Laboration 1

Kopiera foldern DT2002 – lab 1 a_v2 och DT2002 – lab 1 b_v2 från WMV_template foldern till WMV foldern på D-disken. Starta maskinen i folder Lab 1 a. Svara ”Create” på första dialogen. Du får då upp en meny där du kan välja mellan F1 och F2. Välj F1 för continue. Lösenordet är gruka.

Antagande
Du ska göra datautvinning från ett befintligt körande Windowssystem. Du har kopplat in en extern hårddisk till systemet dit du kan spara all information (hårddisken ”USB-disk” finns i den virtuella maskinen). Till din hjälp har du FTK Imager som även den ligger på USB-disken.

Analys
Starta FTK Imager. Vi ska börja med att ansluta en fysisk enhet till FTK Imager. Detta gör det möjligt att navigera i filstrukturen på den anslutna enheten och i vårat fall, se att vi valt rätt enhet. Detta gör vi genom ”Add Evidence Item”, type är physical device. Anslut de båda enheterna och kontrollera vilken som har Windows installerat. Det är denna enhet vi ska göra en avbild av senare.

Inhämtning, RAM
Genomför en datautvinning från RAM-minnet (capture memory). All information sparas till USB-disken.

Inhämtning, hårddisk
Gör därefter en datautvinning från C-disken (create disk image). Avbilden är en fysisk diskavbildning, avbilden ska sparas som en EnCase-fil (E01), casenummer är HH-dagens datum-01, bevisnummer är HDD-001, examiner är ditt namn. Avbildens filnamn sätts till lab1a, fragmentstorleken 4000 MB och compression till 9. Denna process kommer att ta ungefär 15 minuter. Använd gärna den inbyggda hjälpfilen för att bekanta dig med den. Under tiden som utvinningen pågår kan du gå vidare med Linux-uppgiften.

Sammanfattning
Starta om FTK Imager för att bli av med alla anslutna enheter. Vi ska nu kontrollera att utvinningen gått bra. Du kan experimentera lite med FTK Imager och undersöka hur du kan navigera i dina avbilder. Du lägger till en avbild med kommandot add evidence item, type är image file.
Innehåller RAM-dumpen spår efter din utvinningsstrategi? Vaför?
Innehåller disk-avbilden spår efter din utvinningsstrategi? Varför?

Utvinning i Linuxmiljö

Nu ska vi upprepa proceduren från ett Linux-system. Starta maskinen i Lab 1 b. Detta är det Linux-baserade verktyget Backtrack 4. Tryck return i den första menyn som dyker upp på skärmen. När allt laddat klart efter någon minut och du ser en prompt, ge kommandot startx. Detta startar den grafiska miljön. När allt startat så öppnar vi en kommandoprompt. Kontrollera vart du hittar cd-spelaren med hjälp av kommandot df. Nu ska vi göra en avbild av dvdn som ligger färdig i spelaren. Men först måste vi montera hårddisken som vi ska spara data till och för att kunna göra det så måste vi ha en monteringspunkt. Gör följande:

1. Skapa en monteringspunkt (mkdir /mnt/diska)
2. Montera disken (mount /dev/sda /mnt/diska)
3. Skapa en avbild av cd-spelaren (dd if=/dev/hda of=/mnt/diska/cd-001.dd), cd-spelarens enhet kan variera.
4. Skapa en checksumma av avbilden (md5sum /mnt/diska/cd-001.dd > /mnt/diska/cd-001.md5)

Checksummor

Det är viktigt att kunna säkerställa att ingen ändrat innehållet på en hårddisk som undersöks som bevis. Ett sätt att kontrollera konsistensen av en dataenhet över tiden är att använda checksummor. Ett exempel på en checksumma kan vara att summera alla tal på data-enheten och komma ihåg om resultatet var udda eller jämt.

Exempel:
3, 7, 2, 9

Summan är 21 och således udda. Vi antecknar detta. Skulle vi vid ett senare tillfälle ta fram dataenheten och checksumman visar sig vara jämn vet vi att data har ändrats. Denna enkla checksumma används faktiskt ännu idag i vissa data-kommunikationsprotokoll. Däremot är algoritmen aldeles för enkel för vår applikation. Den algoritm som verktyget encase använder är konstruerad på ett sådant sätt att man inte (enkelt) kan vända på beräkningen och få fram data som skulle matcha en given checksumma. Detta gör det mycket svårt för en individ att ändra på ett data-sett så att man dels får med den avsedda datan och dessutom återställer checksumman till den ursprungliga. Det är inte omöjligt, men ytterst svårt och tidskrävande, vilket gör att normal kontroll av materialet tillsammans med en kompletterande checksumme-metod, gör det ytterst osannolikt att någon skulle klara av att åstadkomma en förändring utan att påverka checksumman. Troligtvis skulle det vara långt enklare att se till att ändra på datasettet innan du får möjlighet att ta en första checksumma.

Oavsiktlig kontaminering
Det är viktigt att man tänker sig noga för innan man hanterar ett data-set så att man inte av misstag kontaminerar detsamma. Vi ska belysa detta med ett experiment.

Experiment:
Du ska genomföra ett antal olika operationer på en hårddisk (/dev/sdb) och efter varje operation ska du skapa en checksumma som du dokumenterar. Genom denna process kommer vi att lära oss hur lite det krävs för att en hårddisk ska påverkas.

För att skapa en checksumma ger du kommandot md5sum /dev/sdb.

I sammanfattning ska du skapa en folder /mnt/diskb och montera /dev/sdb till denna folder. Navigera till foldern med cd och lista innehållet med ls. Skapa en tom fil med hjälp av touch filnamn. Radera filen igen. Vänta 2 minuter. Avmontera disken. Mellan varje steg skapar du en checksumma.

Händelse	Checksumma
Innan vi gjort något	
Monterat /dev/sdb	
Gjort cd/ ls	
Skapat fil	
Raderat fil	
Väntat	
Avmonterat	

Analys:

Analysera data i tabellen. Vad hände?
Hur kan vi förhindra att systemet skriver data till den fysiska enheten?
Är checksumman före det du lägger till en fil och efter det du tagit bort filen samma?
Vad betyder detta?

Du är nu klar. Titta över vad du gjort idag och fråga på sådant som är oklart. Fundera över hur du skulle kunna öva vidare på egen hand. Glöm inte att radera alla de virtuella maskiner du använt.

Demonstration

DBAN – disk wipe