket programvarorna allokerar i minnet. Detta beror på att programvarorna sprider ut sig i RAM-minnet och det är därför i princip omöjligt att fastställa detta. Det vi kunde se var att OS-Forensics hade en inbyggd funktion för detta ändamål och ett fåtal programvaror kunde dumpa minnet för en specifik process. I efterhand skulle det ha varit intressant att få en bättre inblick i hur EPROCESS:er fungerar i Windows och hur det virtuella minnet är konstruerat då det dedikeras till den virtuella maskinen. Detta är väsentliga delar att ta upp i framtida forskningar som berör området.

6 Slutsats

Det minne som försvinner vid en exekvering av ett program varierar beroende på programvarans storlek innan och efter installation. Det kan vi se genom våra resultat gällande RAM-dumpar [Tabell 4] som beträffar: FTK Imager Lite, OS-Forensics och Nigilant32. FTK Imager Lite har en fysisk storlek på 59945736 Bytes där verktyget påverkar minnet med 22865625 Bytes vid exekvering. Med programvaran Nigilant32 var den fysiska storleken på 774144 Bytes och efter exekvering av verktyget blev påverkan av minnet 4553686 Bytes. OS-Forensics fysiska storlek var 56072987 Bytes och efter exekvering av verktyget blev förändringen 94884578 Bytes. I de andra tre fallen med forensiska verktyg gick det inte att fastställa denna slutsats eftersom att programvarorna inte kunde vara i ett startläge. En slutsats som vi kunde dra utifrån detta var att den fysiska storleken [3.3 Forensiska verktyg] har en betydelse för hur stort ”footprintet” i RAM-minnet blir. Man bör vara medveten om “Locard’s Exchange Principals” och att det finns mängder av faktorer som kan påverka systemets minne. Det som kan fastslås är att volatilt minne alltid förändras med tiden.

Det går inte att fastställa vart minnesallokeringen inträffar i volatilt minne för en programvara. Alla programvaror avsätter en viss mängd minne men det kan spridas ut över hela minnesstrukturen. Därför går det inte att konsekvent fastställa minnesallokeringen i RAM-minnet. Däremot erbjuder en del programvaror tjänsten att kunna utföra en RAM-dump för en specifik process. Verktygen som vi kunde se att det gick att utföra detta med var: OS-Forensics och Win32dd. I OS-Forensics kunde man enkelt se hur processerna sprider ut sig över minnesstrukturen genom ett grafiskt gränssnitt. Däremot med Win32dd var det inte så lättöverskådligt då det genomfördes i terminalen. Slutsatsen om det konsekvent går att fastställa minnesallokeringen för en programvara i RAM-dumpen är att det inte är relevant då det saknas ett filsystem för RAM-minnet. Det försvårade processen eftersom strukturen i minnet är svårtolkad. Därför blev inriktning på rapporten vinklad mot verktygens påverkan i minnesstrukturen.

Beroende på hur incidenten är utformad kan det vara lämpligt att forma en strategi innan man ger sig i kast med en incident. Vid lokal utvinning krävs fysisk tillgång till det berörda systemet. Denna tillgång behövs inte vid en nätverksutvinning men det krävs användarrättigheter för att kunna utföra något på den andra noden i nätverket. Det har inte utförts några nätverksutvinningar men enligt våra kunskaper anser vi att Win32dd och Nigilant32 är mest lämpade för nätverksutvinning. I experiment för lokal utvinning lämpar sig alla verktyg förutom OS-Forensics och Memoryze då det fysiskt måste installeras på värdsystemet. För en IT-forensiker är strategin olika beroende på tekniska kunskaper och erfarenheter. Kommando-gränssnitt som Win32dd och Memoryze använder är svårare att hantera över lag än de grafiska användargränssnitten Helix, Nigilant32, OS-Forensics och FTK Imager Lite.

Alla verktyg som har använts i detta projekt går att använda vid en incident om man vet vilka artefakter som efterlämnas av programvaran. Det anses lämpligt att tillämpa programverktyg som är fristående som går att använda från ett externt lagringsmedia.

Dessa verktyg är enligt våra tester Win32dd, Nigilant32 och FTK Imager Lite lämpade för medtagande och fristående. De är relativt små verktyg som lätt får plats på en USB-enhet eller en CD-skiva. Det verktyg som utmärkte sig med förvånansvärt goda resultat var FTK Imager Lite som påverkade systemet minst [Diagram 2] i jämförelse med de andra verktygen.

7 Vidare forskning

Akademisk forskning inom volatilt minne är ett relativt nytt område som har goda förutsättningar för att utvecklas inom IT-forensik. För att vidareutveckla detta projekt måste mer undersökningar kring RAM-minnets uppbyggnad och struktur utföras.

Det som Harlan Carvey tar upp i sin bok [6] "Windows Forensics Analysis" gällande skript för minnesutvinning, är inget som vi har valt att inrikta oss på utan metoden lämnas vidare till andras förfogande. Det som går att rationalisera är att tillämpa automatiserade skript som underlättar vid jämförelsen av dump-filer.

Som tidigare nämnts i diskussionen är DCFLDD ett programverktyg som utvecklats för att vara anpassat för ett forensiskt syfte [23]. Det genomfördes inga tester med avseende på det funna verktyget, men för framtida studier kan detta vara ett lärorikt och intressant område att fördjupa sig i.

VMware vCenter Converter är ett program som kan konvertera fysiska system till virtuella maskiner [24]. Detta kan även vara ett spännande område att undersöka då det kan vara av intresse ur ett forensiskt perspektiv. Det man vill kunna konstatera är förändringar i volatilt minne då ett system förflyttas mellan miljöerna.

Referenser

[1] Walter, A. och Petroni, L.N. (2007) ”Voolatools: Integrating Volatile Memory Forensics into the Digital Investigation Process”.
Tillgänglig på http://www.blackhat.com/presentations/bh-dc-07/Walters/Paper/bh-dc-07-Walters-WP.pdf (Mars 2011)

[2] Meyers’, M. (2010) “Managing and troubleshooting PCs”. McGraw-Hill/Osborne Media. ISBN: 978-0-07-171380-1

[3] Mallery, R.J. (2006) “Secure File Deletion: Fact or Fiction?”. Tillgänglig på www.lib.iup.edu/comscisec/SANSpapers/mallery.htm (Mars 2011)

[4] Halderman, A.J., Schoen, D.S., Heninger, N., Clarkson, W., Paul, W., Calandrino, A.J., Feldman, J.A., Appelbaum, J., och Felten, W.E. (2008) “Lest We Remember: Cold Boot Attacks on Encryption Keys”. Tillgänglig på http://citp.princeton.edu/pub/coldboot.pdf (Mars 2011)

[5] Davis, M.R. (2005) “Evolution of Computers and Computing”. Tillgänglig på http://promo.aaas.org/kn_marketing/pdfs/EvoutionofComputers.pdf (Mars 2011)

[6] Carvey, H. (2009) ”Windows Forensic Analysis”. Syngress Publishing. ISBN: 978-1-59749-422-9

[7] Prosise, C. och Mandia, K. (2003) “Incident Response & Computer Forensics”. McGraw-Hill/Osborne Media. ISBN: 9780072226966

[8] Burdach, M. (2006) “Physical Memory Forensics”. Tillgänglig på http://www.blackhat.com/presentations/bh-usa-06/BH-US-06-Burdach.pdf (Mars 2011)

[9] Garcia, L.G. (2007) ”Forensic physical memory analysis: an overview of tools and techniques”. Tillgänglig på
http://www.tml.tkk.fi/Publications/C/25/papers/Limongarcia_final.pdf (April 2011)

[10] Granlund, T. och MacKenzie, D. (2010) ”Ubuntu manuals”. Tillgänglig på http://manpages.ubuntu.com/manpages/intrepid/man1/cmp.1.html (April 2011)

[11] Rubin, P. och MacKenzie, D. (2010) ”Ubuntu manuals”. Tillgänglig på http://manpages.ubuntu.com/manpages/lucid/man1/wc.1.html (April 2011)

[12] Hemert, H.L. (2001) “Digitala kretsar”. Studentlitteratur AB. ISBN: 978-9-14-401918-5

[13] Golden, B. (2009) Virtualization for Dummies – 2nd HP Special Edition. Wiley Publishing, Inc. ISBN: 978-0-470-47832-5

[14] Svensk IT Funktion AB, (2010) ”Hur fungerar virtualisering tekniskt?”.
Tillgänglig på http://www.itf.se/templates/virtualisering.php?categoryID=126&id=372 (April 2011)

[15] Åsblom, J. (2010) ”Windows 7 passerar Vista men XP leder stort”. Tillgänglig på http://www.idg.se/2.1085/1.335181/windows-7-passerar-vista-men-xp-leder-stort (April 2011)

[16] Arora, S. (2006) ” Forensic Analysis with Helix”. Tillgänglig på http://pcquest.ciol.com/content/enterprise/2006/106050502.asp (April 2011)

[17] Mcree, R. (2009) ”Mandiant Memoryze with Audit Viewer”. Tillgänglig på http://www.issa.org/Library/Journals/2009/February/McRee-toolsmith.pdf (April 2011)

[18] Shannon, M.M. (2006) ”Nigilant32 for First Responders Active Memory Imaging”. Tillgänglig på http://www.agileriskmanagement.com/pdfs/Nigilant32forFirstResponders-ActiveMemoryImaging.pdf (April 2011)

[19] PassMark. (2011) ”White paper Building a bootable OSForensics (WinPE)”. Tillgänglig på http://www.osforensics.com/downloads/osforensics-winpe-v1.0.pdf (April 2011)

[20] AccessData. (2010) ”USER GUIDE Find, Organize & Analyze Computer Evidence”. Tillgänglig på http://accessdata.com/downloads/media/Imager_UserGuide.pdf (April 2011)

[21] Kronqvist, S. (2007) ”Brott och digitala bevis”. Norstedts Juridik AB. ISBN: 978-91-39-01236-8

[22] NetMarketShare. (2011) ”Browser Market Share”. Tillgänglig på http://marketshare.hitslink.com/browser-market-share.aspx?qprid=0 (Maj 2011)

[23] Seifried, K. (2008) ”Forensics with Backtrack and Sleuth Kit Sleuthing”. Tillgänglig på http://www.linux-magazine.com/w3/issue/93/026-028_sleuthkit.pdf (Maj 2011)

[24] Principled Technologies, Inc. (2009) ”Migrating legacy physical servers to vmware vsphere virtual machines on dell poweredge m610 blade servers featuring the intel® xeon® processor 5500 series”. Tillgänglig på http://www.principledtechnologies.com/clients/reports/Dell/vSphere-Migration-Guide.pdf (Maj 2011)

[25] Dobos, L. (2009) ” Windows 7 mot Vista och XP - stort prestandatest”. Tillgänglig på http://windows.idg.se/2.12294/1.251011/windows-7-mot-vista-och-xp---stort-prestandatest (Maj 2011)

Bilagor

Bilaga A: Systemspecifikationer

Värdsystemets specifikationer:
Processor: Intel Core 2 Quad Q9550 2.83 GHz
Minne: 8 GB RAM
Operativ: Windows 7 Enterprise 64-bit Sevice pack 1

Virtuella systemets specifikationer:
Processor: AMD Phenom 9850 Quad-Core 2.51 GHz
Minne: 512 MB RAM
Operativ: Windows XP Professional x86 Service pack 3

Bilaga B: Informationsrik data vid live-respons

- Systemtid och datum
- Lista över inloggade användare
- Information om körande processer
- Öppnade filer
- Nätverksinformation
- Nätverksanslutningar
- Vilken port som är ansluten till de olika processerna
- Processens minne
- Nätverksstatus
- Innehållet i ”clipboard”
- Drivrutiners information
- Kommando-historik för systemet
- Utdelade mappar
- Externa enheter eller inloggade resurser

Checklistan är baserad från källorna [6] [7].

Bilaga C: Systemkrav för Windows XP

Operativsystem	Minimum	Maximum
Windows XP Professional SP3 x86	256 MB	4 GB