Stealth File Systems for Proactive Forensics Support in Custom Android ROMs

Guide: Dr. Prabhakar Mateti[1] Sudip Hazra[2]

[1]Wright State University

[2]Amrita Centre For Cyber Security Systems and Networks
Amrita University

12 December 2016

Outline

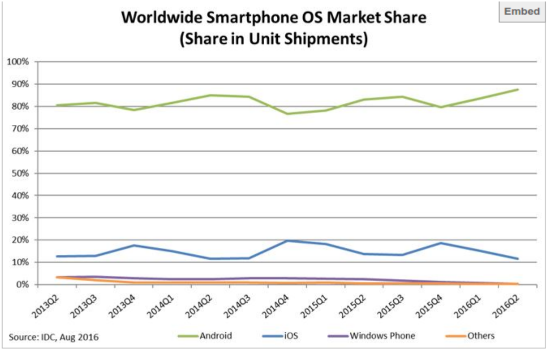
1. Introduction
    - Why Android
    - Why Mobile Forensics is Important
    - Problem Scenario
2. Background
    - Types of Mobile Forensics Investigation
    - Forensic Support Framework
    - Shortfalls
    - Possible Solutions
    - File System in User Space
    - Cloud File System
    - Rootkits
3. Problem Definition
4. Solution Outline
    - Modules
    - Cloud Storage API's
5. Implementation
    - SleuthKit Forensics Toolkit
    - Recovered Email Id's
    - Recovered Images
    - Hiding a Fuse File System using Rootkit-1
    - Hiding a Fuse File System using Rootkit-2
6. Summary

1. Introduction
   - Why Android
   - Why Mobile Forensics is Important
   - Problem Scenario
2. Background
   - Types of Mobile Forensics Investigation
   - Forensic Support Framework
   - Shortfalls
   - Possible Solutions
   - File System in User Space
   - Cloud File System
   - Rootkits
3. Problem Definition
4. Solution Outline
   - Modules
   - Cloud Storage API's
5. Implementation
   - SleuthKit Forensics Toolkit
   - Recovered Email Id's
   - Recovered Images
   - Hiding a Fuse File System using Rootkit-1
   - Hiding a Fuse File System using Rootkit-2
6. Summary

Why Android

Mobile OS Market

Figure : Smartphone OS Market Share, 2016 Q2



Source:http://www.idc.com/prodserv/smartphone-os-market-share.jsp

1. Introduction
    - Why Android
    - **Why Mobile Forensics is Important**
    - Problem Scenario
2. Background
    - Types of Mobile Forensics Investigation
    - Forensic Support Framework
    - Shortfalls
    - Possible Solutions
    - File System in User Space
    - Cloud File System
    - Rootkits
3. Problem Definition
4. Solution Outline
    - Modules
    - Cloud Storage API's
5. Implementation
    - SleuthKit Forensics Toolkit
    - Recovered Email Id's
    - Recovered Images
    - Hiding a Fuse File System using Rootkit-1
    - Hiding a Fuse File System using Rootkit-2
6. Summary

Why Mobile Forensics is Important

The use of cell phones and computers are like a journal or diary of users lives. Just from cell phones, a mobile phone forensic analysis by International Investigators can reveal a great deal of data, including:

- Dialed, incoming and missed calls (history logs)
- Text messages
- Instant message activity
- Email
- Internet activity including search histories
- Phone location information (using GPS) and cell phone tower triangulation

1. Introduction
    - Why Android
    - Why Mobile Forensics is Important
    - Problem Scenario
2. Background
    - Types of Mobile Forensics Investigation
    - Forensic Support Framework
    - Shortfalls
    - Possible Solutions
    - File System in User Space
    - Cloud File System
    - Rootkits
3. Problem Definition
4. Solution Outline
    - Modules
    - Cloud Storage API's
5. Implementation
    - SleuthKit Forensics Toolkit
    - Recovered Email Id's
    - Recovered Images
    - Hiding a Fuse File System using Rootkit-1
    - Hiding a Fuse File System using Rootkit-2
6. Summary

Problem Scenario

To Catch a Thief, Think Like a Thief!

- If criminals and crime organizers use smart phones, what would they do?
- Will they browse? If so which browser? What site?
- How to predict the next move?
- How to Collect Evidence if they erase the Phone Memory aka. Factory Reset.

1. Introduction
    - Why Android
    - Why Mobile Forensics is Important
    - Problem Scenario
2. Background
    - Types of Mobile Forensics Investigation
    - Forensic Support Framework
    - Shortfalls
    - Possible Solutions
    - File System in User Space
    - Cloud File System
    - Rootkits
3. Problem Definition
4. Solution Outline
    - Modules
    - Cloud Storage API's
5. Implementation
    - SleuthKit Forensics Toolkit
    - Recovered Email Id's
    - Recovered Images
    - Hiding a Fuse File System using Rootkit-1
    - Hiding a Fuse File System using Rootkit-2
6. Summary

Types of Mobile Forensics Investigation

- **Reactive Forensics :**
  Investigation done after Crime has happened.Susceptible to Applications like Uninstall-It, can potentially wipe out all user data and Device Encryption can be a barrier for investigation.
- **Proactive Forensics:**
  A suspect or potential terrorist is monitored proactively in realtime to prevent a crime. Real-time monitoring and analysis is possible. Data Encryption will not be a hindrance in evidence collection.Ability to retrieve deleted data and logs and prevent potential crimes .

1. Introduction
   - Why Android
   - Why Mobile Forensics is Important
   - Problem Scenario
2. Background
   - Types of Mobile Forensics Investigation
   - Forensic Support Framework
   - Shortfalls
   - Possible Solutions
   - File System in User Space
   - Cloud File System
   - Rootkits
3. Problem Definition
4. Solution Outline
   - Modules
   - Cloud Storage API's
5. Implementation
   - SleuthKit Forensics Toolkit
   - Recovered Email Id's
   - Recovered Images
   - Hiding a Fuse File System using Rootkit-1
   - Hiding a Fuse File System using Rootkit-2
6. Summary

Forensic Support Framework

Figure : Features of forensic Rom developed by Aiyappan Et.al [1] and Karthik Et.al [2]

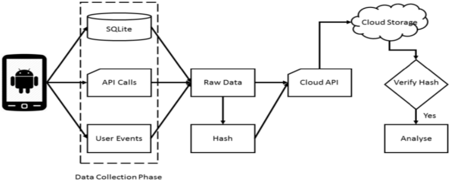


Figure: Forensic Service Architecture

1 .Captures All User Activities.
2 .Key-logging and Call Tapping Facility.
3 .Opportunistically Uploads In Cloud.
4. Hiding the Process using hidepid =2.
5. Data Stored in /forensic partition only accessible to Root.

1 Introduction
- Why Android
- Why Mobile Forensics is Important
- Problem Scenario

2 Background
- Types of Mobile Forensics Investigation
- Forensic Support Framework
- Shortfalls
- Possible Solutions
- File System in User Space
- Cloud File System
- Rootkits

3 Problem Definition

4 Solution Outline
- Modules
- Cloud Storage API's

5 Implementation
- SleuthKit Forensics Toolkit
- Recovered Email Id's
- Recovered Images
- Hiding a Fuse File System using Rootkit-1
- Hiding a Fuse File System using Rootkit-2

6 Summary

Shortfalls

- What if The Suspect Roots the Phone ?
- Can Find the /forensic Partition.

1. Introduction
    - Why Android
    - Why Mobile Forensics is Important
    - Problem Scenario
2. Background
    - Types of Mobile Forensics Investigation
    - Forensic Support Framework
    - Shortfalls
    - **Possible Solutions**
    - File System in User Space
    - Cloud File System
    - Rootkits
3. Problem Definition
4. Solution Outline
    - Modules
    - Cloud Storage API's
5. Implementation
    - SleuthKit Forensics Toolkit
    - Recovered Email Id's
    - Recovered Images
    - Hiding a Fuse File System using Rootkit-1
    - Hiding a Fuse File System using Rootkit-2
6. Summary

Posible Solutions

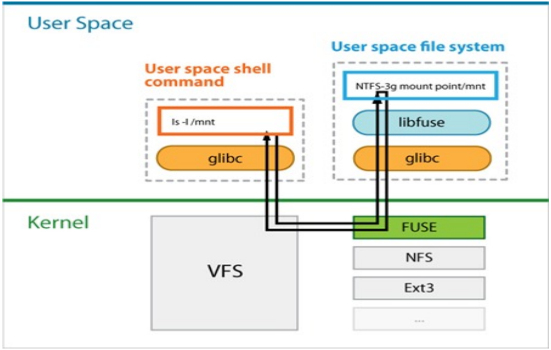
- Encrypting The /forensic partition can Still arise Suspicion.
- Creating A Fuse File System and enable Stealth Features and Copy all Forensically Relevant Data in that File System.

1. Introduction
   - Why Android
   - Why Mobile Forensics is Important
   - Problem Scenario
2. Background
   - Types of Mobile Forensics Investigation
   - Forensic Support Framework
   - Shortfalls
   - Possible Solutions
   - File System in User Space
   - Cloud File System
   - Rootkits
3. Problem Definition
4. Solution Outline
   - Modules
   - Cloud Storage API's
5. Implementation
   - SleuthKit Forensics Toolkit
   - Recovered Email Id's
   - Recovered Images
   - Hiding a Fuse File System using Rootkit-1
   - Hiding a Fuse File System using Rootkit-2
6. Summary

File System in User Space

- The Filesystem in Userspace (FUSE) is a special part of the Linux kernel that allows regular users to make and use their own file-systems without needing to change the kernel or have Root privileges.

Figure : A Fuse Filesystem.



Source:en.wikipedia.org/wiki/Filesystem in Userspace

1. Introduction
   - Why Android
   - Why Mobile Forensics is Important
   - Problem Scenario
2. Background
   - Types of Mobile Forensics Investigation
   - Forensic Support Framework
   - Shortfalls
   - Possible Solutions
   - File System in User Space
   - Cloud File System
   - Rootkits
3. Problem Definition
4. Solution Outline
   - Modules
   - Cloud Storage API's
5. Implementation
   - SleuthKit Forensics Toolkit
   - Recovered Email Id's
   - Recovered Images
   - Hiding a Fuse File System using Rootkit-1
   - Hiding a Fuse File System using Rootkit-2
6. Summary

Cloud File System

- Using FUSE we can mount Cloud Drive in Our System and Use it Like a Local File System.
- Gcsfuse: A user-space file system for interacting with Google Cloud Storage.
- Wingfs: A debian Package to mount various cloud storage drives as user-space file systems.
- Azurefs: A python package to mount Azure blob storage as Local File system.

1. Introduction
    - Why Android
    - Why Mobile Forensics is Important
    - Problem Scenario
2. Background
    - Types of Mobile Forensics Investigation
    - Forensic Support Framework
    - Shortfalls
    - Possible Solutions
    - File System in User Space
    - Cloud File System
    - Rootkits
3. Problem Definition
4. Solution Outline
    - Modules
    - Cloud Storage API's
5. Implementation
    - SleuthKit Forensics Toolkit
    - Recovered Email Id's
    - Recovered Images
    - Hiding a Fuse File System using Rootkit-1
    - Hiding a Fuse File System using Rootkit-2
6. Summary

Rootkits

A rootkit is a clandestine computer program designed to provide continued privileged access to a computer while actively hiding its presence. rootkit allows someone to maintain command and control over a computer without the computer user/owner knowing about it. Types of Rootkits:

- User Level Rootkits
- Kernel Level Rootkits Like:
  - Hooking System Calls
  - Direct Kernel Object Manipulation (DKOM)
  - Interrupt Descriptor Table (IDT) Hooking

Problem Definition

What is the Goal

- To mount the Cloud storage as a Local File system in Android.
- To Provide Support for Multiple Cloud storage Providers.
- The forensic file system will copy itself in parts to the cloud file system.
- The file system will be opportunistically get uploaded to the cloud storage.
- To hide both the Cloudfs and the forensic partition using Rootkits.

Architecture Diagram

Here we propose a Stealth File system with cloud support Below the Android Software stack.

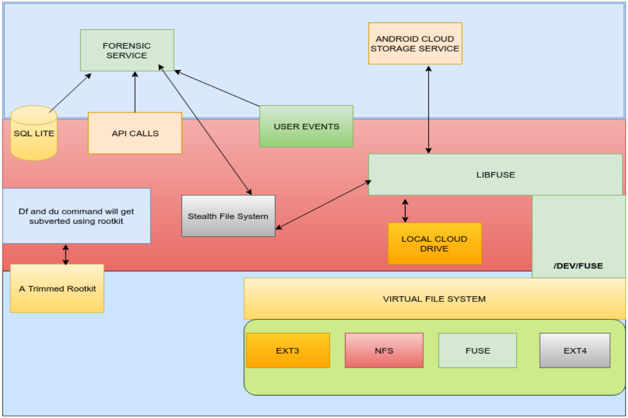


Figure : Android Cloud Storage Service

1. Introduction
   - Why Android
   - Why Mobile Forensics is Important
   - Problem Scenario
2. Background
   - Types of Mobile Forensics Investigation
   - Forensic Support Framework
   - Shortfalls
   - Possible Solutions
   - File System in User Space
   - Cloud File System
   - Rootkits
3. Problem Definition
4. Solution Outline
   - Modules
   - Cloud Storage API's
5. Implementation
   - SleuthKit Forensics Toolkit
   - Recovered Email Id's
   - Recovered Images
   - Hiding a Fuse File System using Rootkit-1
   - Hiding a Fuse File System using Rootkit-2
6. Summary

Modules

Stealth File Systems

- The stealth file system will be a seperate file system which will be used by the forensic service to copy all the forensically relevant data from the relevant device partitions like /data to the stealth file system.
- The stealth file system will be based on ext4 file systems, rootkits will be used to hide the file volumes from commands like df and du.
- The rootkit will also hide the forensic service running in the background by hooking on to the sys_call_table and filtering out the output.

Modules

Cloud File System

- The cloud File system will made using fuse , and we will be able to localy mount a cloud drive as the cloud file system. It will support various cloud providers using apis
- **As per the official android documentation the external storage (SD cards) are accessed by the Android system using FUSE which implies that FUSE is supported by the kernel directly so we dont need to add fuse support in kernel.**

1. Introduction
   - Why Android
   - Why Mobile Forensics is Important
   - Problem Scenario
2. Background
   - Types of Mobile Forensics Investigation
   - Forensic Support Framework
   - Shortfalls
   - Possible Solutions
   - File System in User Space
   - Cloud File System
   - Rootkits
3. Problem Definition
4. Solution Outline
   - Modules
   - Cloud Storage API's
5. Implementation
   - SleuthKit Forensics Toolkit
   - Recovered Email Id's
   - Recovered Images
   - Hiding a Fuse File System using Rootkit-1
   - Hiding a Fuse File System using Rootkit-2
6. Summary

Cloud Storage API's

Cloud Storage API's Examples:

Dropbox Cloud Storage Api's:

1 .Create a Dropbox folder

post('https://api.dropbox.com/1/fileops/create_folder', args)

2.Rename a Dropbox file/directory object.

post('https://api.dropbox.com/1/fileops/move', args)

3.Delete a Dropbox file/directory object.

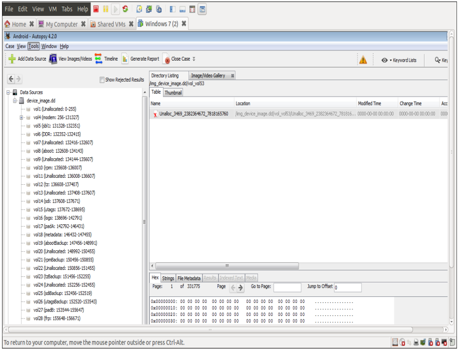
post('https://api.dropbox.com/1/fileops/delete', args)

4.Get Dropbox metadata of path.

get('https://api.dropbox.com/1/metadata/auto' + path, args)

1. Introduction
   - Why Android
   - Why Mobile Forensics is Important
   - Problem Scenario
2. Background
   - Types of Mobile Forensics Investigation
   - Forensic Support Framework
   - Shortfalls
   - Possible Solutions
   - File System in User Space
   - Cloud File System
   - Rootkits
3. Problem Definition
4. Solution Outline
   - Modules
   - Cloud Storage API's
5. **Implementation**
   - **SleuthKit Forensics Toolkit**
   - Recovered Email Id's
   - Recovered Images
   - Hiding a Fuse File System using Rootkit-1
   - Hiding a Fuse File System using Rootkit-2
6. Summary

Results of Android Device Forensics after factory reset



```
Data Sources
  device_image.dd
    vol1 (Unallocated: 0-255)
    vol4 (modem: 256-131327)
    vol5 (sbl1: 131328-132351)
    vol6 (DDR: 132352-132415)
    vol7 (Unallocated: 132416-132607)
    vol8 (aboot: 132608-134143)
    vol9 (Unallocated: 134144-135607)
    vol10 (rpm: 135608-136007)
    vol11 (Unallocated: 136008-136607)
    vol12 (tz: 136608-137407)
    vol13 (Unallocated: 137408-137607)
    vol14 (sdi: 137608-137671)
    vol15 (utags: 137672-138695)
    vol16 (logs: 138696-142791)
    vol17 (padA: 142792-146431)
    vol18 (metadata: 146432-147455)
    vol19 (abootBackup: 147456-148991)
    vol20 (Unallocated: 148992-150455)
    vol21 (rpmBackup: 150456-150855)
    vol22 (Unallocated: 150856-151455)
    vol23 (tzBackup: 151456-152255)
    vol24 (Unallocated: 152256-152455)
    vol25 (sdiBackup: 152456-152519)
    vol26 (utagsBackup: 152520-153543)
    vol27 (padB: 153544-155647)
    vol28 (frp: 155648-156671)
```

/img_device_image.dd/vol_vol53

Name	Location	Modified Time	Change Time	Acc
Unalloc_3469_2382364672_7818165760	/img_device_image.dd/vol_vol53/Unalloc_3469_2382364672_781816...	0000-00-00 00:00:00	0000-00-00 00:00:00	000

```
Page:   1   of  331775        Page

0x00000000: 00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
0x00000010: 00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
0x00000020: 00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
0x00000030: 00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
```

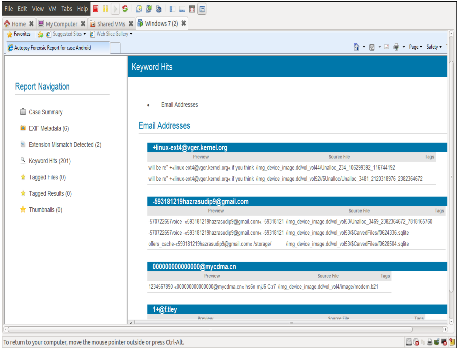
To return to your computer, move the mouse pointer outside or press Ctrl-Alt.

Figure : Sleuthkit Forensic Toolkit

1. Introduction
   - Why Android
   - Why Mobile Forensics is Important
   - Problem Scenario
2. Background
   - Types of Mobile Forensics Investigation
   - Forensic Support Framework
   - Shortfalls
   - Possible Solutions
   - File System in User Space
   - Cloud File System
   - Rootkits
3. Problem Definition
4. Solution Outline
   - Modules
   - Cloud Storage API's
5. Implementation
   - SleuthKit Forensics Toolkit
   - **Recovered Email Id's**
   - Recovered Images
   - Hiding a Fuse File System using Rootkit-1
   - Hiding a Fuse File System using Rootkit-2
6. Summary

Recovered Email Id's

Very Few artifacts were Recovered after the device was factory reset.



Report Navigation

- Case Summary
- EXIF Metadata (6)
- Extension Mismatch Detected (2)
- Keyword Hits (201)
- Tagged Files (0)
- Tagged Results (0)
- Thumbnails (0)

Keyword Hits

- Email Addresses

Email Addresses

+linux-ext4@vger.kernel.org

Preview	Source File	Tags
will be re" +«linux-ext4@vger.kernel.org« if you think	/img_device_image.dd/vol_vol44/Unalloc_234_106299392_116744192	
will be re" +«linux-ext4@vger.kernel.org« if you think	/img_device_image.dd/vol_vol52//$Unalloc/Unalloc_3481_2120318976_2382364672	

-593181219hazrasudip9@gmail.com

Preview	Source File	Tags
-570722657voice -«593181219hazrasudip9@gmail.com« -59318121	/img_device_image.dd/vol_vol53/Unalloc_3469_2382364672_7818165760	
-570722657voice -«593181219hazrasudip9@gmail.com« -59318121	/img_device_image.dd/vol_vol53/$CarvedFiles/f0624336.sqlite	
offers_cache-«593181219hazrasudip9@gmail.com« /storage/	/img_device_image.dd/vol_vol53/$CarvedFiles/f0628504.sqlite	

000000000000000@mycdma.cn

Preview	Source File	Tags
1234567890 «000000000000000@mycdma.cn« hs6n mjJ6 C:r7	/img_device_image.dd/vol_vol4/image/modem.b21	

1+@f.tley

Preview	Source File	Tags

To return to your computer, move the mouse pointer outside or press Ctrl-Alt.

Figure : Recovered Email Id's

1 Introduction
- Why Android
- Why Mobile Forensics is Important
- Problem Scenario

2 Background
- Types of Mobile Forensics Investigation
- Forensic Support Framework
- Shortfalls
- Possible Solutions
- File System in User Space
- Cloud File System
- Rootkits

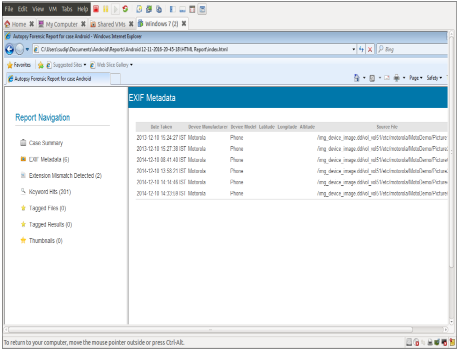
3 Problem Definition

4 Solution Outline
- Modules
- Cloud Storage API's

5 Implementation
- SleuthKit Forensics Toolkit
- Recovered Email Id's
- **Recovered Images**
- Hiding a Fuse File System using Rootkit-1
- Hiding a Fuse File System using Rootkit-2

6 Summary

Recovered Images



Autopsy Forensic Report for case Android - Windows Internet Explorer

C:\Users\sudip\Documents\Android\Reports\Android 12-11-2016-20-45-18\HTML Report\index.html

Autopsy Forensic Report for case Android

Report Navigation

- Case Summary
- EXIF Metadata (6)
- Extension Mismatch Detected (2)
- Keyword Hits (201)
- Tagged Files (0)
- Tagged Results (0)
- Thumbnails (0)

EXIF Metadata

Date Taken	Device Manufacturer	Device Model	Latitude	Longitude	Altitude	Source File
2013-12-10 15:24:27 IST	Motorola	Phone				/img_device_image.dd/vol_vol51/etc/motorola/MotoDemo/Picture
2013-12-10 15:27:38 IST	Motorola	Phone				/img_device_image.dd/vol_vol51/etc/motorola/MotoDemo/Picture
2014-12-10 08:41:40 IST	Motorola	Phone				/img_device_image.dd/vol_vol51/etc/motorola/MotoDemo/Picture
2014-12-10 13:58:21 IST	Motorola	Phone				/img_device_image.dd/vol_vol51/etc/motorola/MotoDemo/Picture
2014-12-10 14:14:46 IST	Motorola	Phone				/img_device_image.dd/vol_vol51/etc/motorola/MotoDemo/Picture
2014-12-10 14:33:59 IST	Motorola	Phone				/img_device_image.dd/vol_vol51/etc/motorola/MotoDemo/Picture

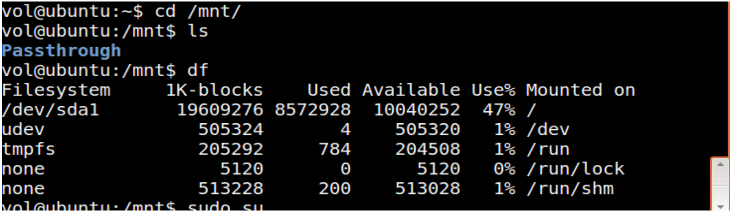
To return to your computer, move the mouse pointer outside or press Ctrl-Alt.

Figure : Recovered Images

1. Introduction
   - Why Android
   - Why Mobile Forensics is Important
   - Problem Scenario
2. Background
   - Types of Mobile Forensics Investigation
   - Forensic Support Framework
   - Shortfalls
   - Possible Solutions
   - File System in User Space
   - Cloud File System
   - Rootkits
3. Problem Definition
4. Solution Outline
   - Modules
   - Cloud Storage API's
5. Implementation
   - SleuthKit Forensics Toolkit
   - Recovered Email Id's
   - Recovered Images
   - **Hiding a Fuse File System using Rootkit-1**
   - Hiding a Fuse File System using Rootkit-2
6. Summary

Hiding a Fuse File System using Rootkit

The rootkit was implemented in Ubuntu 12.04 32 bit, It highjackes the write system call and filter out the file system name and returns the output.



```
vol@ubuntu:~$ cd /mnt/
vol@ubuntu:/mnt$ ls
Passthrough
vol@ubuntu:/mnt$ df
Filesystem     1K-blocks    Used Available Use% Mounted on
/dev/sda1       19609276 8572928  10040252  47% /
udev              505324       4    505320   1% /dev
tmpfs             205292     784    204508   1% /run
none                5120       0      5120   0% /run/lock
none              513228     200    513028   1% /run/shm
vol@ubuntu:/mnt$ sudo su
```

Figure : Normal Execution of df Command

1 Introduction
- Why Android
- Why Mobile Forensics is Important
- Problem Scenario

2 Background
- Types of Mobile Forensics Investigation
- Forensic Support Framework
- Shortfalls
- Possible Solutions
- File System in User Space
- Cloud File System
- Rootkits

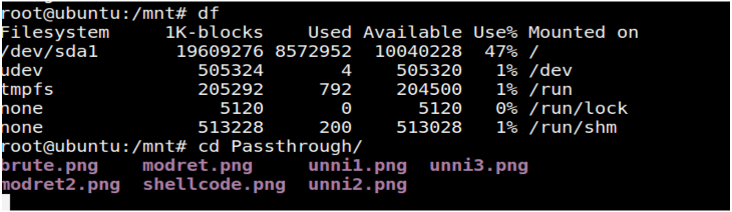
3 Problem Definition

4 Solution Outline
- Modules
- Cloud Storage API's

5 Implementation
- SleuthKit Forensics Toolkit
- Recovered Email Id's
- Recovered Images
- Hiding a Fuse File System using Rootkit-1
- Hiding a Fuse File System using Rootkit-2

6 Summary

Output of df after Rootkit installation



```
root@ubuntu:/mnt# df
Filesystem     1K-blocks    Used Available Use% Mounted on
/dev/sda1       19609276 8572952  10040228  47% /
udev              505324       4    505320   1% /dev
tmpfs             205292     792    204500   1% /run
none                5120       0      5120   0% /run/lock
none              513228     200    513028   1% /run/shm
root@ubuntu:/mnt# cd Passthrough/
brute.png    modret.png     unni1.png  unni3.png
modret2.png  shellcode.png  unni2.png
```

Figure : Execution of df Command after rootkit is installed

Summary

- Process Hiding can also be Implemented Using this Technique
- The Stealth File-system will periodically copy the forensically relevant data from the normal file system
- This data will be moved to the Mounted Cloud Drive and opportunistically uploaded to the cloud server.

Summary

Summary

This Framework can effectively Hide the forensic as well as the cloud file system so that even if the Suspect is connecting to adb to check the internal state , He will not be able to find the hidden File systems.

References I

- *Android forensic support framework*, Aiyappan.P Advisor:Prabhaker Mateti, M.Tech thesis, Amrita Vishwa Vidyapeetham,2015
- *Proactive Forensic Support for Android Devices*, Karthik K. Advisor:Prabhaker Mateti M.Tech thesis, Amrita Vishwa Vidyapeetham,2016
- *Android platform based linux kernel rootkit*, Dong-Hoon You, Bong-Nam Noh, Malicious and Unwanted Software (MALWARE), 2011 6th International Conference ,IEEE