

Recuperación de Datos: Data Carving y Archivos Fragmentados

**Jornada
Nacional de
Seguridad
Informática**

COMPUTACIÓN FORENSE: RASTREANDO LA INSEGURIDAD INFORMÁTICA

JUNIO 20, 21 Y 22 DE 2007

Biblioteca Luis Ángel Arango

Calle 11 No 4-14

Bogotá D.C., Colombia.

Daniel A. Torres Falkonert
datorres@poligran.edu.co

Johany A. Carreño Gamboa
jcarreno@poligran.edu.co



Motivación

Recuperación de Datos en Sistemas de Archivos

1. Recuperación de Datos

2. Perdida de Metadatos de
Sistemas de Archivos

3. Técnicas Antiforenses

Fundamentos de File Carving

Recuperación de Datos:
File Carving y Archivos Fragmentados



Definición.

Identificación y Recuperación de Archivos basados en las características de formato de los archivos sin el conocimiento de las estructuras del sistema de archivos

Fundamentos de File Carving

Recuperación de Datos:

File Carving y Archivos Fragmentados



File Carving es una poderosa herramienta porque permite:

1. Identificar y recuperar archivos de interés, que hayan sido borrados, o se encuentren en un sistema de archivo dañado, memoria, o información que se encuentra en el archivo de paginación y tráfico de red
2. Asistir en la recuperación de archivos y datos que no son tenidos en cuenta por el sistema operativo y el sistema de archivos
3. Asistir en un proceso forense de recuperación de archivos y datos, así como en un proceso normal de recuperación de datos

Limitaciones

Recuperación de Datos:
File Carving y Archivos Fragmentados



- La mayoría de herramientas solo pueden recuperar archivos que no se encuentran fragmentados en el disco
- Se producen un gran número de falsos positivos, las herramientas no realizan una validación exhaustiva del archivo recuperado
- Se puede recuperar el contenido de los archivos pero no sus metadatos ni la estructura de directorios
- Es relativamente simple engañar a las herramientas (Técnicas Antiforenses)
- Es un proceso muy lento y requiere mucho espacio

Metodología

Recuperación de Datos:
Data Carving y Archivos Fragmentados



- Identificación
- Validación
- Extracción

Identificación

Recuperación de Datos:
File Carving y Archivos Fragmentados



En general, los archivos de diferentes formatos poseen características invariantes

Ejemplo.

Archivos JPG:

Encabezado(header)

`\xff\xd8\xff\xe0\x00\x10`

Fin de archivo (footer)

`\xff\xd9`

Identificación

Recuperación de Datos:

File Carving y Archivos Fragmentados

Para recuperar un archivo JPEG:

- Encontrar la información de encabezado y fin de archivo
- En general, las técnicas actuales solo analizan el encabezado y el final del archivo

Hexdump de atlantis.jpg

Encabezado

```
00000000 ff d8 ff e0 00 10 4a 46 49 46 00 01 01 00 00 01 |.....JFIF.....|
00000010 00 01 00 00 ff db 00 43 00 1f 15 17 1b 17 13 1f |.....C.....|
00000020 1b 19 1b 23 21 1f 24 2e 4d 32 2e 2a 2a 2e 5e 43 |...#!$.M2.**.^C|
00000030 47 38 4d 6f 62 75 73 6d 62 6c 6a 7b 8a b1 96 7b |G8Mobusmb|}{...{|
00000040 83 a7 84 6a 6c 9a d1 9c a7 b6 bc c6 c8 c6 77 94 |...j|.....w.|
00000050 d9 e8 d7 c0 e6 b1 c2 c6 be ff db 00 43 01 21 23 |.....C.!#|
00000060 23 2e 28 2e 5a 32 32 5a be 7f 6c 7f be be be be |#.(.Z22Z..l.....|
00000070 be be be be be be be be be be be be be be be be |.....|
```

Fin de archivo

```
000001b0 1a 25 a9 88 c7 3a 0a d5 df 73 ce 81 eb 9f ed f9 |.%...:....s.....|
000001c0 6a a6 2e 42 7b 4d 49 10 08 1f 2a 58 7d 98 c7 de |j..B{Ml...*X}...|
000001d0 a6 6e 89 0b 07 7c 7e 55 2b b2 ee 13 d9 20 50 a8 |.n...|~U+.... P.|
000001e0 8e 19 42 3d 28 74 b5 1e a3 ba 48 6d b8 20 0d 3c |..B=(t....Hm. <|
000001f0 a8 14 c5 c1 96 59 1e 5f 90 a8 51 2e 6e 50 a7 d0 |.....Y...Q.nP..|
00000200 b6 b5 28 06 c2 4f 69 9c c6 69 66 a3 b5 4e af 0c |..(..Oi..if..N..|
00000210 89 ac 57 46 f5 2e 0c de a1 2d 2f 42 41 c9 d5 24 |..WF.....-/BA..$|
00000220 d0 d9 b9 5b 28 d2 90 98 99 c8 a2 5e 12 a0 85 13 |...[(.....^....|
00000230 90 00 a5 c0 91 54 53 4f ff d9 |.....TSO..|
```


Archivos Fragmentados

Recuperación de Datos:
File Carving y Archivos Fragmentados



- Si se encuentra una referencia del sistema de archivos (metadatos) a la cadena de clusters del archivo puede no ser necesario realizar Data Carving
- Los sistemas de archivos modernos evitan al máximo la fragmentación, sin embargo esta es inevitable
- Los sistemas de archivos actuales optimizan el uso en disco evitando al máximo la fragmentación, incluso llegan a reubicar archivos que han cambiado su tamaño
- Utilizando las herramientas adecuadas es posible forzar la fragmentación

Verificación

Recuperación de Datos:
File Carving y Archivos Fragmentados



Programa en ANSI C que utiliza la librería libsndfile V.1.0.11 para verificar el formato de un archivo PCM WAV.

Se utiliza la función `sf_command` con el flag `SFC_GET_LOG_INFO`. Esto lee e imprime los metadatos del archivo `wav`

Verificación

Recuperación de Datos:
File Carving y Archivos Fragmentados

```
4 #include <ctype.h>
5 #include <sndfile.h>
6
7 #define BUFFER_LEN (1 << 16)
8 static double data [BUFFER_LEN] ;
9
10 void
11 main (int argc, char *argv [])
12 { static char strbuffer [BUFFER_LEN] ;
13   char *infilename ;
14   SNDFILE *sndfile ;
15   SF_INFO sfinfo ;
16
17   infilename = argv [1] ;
18
19   sndfile = sf_open (infilename, SFM_READ, &sfinfo) ;
20
21   if (! sndfile) //si el archivo no es válido
22     printf ("Error:%s:", infilename) ;
23   else
24     { //Si el archivo es válido
25       sf_command (sndfile, SFC_GET_LOG_INFO, strbuffer, BUFFER_LEN) ;
26       puts (strbuffer); //imprimir informacion (metadatos) del archivo
27     }
28   sf_close (infile) ;
29 }
```

Verificación

Recuperación de Datos:
File Carving y Archivos Fragmentados



Script en perl que utiliza el la herramienta que verifica la integridad de un archivo con formato PCM WAV y segun la respuesta lo clasifica en una de las 3 Categorías:

. Erróneo .Incompleto .Completo

Verificación

Recuperación de Datos: File Carving y Archivos Fragmentados

```
1 use Tie::File;
2 use strict;
3
4 my $inputfile = ARGV[1];
5 my $logfile = ARGV[2];
6 my @infile;
7
8 tie @infile, 'Tie::File', $inputfile;
9 open (LOGFILE, ">$logfile.$dateumx");
10
11 foreach $arch_wav (@infile) {
12
13     my $cmdout = `./sndfile-info $arch_wav`; #ejecución del programa que lee archivos wav
14
15     if ($cmdout =~ /^Error/){ # El archivo no es valido
16         print LOGFILE "+$_: Archivo no valido\n";
17     }else{ #2-Si el archivo es valido hay 2 opciones
18         $verif = `echo "$cmdout" | /bin/grep RIFF`; # La línea donde aparece esta cadena
19                                                     # es clave para la verificación
20         if ($verif =~ /\(should/) # Cadena que aparece cuando el tamaño del archivo en
21                                     # los metadatos no corresponde con el tamaño real
22             print LOGFILE "+$_: Archivo incompleto\n";
23         else{ # 2b que el archivo este completo
24             print LOGFILE "+$_: Archivo correcto\n";
25         }
26     }
27 }
28 }
29
30
31 close(LOGFILE) if ($args{"1"});
32 untie @infile;
33 exit(0);
```

Construcción de Casos

Recuperación de Datos:
File Carving y Archivos Fragmentados



Creación del espacio de pruebas

```
dcfldd if=/dev/urandom of=fs.random.dd bs=1M count=45
```

Creación de los fragmentos

```
split -b 4k -d atlantis.jpg
```

Construcción de Casos

Recuperación de Datos:
File Carving y Archivos Fragmentados

Archivo sin fragmentación

```
dd if=x00 of=fs.random.dd bs=4k seek=100 count=1 conv=notrunc  
dd if=x01 of=fs.random.dd bs=4k seek=101 count=1 conv=notrunc  
dd if=x02 of=fs.random.dd bs=4k seek=102 count=1 conv=notrunc  
dd if=x03 of=fs.random.dd bs=4k seek=103 count=1 conv=notrunc
```

Archivo bi-fragmentado

```
dd if=x00 of=fs.random.dd bs=4k seek=200 count=1 conv=notrunc  
dd if=x01 of=fs.random.dd bs=4k seek=201 count=1 conv=notrunc  
dd if=x02 of=fs.random.dd bs=4k seek=202 count=1 conv=notrunc  
dd if=x03 of=fs.random.dd bs=4k seek=206 count=1 conv=notrunc
```

Construcción de Casos



Recuperación de Datos:
File Carving y Archivos Fragmentados

Archivo incompleto

```
dd if=x00 of=fs.random.dd bs=4k seek=100 count=1 conv=notrunc  
dd if=x01 of=fs.random.dd bs=4k seek=101 count=1 conv=notrunc
```

Archivo multi-fragmentado

```
dd if=x00 of=fs.random.dd bs=4k seek=200 count=1 conv=notrunc  
dd if=x01 of=fs.random.dd bs=4k seek=202 count=1 conv=notrunc  
dd if=x02 of=fs.random.dd bs=4k seek=204 count=1 conv=notrunc  
dd if=x03 of=fs.random.dd bs=4k seek=206 count=1 conv=notrunc
```


Construcción de Casos

Recuperación de Datos:
File Carving y Archivos Fragmentados

Archivo secuencial con fragmentos en orden inverso

```
dd if=x03 of=fs.random.dd bs=4k seek=100 count=1 conv=notrunc  
dd if=x02 of=fs.random.dd bs=4k seek=101 count=1 conv=notrunc  
dd if=x01 of=fs.random.dd bs=4k seek=102 count=1 conv=notrunc  
dd if=x00 of=fs.random.dd bs=4k seek=103 count=1 conv=notrunc
```

Archivo con fragmentos muy dispersos

```
dd if=x00 of=fs.random.dd bs=4k seek=300 count=1 conv=notrunc  
dd if=x01 of=fs.random.dd bs=4k seek=350 count=1 conv=notrunc  
dd if=x02 of=fs.random.dd bs=4k seek=400 count=1 conv=notrunc  
dd if=x03 of=fs.random.dd bs=4k seek=450 count=1 conv=notrunc
```

Casos Patológicos

Fragmentación

Recuperación de Datos:
File Carving y Archivos Fragmentados

Volumen (C:)

Tamaño del volumen	= 35,00 GB
Tamaño de clúster	= 4 KB
Espacio utilizado	= 18,92 GB
Espacio libre	= 16,08 GB
Porcentaje de espacio disponible	= 45 %

Fragmentación del archivo

Cantidad de archivos	= 107.736
Tamaño promedio de archivo	= 263 KB
Cantidad de archivos fragmentados	= 11.884
Cantidad de fragmentos en exceso	= 59.574
Promedio de fragmentos por archivo	= 1,55

La fragmentación ocurre naturalmente

Estado del Arte

Recuperación de Datos:
Data Carving y Archivos Fragmentados

- Smart Data Carving
- Análisis de entropía (para encontrar límites entre archivos)
- In Place Carving
- Fast Object Validation
- Network Traffic Carving
- ...

Herramientas

Recuperación de Datos:
Data Carving y Archivos Fragmentados



- Foremost
- Scalpel
- Encase
- WinHex
- FTK
- Entre otras...

Conclusiones

Recuperación de Datos:
Data Carving y Archivos Fragmentados



Los ejemplos, técnicas y demás mostrados en ésta conferencia, no son la última palabra ni los únicos métodos prácticos para detectar tipos de archivos en el mundo real. Sin embargo, la información suministrada pretende motivar a los presentes a unir esfuerzos y proveer información para la generación de futuros estudios

Se debe dar crédito a la idea que todos los patrones asociados a los datos son una manera futura de identificar los formatos de archivos esperados

Se hace necesario modelar y construir mecanismos automatizados para que las redes puedan reconocer ciertos tipos de patrones según los metadatos asociados a los archivos

La técnicas anti-forenses están evolucionando a un ritmo muy acelerado

Trabajo Futuro

Recuperación de Datos:
Data Carving y Archivos Fragmentados

- Automatización en la identificación de formatos de archivos utilizando machine learning (Técnicas de Minería de Datos, Redes Neuronales, entre otros)
- Semantic Carving
- Identificación y reconstrucción de fragmentos basados en el análisis de los mismos

Referencias



1. G.G. Richard III, V. Roussev, L. Marziale, "In-place File Carving," Research Advances in Digital Forensics III, Springer, 2007.
<http://www.cs.uno.edu/~golden/Stuff/ifip2007-final.pdf> Consultado en mayo de 2007
2. G. G. Richard III, V. Roussev, "Scalpel: A Frugal, High Performance File Carver," Proceedings of the 2005 Digital Forensics Research Workshop (DFRWS 2005), New Orleans, LA.; http://www.dfrws.org/2005/proceedings/richard_scalpel.pdf Consultado en mayo de 2007
3. Farmer, Dan; Venema, Wietse "Forensic Discovery". (2004) Addison Wesley Professional. ISBN: 020163497X
4. B. Carrier, "File System Forensic Analysis" (2005), Addison-Wesley
5. S. Garfinkel, "Carving Contiguous and Fragmented Files with Fast Object Validation"; <http://www.simson.net/clips/academic/2007.DFRWS.pdf>; consultado en mayo de 2007
6. <http://computer.forensikblog.de/en/topics/carving/> ; Consultado en junio de 2007