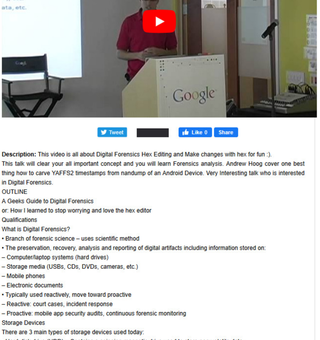
A GEEK'S GUIDE TO DIGITAL FORENSICS



Description: This video is all about Digital Forensics Hex Editing and Make changes with hex for fun :).
This talk will clear your all important concept and you will learn Forensics analysis. Andrew Hoog cover one best thing how to carve YAFFS2 timestamps from nandump of an Android Device. Very interesting talk who is interested in Digital Forensics.
OUTLINE
A Geeks Guide to Digital Forensics
or: How I learned to stop worrying and love the hex editor
Qualifications
What is Digital Forensics?
• Branch of forensic science – uses scientific method
• The preservation, recovery, analysis and reporting of digital artifacts including information stored on:
– Computer/laptop systems (hard drives)
– Storage media (USBs, CDs, DVDs, cameras, etc.)
– Mobile phones
– Electronic documents
• Typically used reactively, move toward proactive
– Reactive: court cases, incident response
– Proactive: mobile app security audits, continuous forensic monitoring
Storage Devices
There are 3 main types of storage devices used today:
• Hard-disk drive (HDD) – Contains a spinning magnetic drive used to store non-volatile data.
• Solid-state drive (SSD) – Contains internal microchips for the purpose of storing non-volatile data.
• NAND Flash memory
• Typically found in smart phones. USB thumb drivers and other portable devices
• Not removable like typical HDD or SSD
• Very unique characteristics from standard HDD (limited writes/erase)
• In constant state of change (FTL)
Acquisition strategies
Forensics Analysts can acquire/receive data 3 different ways
• Backup Files
– Backup files are provided from the "custodian". This could include backup software from corporations, PST file, iTunes backup, etc.
• Logical Acquisition
– A copy of the file system is created (i.e. tar.gz of / or recursive copy that preserves date/time)
• Physical Acquisition
– Creates an exact digital replica of the storage medium
– Can recover deleted data
– This process requires specialized analysis tools and techniques
– Drive management firmware may still affect acquisition (FTL, bad blocks, etc.)
Image Verification
• Hash value – A calculated hex signature based on a set of data.
– A hash value can be used to verify forensic image integrity. One slight change in source will cause "avalanche" effect in hash value
– In order to prove that two data sets are identical, their hash values must match.
– In some instances, hash values are not stable (NAND Flash) so a hash of the data as it's extracted is taken but won't necessarily match if source is imaged again
• Common hash techniques
– md5 (128-bit value)
– sha256 (256-bit value)
• md5 of "Andrew Hoog" = 9bdbad9aecd74fce6e6bb48ee18100b8
How to acquire a forensic image
• If possible, connect drive to a physical write blocker
– This prevents any writes to the drive
– There are software techniques but not as effective
– Generally, impossible with NAND Flash devices
• Forensically acquire device with software
– Open source: dd, dcfdd and dc3dd (we use the later)
– Free: FTK Imager and many others
– Commercial: FTK, EnCase, etc.
• Perform verification of source and image with hash signature and record in Chain of Custody
Example imaging with dc3dd
• Department of Defense's Cyber Crime Center dc3dd
– Patched version of GNU dd
– includes a number of features useful for forensics
– Free and open source
• Command:
– dc3dd if=/dev/disk1 of=/dev001.dc3dd vss=on hash=sha256 hlog=drive001.hashlog log=drive001.log rec=off
– rec=off determines how to handle I/O errors (recover=off)
– Full details: http://dc3dd.sourceforge.net/
• ./configure; make; sudo make install
Handling failing drives
• May run into drive issues, have to decide how to handle
– Stop on error
– Continue, fill with NULLs (0×00)
– Skip (would result in smaller dd image, not recommended)
• Example of errors:
• Potential workaround
– GNU ddrescue – very powerful alternative, install from source
– Will rescue blocks, read drive backwards, restart where last left off
– http://www.gnu.org/software/ddrescue/ddrescue.html
"Typical" forensic analysis steps
• Create timeline of events
• File system modified, accessed, changed and created
• Metadata from files (images, documents, flash cookies, etc)
• Mount dd image read-only
• Generate list of all files (allocated and deleted)
• Analyze key files
• Windows: Registry, LNK files, user profile, web history, etc.
• Linux: Bash history, recently-used.xbel, gvfs-metadata, etc.
• Recover deleted files
• File carving (handles unallocated)
• Search files, dd image, etc.
• Many specialized techniques
Analyzing forensic image (F/OSS)
• The Sleuth Kit by Brian Carrier
– Brain author of excellent book File System Forensics Analysis (FSFA)
– Actively maintained, just released 3.2.2 (06/13/2011)
– Supports NTFS, FAT, UFS 1, UFS 2, EXT2FS, EXT3FS, and ISO 9660
– http://sleuthkit.org/
• Programs to start with:
– mmls – Media Management ls, generally partition info:
TSK – File system info
• fsstat – File system information:
TSK – listing (all) files
• fls – Forensic list
– Power utility which can list allocated/deleted files
– Provides offset so recovery is possible
– Build MACB for timeline analysis
– fls -z CST6CDT -s 0 -m '/' -f ext3 -r -o 63 -i raw file.dd > body
mactime – make body file human friendly
• mactime -b body -z CST6CDT -d > timeline.csv
– Takes body file and turns into CSV or other format
Mount dd image read-only
• Determine file system offset in dd image:
• Mount FAT16 (and many others fs') partition read only:
• Perform additional analysis on files
Log2timeline
• Kristinn Gudjonsson developed this software
– Written in Perl (trying to convince him to move to Python)
– Extracts timeline artifacts from many file types including
• Evt/evtx, registry, SMTP, prefetch, browser history, etc. (46 and climbing)
– 10+ export formats
– http://log2timeline.net/
• timescanner -d ~/mnt/sdcard -z CST6CDT -w body.tx
• If you output in body format, can combine with TSK's fls output and generate full timeline of file system and file metadata
Regripper
• Harlan Carvey developed this software
– Written in Perl
– Windows is primary platform, there is a Linux port
– Parses Windows registry files
• Support hives: NTUSER.dat, system, software, sam, security, etc.
– http://regripper.wordpress.com/regripper/
Scalpel
• Download scalpel src at:
• wget http://www.digitalforensicssolutions.com/Scalpel/scalpel-2.0.tar.gz
• Compile
– tar xzvf scalpel-2.0.tar.gz
– cd scalpel-2.0/
– ./configure; make
– sudo cp scalpel /usr/local/bin
• Run scalpel
$ scalpel -c scalpel.conf ~/Desktop/image.dd
$ scalpel -c android-scalpel.conf ~/Desktop/android-image.nanddump
• Examine data in "scalpel-output" directory
Android Flash Memory
• Android devices use a raw flash device, and therefore need a Flash Transition Layer (FTL)
– FTL provides basic block interface to developers
– Handles wear leveling, bad block management, metadata, etc.
• FTL is provided by Memory Technology Device (MTD)
– MTD is open source
– Newer Android devices are moving to eMMC where FTL controller is embedded with the memory (similar to thumb drives and SSD)
• MTD divides memory into blocks, each of which is 128K with a 64 byte Out-of-Band (OOB) area
– OOB houses YAFFS2 tags, meta data, bad blocks and ECC
YAFFS2 – Block/Chunk/OOB diagram
Android Forensics
• Logical recovery can be achieved through Content Providers
– We've developed free tool for law enforcement: AFLogical
– Commercial: viaExtract – http://viaforensics.com/products/viaext/tract/
• Secure Backup CPros
– To extract more data, we first need to escalate privileges on the device.
• This presentation is not intended to cover these techniques (a.k.a. get a Google Dev phone or go read XDA)
• Logical Acquisition
– With escalated privileges, we can simply connect to the device using the Android Debug Bridge (adb) and execute an adb pull command on the files that we wish to acquire. (i.e. /data/data)
Android Forensics – Physical acquisition
• Physical Acquisition
– Android dd image
• The dd utility on Android devices is only capable of reading the non-OOB data from the YAFFS2 MTD partition
– Full NAND image
• Includes OOB
• We use an in-house developed nanddump utility capable of reading and extracting all data from the YAFFS2 partition (and dealing with bad blocks)
• Allows an examiner to take full advantage of the YAFFS2 features, primarily artifacts from being a log-structure file system
YAFFS2 Timeline
nanddump -c /dev/mtd0ro | grep -v "00 00 00 00 00 00 00 00 00 00 00 00 00 00" | grep -v "FFFFFFFFFFFFFFFF" | less
0×00006800: 10 00 00 00 10 10 00 00 FF FF 96 96 c6 56 13 e2 |.........ÿÿ.––Æ V.â|
0×00006810: 47 87 47 47 00 00 00 00 00 00 00 00 00 00 00 00 |G‡GG..........|
0×00006900: 00 00 00 00 00 00 00 00 00 00 00 00 FF FF 16 00 |............ÿÿ..|
0×00006910: d6 00 00 00 57 00 00 00 63 99 d5 d4 d7 99 d5 d4 |Ö...W...c.ÕÔ×.ÕÔ|
0×00006920: 42 a9 d5 d4 51 00 00 00 FF FF FF FF FF FF 5B 4D |B©ÕÔQ...ÿÿÿÿÿÿ[M|
0×00006fc0: FF FF FF FF FF FF FF FF 60 00 00 00 |ÿÿÿÿÿÿÿÿ`...|
0×00006fe0: FF FF FF FF FF FF 00 00 00 00 FF FF |ÿÿÿÿÿÿ....ÿÿ|
0×00006ff0: FF FF FF FF FF FF 00 00 00 00 00 00 |ÿÿÿÿÿÿ......|
OOB Data: FF 10 01 00 00 20 10 00 01 10 10 00 08 51 00 |.....ÿÿÿÿÿÿ.....Q.|
OOB Data: 00 00 51 af e2 e2 10 00 00 00 FF FF FF FF |..Q¯ââ.....ÿÿÿÿ|
OOB Data: FF FF FF FF FF FF 00 3c f8 3c FF FF |ÿÿÿÿÿÿ.<ø<ÿÿ|
OOB Data: FF FF FF FF FF FF FF FF FF FF FF FF |ÿÿÿÿÿÿÿÿÿÿÿÿ|
• Number as written to NAND flash: 63 99 d5 d4 (0×6399d5d4)
• Converted from little endian to big endian: 4d 5d 99 36 (0×4d5d9936 which is the hex read from right to left)
• Converting 0×4d5d9936 (hex) to base 10 is 1297979702
• Unix time stamp 1297979702 in human date time format is Thu Feb 17 15:55:02 CST 2011 (date -d @1297979702)
YAFFS2 Timeline
• Using this information, we can isolate a number of important artifacts
– atime (accessed time) for a directory along with mtime and ctime
– Object ID to the directory within the OOB
– Object ID for files and cross-reference to make sure it is consistent with debug data.
• Additional analysis would allow us to create the MAC times for each file and directory on the NAND.
• It is also possible to gather additional meta data information from ObjectHeaders found on the NAND.
Proactive forensics
• Forensics has typically been used reactively
• By moving forensic techniques into proactive security services, excellent results are achieved
– appWatchdog: basic security testing for mobile apps
• http://viaforensics.com/appwatchdog/
– Mobile app security: see online presentation
• http://viaforensics.com/computer-forensics/mobile-app-security-presentation-andrew-hoog.html
– liveForensics: continuous forensic monitoring of key assets
• http://viaforensics.com/services/security/liveforensics/
Contact viaForensics
Andrew Hoog
Chief Investigative Officer
ahoog@viaforensics.com
http://viaforensics.com

Source :- https://viaforensics.com/computer-forensics/google-tech-talk-geeks-guide-to-digital-forensics-june-2011.html (googletalk)

Tags: google , hacking , hack ,

Comments:

Login to post a comment

VIDEO POSTED BY

By **tinitee**
3576 Views, Posted Mon 16 Jul 2012 ago

View All His Videos

ST COURSE VIDEOS

Defeating Getimages.inc: Checks In File Uploads


Challenge 6: Digest Authentication Reloaded


Challenge 5: Digest Authentication Attack


Bas ic Authentication And Form Bruteforcing


Http Bas ic Authentication Attack (Easy)


Challenge 2: Http Form Attack's Reloaded


Http Vars Tampering Demo


Web Application Pentesting Course Introduction


Introducing Pentester Academy


Hack Of The Day: 13: Remote Shellcode Launcher Tel ling Shellcode Over A Network


Hack Of The Day: 12: Pivots And Port Forwards In Scenario Based Pentesting

Hack Of The Day: Customizing Shellcode For Fun And Profit

Hack Of The Day: How Do I Run Unt rusted Shellcode?

Securitytube Linux As sembly Expert Exam Format

[Slae] Writing An Custom Inverton Encoder

[Slae] Eneave Shellcode Stack Method

[Slae] Shellcoding Bas ics

[Slae] Ut ng Libc And Nas m

[Slae] Hello World In As sembly Language

[Slae] What Is Assembly Language?