Computer Forensics, Del 2

Dette er så del 2 af min serie af artikler der handler om Computer Forensic. Den første artikel handlede om emnet i bredde termer og generelle vendinger. Her i artikel 2 tager jeg så fat på nogle af de værktøjer og procedure man bruger, samt hvordan det g

Skrevet den **06. Feb 2009** af **bufferzone** I kategorien **Sikkerhed / Generelt** | ★★★☆☆

Computer Forensics, del 2

Computer Forensics, Del 1: http://www.eksperten.dk/artikler/616
Computer Forensics, Del 2: http://www.eksperten.dk/artikler/685

Forensic system
Du er nødt til at have et system der kan foretage de nødvendige teste og undersøgelser, og selvom det er bedst at have et dedikeret system, der kun bruges til dette formål, kan det sagtens sættes op fra gang til gang.

Du kommer ikke udenom at anvende en Linux box. Jeg vil ikke her ind i en ligegyldig diskussion om hvilket operativsystem, der er bedst. I min verden drejer det sig om, at kende de forskellige systemers stærke og svage sider, og så udnytte de forskellige systemer der hvor det giver mening.
En at Linux systemernes meget stærke sider, er den kontrol du har over alle aspekter og den dybde du kan gå ned i systemet og ændre forskellige parametre. Dette har du brug for dels for at kunne sikre at intet bliver ændret eller ødelagt utilsigtet.

Din Linux box bør have så mange ledige pladser til harddiske som muligt (mindst 2), der bør være en brænder så du kan flytte resultater og fil eksempler og hvis du har brug for at kunne analysere f.eks. SCSI-, SATA-diske eller USB devices, skal der også være understøttelse og kontroller til dette.
Hvilken distribution du vælger er mindre væsentlig, du bør gå efter stabilitet og at du kender og er tryk ved den. Selv anvender jeg forskellige distributioner f.eks. Debian, Fedora og PHLAK, men også Knoppix bruger jeg til at boote døde systemer og foretage andre ting der med fordel kan gøres fra en CD bootable. distribution.

Beviserne sikres.

Som jeg allerede var inde på i artikel 1, er sikring og fuldstændig bevarelse af de oprindelige beviser meget vigtige, især i retslig sammenhæng. Og ikke blot skal vi bevare dem uændret, vi skal også kunne bevise, at det er sådan.

Vi placerer den disk der skal undersøges i vores Linux box og booter. Da Linux oftest som default ikke mounter ekstra harddiske (test om din distribution gør og hvis den gør, så ændre det), men tillader, at vi gør det efterfølgende, har vi sikkerhed for, at der ikke bliver ændret noget ved disken. Når vi efterfølgende mounter disken har vi også kontrol over hvordan dette gøres og kan dermed sikre, at der til stadighed ikke ændres ved denne. Mere herom senere.

MD5 Checksum
Vi starter med at tage en MD5 checksum af den disk vi skal undersøge. Dette gøres for senere at kunne bevise, at disken ikke har ændret sig mens den har været i vores varetægt, og for at kunne bevise at de images vi laver og efterfølgende arbejder på, er fuldstændige identiske med originalen, og svaret er "ja du

kan godt tage en MD5 checksum af en disk der ikke er mountet.

> MD5 beskrives I detaljer I RFC 1321. (http://www.faqs.org/rfcs/rfc1321.html). MD5 algoritmen tager et input i form at noget tekst af vilkårlig længde og producere et output i form af et 128-Bit "fingeraftryk" eller "message digest" af inputtet, kaldet en hash værdi. Det ligger i algoritmen, at det er umuligt at producere det samme 128-bit fingeraftryk af to forskellige tekster (selv to marginalt forskellige tekster), samt at du ikke kan regne dig tilbage til teksten ud fra hashværdien (en såkaldt one-way-hash).

På mit forensic system er Linux box'ens harddisk benævnt hda og den hardisk vi skal analysere bliver så benævnt hdb. Herunder kan du se hvordan MD5 Checksummen tages, hvor første linie er kommandoen og den anden linie er den resulterende checksum.

```
# md5sum /dev/hdb
812f450f73b6bef45af3c92d52bd7e23  /dev/hdb
```

Vi har nu en checksum vi dels kan sammenligne med senere checksumme taget fra samme system, og vi kan sammenligne den med de checksumme vi tager fra de imagebackups vi laver at systemet.
MD5 checksumme bruges både til at validere vores beviser (filer og data fra drev og andre medier) og til at validere vores værktøjer. Ved at tage en MD5 checksum af et værktøj, kan vi efterfølgende genskabe resultaterne og bevise at det er gjort med et identisk værktøj.

Kontrol af disken
En kontrol af hvad den disk vi skal undersøge egentlig indeholder, kan vi foretage med fdisk kommandoen. Vi bruger subkommandoen p til at udskrive diskens partitionstabel (under forudsætning af at denne ikke er ødelagt).

```
# fdisk /dev/hdb

Command (m for help): p

Disk /dev/hdb: 4321 MB, 4321787904 bytes
255 heads, 63 sectors/track, 525 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot    Start       End    Blocks   Id  System
/dev/hdb1   *         1       128   1028128+  83  Linux
/dev/hdb2           129       525   3188902+   5  Extended
/dev/hdb5           449       525    618471   82  Linux swap
/dev/hdb6           129       448   2570337   83  Linux

Partition table entries are not in disk order
```

Vi kan her se hvordan disken er opdelt i logiske partitioner og det er tydeligt at vi står over for en harddisk der med stor sandsynlighed indeholder et linux system. Disse oplysninger er måske ikke umiddelbart nødvendige for at finde beviserne, men det er vigtigt, især retsligt, at vi finder alle tilgængelige informationer, dels fordi vi ikke på nuværende tidspunkt kan vide hvad vi skal bruge og ikke bruge og dels for at vise at vi er grundige og metodiske.

Klargøring

Først skal vi have gjort en disk klar til vores bit stream backup. Denne skal være mindst lige så stor som den disk vi skal kopiere gerne større.
Da vi foretager en bit for bit streaming og skal have alt med fra disken inklusiv ubrugt plads m.v. er vi nødt til at starte med at wipe den disk vi skal anvende.
Når man sletter en fil, fjernes data i virkeligheden ikke fra disken. Det eneste der fjernes er referencen til hvor på disken filen befinder sig. Indtil det område overskrives igen, vil data stadig kunne læses (det er dette vi benytter os ag når vi med forskelligt software genskaber slettede filer). Selv ikke en formatering sletter data ude på disken, en formatering ligger kun en ny fil allokerings tabel ind på disken, d.v.s. overskriver den gamle så alle referencerne er fjernet.
En wipe gør dels det at den fjerner referencerne, og så overskrives diskene med forskelligt mere eller mindre tilfældigt data, afhængig af hvilket program der anvendes.

Linux har indbygget mulighed for at wipe en disk. Til dette formål anvender vi samme kommando som vi sener skal bruge til at lave selve bit stream imaget med. Vi udnytter linux begrebet "dev zero", der er et virtuelt drev ligesom "dev null". Linux vil behandler både "dev zero" og "dev null" som en hver anden disk eller medie, du kan lave backup (det går stærkt, men du vil have svært ved at finde filerne bagefter) du kan kopiere filer til "dev null" og "dev zero", og du kan også læse fra begge drev.
Læses fra "dev null" returneres et "ende of file" mens en læsning fra "dev zero" vil give null karakteren tilbage (/0, ikke karakteren 0, hvilket er ligegyldigt i vores tilfælde). Dette betyder at vi kan overskrive enhver disk med null karakterer ved at lave et bit stream image fra dev zero til den disk vi ønsker at wipe.

```
# dd if=/dev/zero/ of=/dev/hdc
39102336+0 records in
39102336+0 records out
```

Denne proces er ganske langsommelig, så læs resten af artiklen mens null karkaterene strømmer ned.

Hvis du hellere vil anvende windows, så se på:

BCWipe
http://www.jetico.com/
BCWipe er et effektivt lille program med mange features der ret effektivt fjerner og overskriver data fra medier. Den kan indstilles til at overskrive stort set alle områder af en disk, og hvis man vil være sikker indstilles den til at gøre dette flere gange (Normalt køres mindst 7 passes hvis man vil være sikker på at data ikke bare kan genskabes).

Bit stream images

> Bit Stream Back-up: A bit stream kopiering er en sector-for-sector/bit-for-bit kopi af en harddisk. En bit stream kopi repræsentere ikke blodt filer og bibliotheksstruktur, den repræsentere alt latent data på mediet. Computer forensics specialiser anvender en række forskellige programmer og utilities til dette f.eks. Encase, SnapBack, Ghost og den gratis Linux dd.

Som beskrevet i Artikel 1., er det ikke blot data, men også de områder af diskene der ikke umiddelbart indeholder data, der er interessante og som derfor skal medtages i backuppen. Disse områder kan nemlig sagtens indeholde både skjult data og bevidst eller ubevidst ødelagt data og andre spor.
Den type backup image vi har taler om, kaldes for bit stream image/backup. Her laves et image af

originalen, bit for bit, uden hensyn til indhold eller data. Alle diskens nuller og ettaller kopiere over direkte over som de er, hvormed alt kommer med på kopien. Master Boot record, data, formateret tom plads, formateret delvist overskrevet plads, ødelagte partitioner og data, krypteret data og ikke formateret plads (Master Boot recordt, unallocated space og disk slack.)

Når du skal vælge værktøj til bit stream backup, er det første du skal beslutte dig for, om dine beviser skal kunne bruges i en retssag eller ej. Hvis der er mulig for dette, bør du vælge et værktøj, der allerede har været brugt i retslig sammenhæng, eller som anvendes af andre retslige/myndigheds instanser, helst i det land hvor retssagen skal køre, men alternativt i international sammenhæng. I Danmark har der ikke været ret mange sager, og myndighederne holder kortene ret tæt til kroppen. Du har derfor ikke ret mange andre muligheder end at gå efter internationalt anerkendte produkter. Her er nogle eksempler:

SafeBack
http://www.forensics-intl.com/safeback.html
http://www.lyme.com/
SafeBAck er et meget anerkendt og i international sammenhæng meget brugt stykke software til forensic bit stream imaging.

Snapback
http://www.snapback.com/
Snapback er i virkeligheden er backup og restore løsning til Windows servere der også kan lave bit stream backup, og dermed bruges i denne sammenhæng. Denne applikation er en løsning for dem der allerede bruger Snapback til backup, skal du betale, vil jeg anbefale at du vælger en løsning der anvendes i forensic sammenhæng.

EnCase Forensic V5
http://www.guidancesoftware.com/orderv5today.shtm
Lige som SafeBack en anerkendt og meget brugt applikation i forensic sammenhæng.

Image MASSter Solo II Forensic Hard Drive Duplicator
http://www.upgradesolutions.com/products/imagemassterforensic.html
Her er der tale om et stykke hardware. D.v.s. en færdig løsning hvor i diske kan placeres og analyseres. En løsning for den der ofte har brug for at lave den slags analyser.

Endelig kan du anvende Norton Gost, men den anvendes ikke at Law
Enforcement.

Bit stream imaging med Linux
For at vise et eksempel på hvordan dette kan gøres med vores linux box anvender vi her dd kommandoen er en del af linux. Og bemærk at vi ikke har mountet drevet endnu.
Igen bruger vi linux kommandoen dd og denne gang med subkommandoen time for atvise hvor meget tid der faktisk går med disse ting. Vi bit stream kopiere fra det drev vi skal analysere hdb til den nyligt wipede disk hdc.

```
# time dd if=/dev/hdb/ of=/dev/hdc
20044080+0 records in
20044080+0 records out


real    36m7.687s
user    0m27.480s
sys    8m4.930s
```

Herefter forestår kun at tage en md5 checksum for at verificere og dokumentere at bit stream imaget indeholder præcist det samme som orginalen.

Det var så den anden artikel i serien. Artikel nr. 3 vil indeholde selve analysen af disken samt (hvis der er plads) beskrivelse af hvordan man kan lave analyse af en ukendt binær fil. Igen vil jeg koncentrere mig om at vise hvordan man selv kan lave tingende med en linux box samt give anvisninger på hvordan det gøres professionelt.

Hvis du har kommentarer og/eller spørgsmål, er du velkommen til at kontakte mig på kim@bufferzone.dk lige som jeg selvfølgelig er at finde på Eksperten. Jeg modtager kommentarer og rettelser (f.eks. af stavefejl) med tak. Jeg beder om at du ikke stiller spørgsmål i kommentarerne til artiklen, der kan jeg jo ikke besvare dem

Kommentar af cyberfinn d. 24. Apr 2005 | 1

Utrolig god artikel..

Kommentar af frewald d. 24. Apr 2005 | 2

Glæder mig til resten af serien

Kommentar af dustie d. 20. Dec 2008 | 3

Som altid super godt skrevet :)

Kommentar af aros d. 25. Apr 2005 | 4

Kan ikke vente på at resten kommer

Kommentar af j4anus d. 27. Apr 2005 | 5

<3 bufferzone!