D Y PATIL
DEEMED TO BE
UNIVERSITY
RAMRAO ADIK
INSTITUTE OF TECHNOLOGY
NAVI MUMBAI

RAMRAO ADIK INSTITUTE OF TECHNOLOGY
D. Y. PATIL VIDYANAGAR, SECTOR - 7, NERUL, NAVI MUMBAI - 400 706
WEBSITE: http://www.dypatil.edu/mumbai/rait/

Lab Manual

Third Year Semester-V

Subject: Foundation of Cyber security and Digital Forensics

Major/Minor

Sr. No.	**Module**	**Detailed Content**	**Hours**	**CO Mapping**
1	Cyber Security	1. Study and analysis of different cyber attack and their impact. 2. Analysis of cyber security risk and preventive measures.	4	CO1
2	Cryptography	1. Implementation of any symmetric and asymmetric key algorithm. 2. Implementation of SHA-1 hash algorithm. 3. Generate Digital certificate and sign using Digital signature	6	CO2
3	Cyber Security Safeguards	1. Implement IDS log using snort. 2. Detect and analyze the Security Vulnerabilities of E-commerce services. 3. Detect vulnerabilities of windows system and analyse corresponding attacks. 4. Demonstration of password cracking attack and its countermeasures.	8	CO3
4	Digital Forensics	1. Evidence Collection and Forensic Analysis	2	CO4
5	Digital Forensics tools		4	CO5
6	Cyber Forensics Laws	Study of any laws in Indian IT ACT 2000.	2	CO6

Experiment No. 1A

1. **Aim:** Study and analysis of different cyber-attack and their impact.
2. **What will you learn by performing this experiment?**
   Different types of Cyber-attacks
   Vulnerabilities and threats

3. **Software Required:**Windows

4. Part 1: Conduct search of high profile cyberattacks: Using your favorite search engine conduct a search for each of the cyberattacks listed below. Your search will likely turn up multiple results ranging from news articles to technical articles.

   a. The Stuxnet Virus
   b. Marriott data breach
   c. United Nations data breach
   d. Microsoft customer support database breach
   e. Lifelabs data breach

5. Part 2: Write an analysis of a cyberattack.

   a. Who were the victims of the attacks?

   b. What technologies and tools were used in the attack?

   c. When did the attack happen within the network?

   d. What systems were targeted?

   e. What was the motivation of the attackers in this case? What did they hope to achieve?

   f. What was the outcome of the attack? (stolen data, ransom, system damage, etc.)

6. Conclusion: In this, we have learned about different cyber-attacks and their impact.

Experiment No. 1B

1. **Aim:** Analysis of cyber security risk and preventive measures.
2. **What will you learn by performing this experiment?**
   Different Cyber risk
   Use of Microsoft Threat monitoring tool

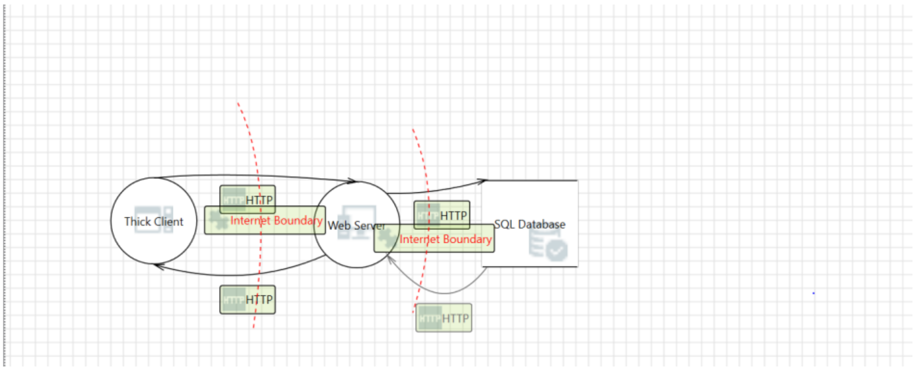
3. **Software Required:** Windows, Microsoft Threat Modeling Tool 2016

4. **Theory:**
   A vulnerability is a weakness/flaw that can be exploited by cybercriminals to gain unauthorized access to a computer system.

   Threat is a process that magnifies the likelihood of a negative event, such as the exploit of a vulnerability.

   Risk is the potential for loss, damage or destruction of assets or data caused by a cyber threat.
   Risk= Vulnerabilities X Threat

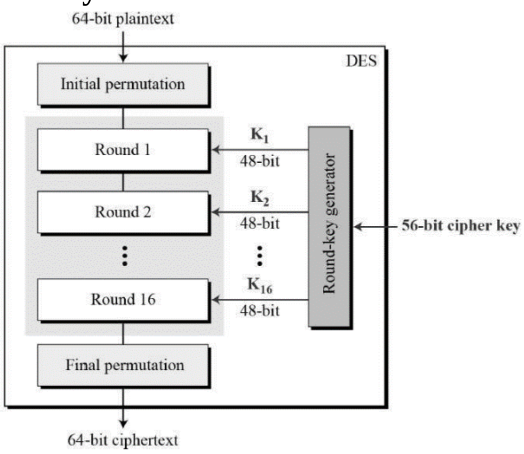
5. Design your network using Microsoft Threat Modelling tool 2016.



6. Observe the threats and generate report. Also write solution to prevent threats for any two threats.
7. **Conclusion:** In this way we have learnt analysis of cyber risk and its preventive measures.

Experiment No. 2A

1. **Aim:** Design a application which take a input a string, encrypt it using DES and display the output.
2. **What will you learn by performing this experiment?**
   Symmetric key cryptography
3. **Software Required:** Windows
4. **Theory:**



The algorithm process breaks down into the following steps:

a. The process begins with the 64-bit plain text block getting handed over to an initial permutation (IP) function.
b. The initial permutation (IP) is then performed on the plain text.
c. Next, the initial permutation (IP) creates two halves of the permuted block, referred to as Left Plain Text (LPT) and Right Plain Text (RPT).
d. Each LPT and RPT goes through 16 rounds of the encryption process.
e. Finally, the LPT and RPT are rejoined, and a Final Permutation (FP) is performed on the newly combined block.
f. The result of this process produces the desired 64-bit ciphertext.

The encryption process step (step 4, above) is further broken down into five stages:

a. Key transformation
b. Expansion permutation
c. S-Box permutation
d. P-Box permutation
e. XOR and swap

5. Conclusion: In this way we have learnt DES algorithm and its application.

Experiment No. 2B

1. **Aim:** Implement short message/brute force attack and factorisation attack for RSA encryption.
2. **What will you learn by performing this experiment?**
   Asymmetric key cryptography
3. **Software Required:** Windows, python/java
4. **Theory:**

   a. Select two large prime numbers, x and y . The prime numbers need to be large so that they will be difficult for someone to figure out.
   b. Calculate $n = xy$.
   c. Calculate the totient function; $\phi(n)=(x-1)(y-1)$.
   d. Select an integer e, such that e is co-prime to $\phi(n)$ and $1<e<\phi(n)$.
   e. The pair of numbers (n,e) makes up the public key.
   f. Calculate d such that $e.d = 1 \bmod \phi(n)$.
   g. The pair (n,d) makes up the private key.
   h. Given a plaintext P, represented as a number, the ciphertext C is calculated as:
   $C=P^e \bmod n$.
   i. Using the private key (n,d), the plaintext can be found using:
   $P= C^d \bmod n$.

Attacks on RSA:

- Short message attack: In this we assume that attacker knows some blocks of plain text and tries to decode cipher text with the help of that.
- Factorisation attack: If attacker will able to know P and Q using N, then he could find out value of private key.

5. Conclusion: In this way we have learnt RSA algorithm and cryptanalysis on RSA.

Experiment No. 3

1. **Aim:** Implement SHA-1 algorithm.
2. **What will you learn by performing this experiment?**
   Hashing algorithm

3. **Software Required:** Windows, python/java
4. **Theory:**

   - Padding: Length of the message is 64 bits short of multiple of 512 after adding.
   - Append a 64-bit length value of original message is taken.
   - Divide the input into 512-bit blocks
   - Initialise CV 5-word (160-bit) buffer (A,B,C,D,E) to

     A=01 23 45 67,

     B=89 AB CD EF,

     C=FE DC BA 98,

     D=76 54 32 10,

     E=C3 D2 E1 F0

   - Process Blocks now the actual algorithm begins. message in 16-word (512-bit) chunks:
     - Copy CV into single register for storing temporary intermediate as well as the final results.
     - Divide the current 512-bit blocks into 16 sub-blocks, each consisting of 32 bits.
     - Has No. Of Rounds=4, each round consisting of 20 bit /step iteration operations on message block & buffer
     - expand 16 words into 80 words(20*4) by mixing & shifting.K[t] constant= Where t=0 to 79
     - Form new buffer value by adding output to input.
   - Output hash value is the final buffer value

5. Conclusion: In this way we have learnt SHA algorithm.

Experiment No. 4

1. **Aim:** Generate Digital certificate and sign using Digital signature.
2. **What will you learn by performing this experiment?**
   Hashing algorithm

3. **Software Required:** Linux, Java keytool and keystore
4. **Theory:** In the "normal" process, two people are usually involved:
   1) the person who wants to share their public key, and
   2) the person who wants to use the first person's public key.

   Usually the sender does these tasks:

   - Create the private key keystore file.
   - Export the certificate file from the private key keystore.
   - Sends the certificate to the second person.

   Then, the second person normally does this task:

   - Imports the certificate from the first person into their public key keystore.

   Demonstration of following keytool tasks:

   - How to create a keystore that contains a private key.
   - How to create a temporary certificate from that private keystore.
   - How to use that certificate to generate a public key keystore.
   - How to query and verify your keystores with the keytool command.

   Steps as follows:

   I. Create private key and keystore: To get started, the first thing we need to do is create a private key keystore. This is going to be a file on your filesystem, and I'm going to name mine privateKey.store. To create this “private key keystore,” run the following keytool command:
      $ keytool -genkey -alias rait -keystore privateKey.store
      This keytool command can be read as:
      - I want to generate a new private key (genkey)
      - I want to create an alias for this key named "rait"
      - I want to store this information in the file named privateKey.store
      - The password for accessing the keystore file is "rait123".
      - The password for my rait alias is "dypatil".

After you issue this command, keytool prompts you with the following questions. I have provided my own example answers to these prompts so you can see exactly how this works.

II. Generate a temporary certificate file: Remember that our end game is to generate a keystore that contains our public key. To do that, we have to take an intermediate step of creating a "certificate file" from our private keystore. To create this certificate file, use this keytool command:

```
$ keytool -export -alias rait -file certfile.cer -keystore privateKey.store
```

This command can be read like this: "Export the information for the alias 'rait' to the file named 'certfile.cer,' getting the information you need from the file named privateKey.store."

```
Enter keystore password:  rait123
Certificate stored in file <certfile.cer>
```

III. Import this certificate into a new public keystore: Now that you have this intermediate certificate file, you can create your public key keystore file from it, using this command:

```
$ keytool -import -alias publicFtpCert -file certfile.cer -keystore publicKey.store
```

This command can be read as: "Import the alias named 'publicFtpCert' from the file named certfile.cer, and store this information in the file named publicKey.store."

IV. Also, at this point you no longer need the intermediate certificate file, so you can delete it:

```
$ rm certfile.cer
```

V. How to view information about a keystore (keytool list): Technically that's all you need to know to (a) create a private keystore, (b) export a certificate for an alias in your private keystore, and (c) import that certificate into your keystore of known public certificates, but ... it's also very nice to be able to query a keystore to see what it contains. To do that, you use the "list" option of the keytool command, like this

```
$ keytool -list -v -keystore privateKey.store
```

5. Conclusion: In this way we have learnt generation of Digital Certificate.

Experiment No. 5

1. **Aim:** Demonstration of Password Cracking attack and its Countermeasures.
2. **What will you learn by performing this experiment?**
    - Ways of Password cracking
    - Countermeasures
3. **Software Required:** Windows, vmware16 pro/virtualbox, Kali Linux
4. **Theory:** Password cracking is the process of using an application program to identify an unknown or forgotten password to a computer or network resource. It can also be used to help a threat actor obtain unauthorized access to resources.

    Methods of Password Attacks: Password cracking methods, including the following:
    - Brute force. This attack runs through combinations of characters of a predetermined length until it finds the combination that matches the password.
    - Dictionary search. Here, a password cracker searches each word in the dictionary for the correct password. Password dictionaries exist for a variety of topics and combinations of topics, including politics, movies and music groups.
    - Phishing. These attacks are used to gain access to user passwords without the use of a password cracking tool. Instead, a user is fooled into clicking on an email attachment. From here, the attachment could install malware or prompt the user to use their email to sign into a false version of a website, revealing their password.
    - Malware. Similar to phishing, using malware is another method of gaining unauthored access to passwords without the use of a password cracking tool. Malware such as keyloggers, which track keystrokes, or screen scrapers, which take screenshots, are used instead.
    - Rainbow attack. This approach involves using different words from the original password in order to generate other possible passwords. Malicious actors can keep a list called a rainbow table with them. This list contains leaked and previously cracked passwords, which will make the overall password cracking method more effective.
    - Guessing. An attacker may be able to guess a password without the use of tools. If the threat actor has enough information about the victim or the victim is using a common enough password, they may be able to come up with the correct characters.

    Some of password cracking tools are as follows:

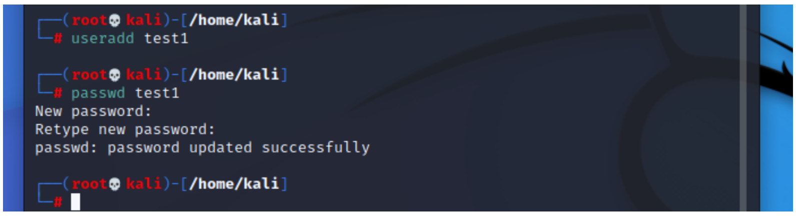
    - Cain and Abel. This password recovery software can recover passwords for Microsoft Windows user accounts and Microsoft Access passwords.

Cain and Abel uses a graphical user interface, making it more user-friendly than comparable tools. The software uses dictionary lists and brute-force attack methods.

- Ophcrack. This password cracker uses rainbow tables and brute-force attacks to crack passwords. It runs on Windows, macOS and Linux.
- John the Ripper. This tool uses a dictionary list approach and is available primarily for macOS and Linux systems. The program has a command prompt to crack passwords, making it more difficult to use than software like Cain and Abel.

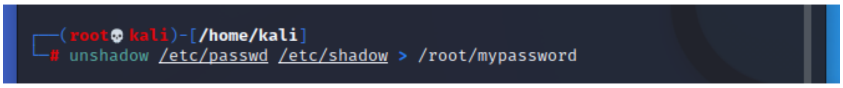
Demonstration of Password cracking with john the ripper in Kali Linux Platform

- Create a new user and set its password
  # useradd test
  # passwd test



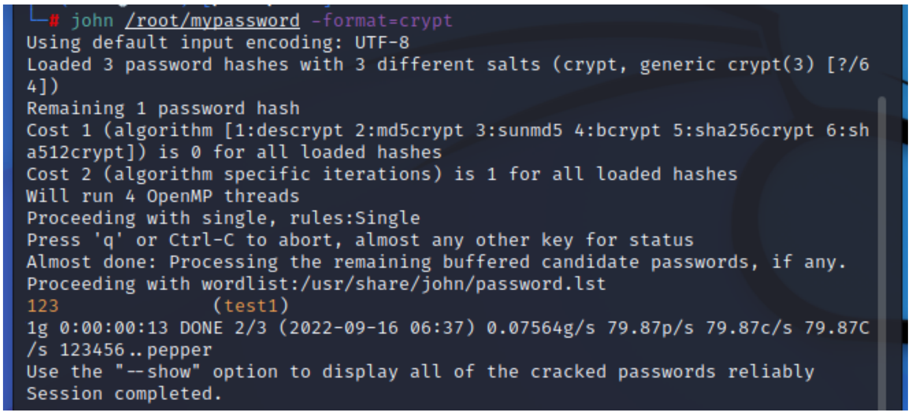
```
┌──(root💀kali)-[/home/kali]
└─# useradd test1

┌──(root💀kali)-[/home/kali]
└─# passwd test1
New password:
Retype new password:
passwd: password updated successfully

┌──(root💀kali)-[/home/kali]
└─# 
```

- Store content of /etc/passwd and /etc/shadow in a file
  # unshadow /etc/passwd /etc/shadow > /root/mypassword



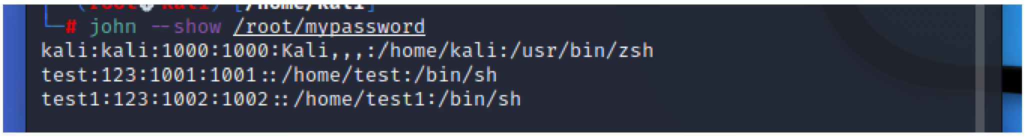
```
┌──(root💀kali)-[/home/kali]
└─# unshadow /etc/passwd /etc/shadow > /root/mypassword
```

- Crack the hash value the password using dictionary based that in stored in /usr/share/john/password.lst
  # john /root/mypassword -format=crypt



```
└─# john /root/mypassword -format=crypt
Using default input encoding: UTF-8
Loaded 3 password hashes with 3 different salts (crypt, generic crypt(3) [?/6
4])
Remaining 1 password hash
Cost 1 (algorithm [1:descrypt 2:md5crypt 3:sunmd5 4:bcrypt 5:sha256crypt 6:sh
a512crypt]) is 0 for all loaded hashes
Cost 2 (algorithm specific iterations) is 1 for all loaded hashes
Will run 4 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
123              (test1)
1g 0:00:00:13 DONE 2/3 (2022-09-16 06:37) 0.07564g/s 79.87p/s 79.87c/s 79.87C
/s 123456..pepper
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

- Display the cracked password
  # john –show /root/mypassword



```
┌──(root💀kali)-[/home/kali]
└─# john --show /root/mypassword
kali:kali:1000:1000:Kali,,,:/home/kali:/usr/bin/zsh
test:123:1001:1001::/home/test:/bin/sh
test1:123:1002:1002::/home/test1:/bin/sh
```

Countermeasures:

- Multi-factor authentication. Using a physical token (like a Yubikey) or a personal device (like a mobile phone) to authenticate users ensures that passwords are not the sole gate to access.
- Remote access. Using a smart remote access platform like OneLogin means that individual websites are no longer the source of user trust. Instead, OneLogin ensures that the user's identity is confirmed, then logs them in.
- Biometrics. A malicious actor will find it very difficult to replicate your fingerprint or facial shape. Enabling biometric authentication turns your password into only one of several points of trust that a hacker needs to overcome.
- Alphanumeric: Combine letters and a variety of characters. Using numbers and special characters, such as periods and commas, increases the number of possible combinations.
- Avoid reusing a password. If a password is cracked, then a person with malicious intent could use that same password to easily access other password-protected accounts the victim owns.
- Pay attention to password strength indicators. Some password-protected systems include a password strength meter, which is a scale that tells users when they have created a strong password.
- Avoid easy-to-guess phrases and common passwords. Weak passwords can be a name, a pet's name or a birthdate -- something personally identifiable. Short and easily predictable patterns, like 123456, password or qwerty, also are weak passwords.
- Use encryption. Passwords stored in a database should be encrypted.

5. Conclusion: In this way we have learnt Password Cracking methods and its countermeasures.

Experiment No. 6

1. **Aim:** Detect and analyse the Security Vulnerabilities of E-commerce services.
2. **What will you learn by performing this experiment?**
   - Detection and analyse the vulnerabilities of ecommerce sites.
   - Tools required for detecting the vulnerabilities.
3. **Software Required:** Windows, Vmware, Kali Linux, Nikto , Vega
4. **Theory:**

   Major Ecommerce Security Threats & Issues
   - Financial Frauds
   - Spam
   - Phishing
   - Bots
   - DDoS Attacks
   - Brute Force Attacks
   - SQL Injections
   - XSS
   - Trojan Horses

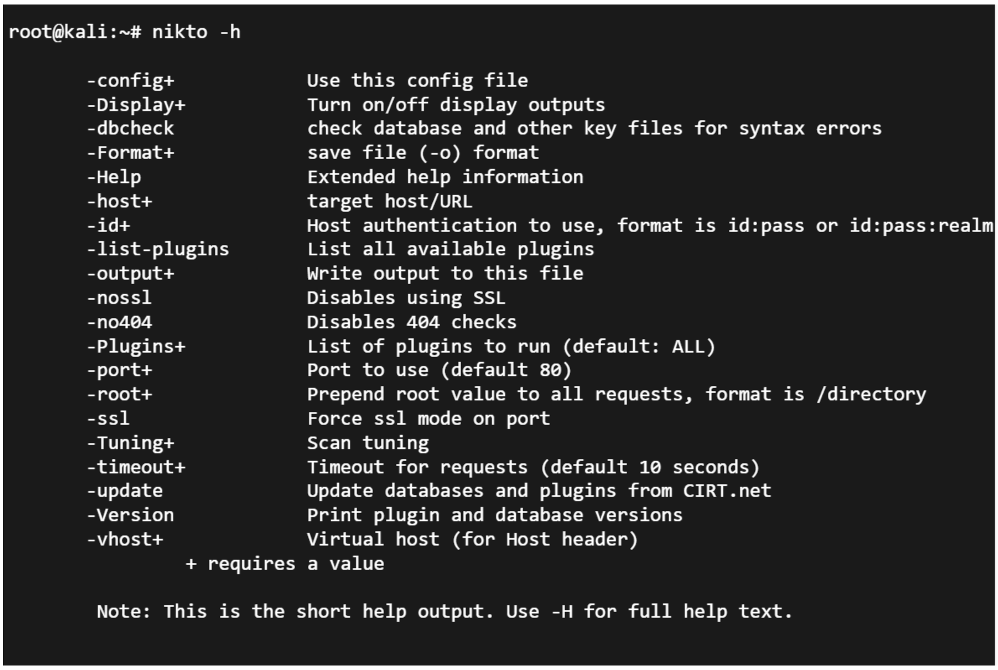
   Ecommerce Security Solutions

   - Switch to HTTPS
   - Secure Your Servers and Admin Panels
   - Payment Gateway Security
   - Antivirus and Anti-Malware Software
   - Use Firewalls
   - Secure your website with SSL certificates
   - Employ Multi-Layer Security
   - Ecommerce Security Plugins
   - Backup Your Data
   - Stay Updated
   - Opt for a Solid Ecommerce Platform
   - Train Your Staff Better
   - Keep an Eye out for Malicious Activity
   - Educate Your Clients

   Nikto: Nikto is an open source web server and web application scanner. Nikto can perform comprehensive tests against web servers for multiple security threats, including over 6700 potentially dangerous files/programs. Nikto can also perform checks for outdated web servers software, and version-specific problems.

Demonstration of Nikto:

- **sudo apt install nikto**
- **nikto -h**



```
root@kali:~# nikto -h

       -config+            Use this config file
       -Display+           Turn on/off display outputs
       -dbcheck            check database and other key files for syntax errors
       -Format+            save file (-o) format
       -Help               Extended help information
       -host+              target host/URL
       -id+                Host authentication to use, format is id:pass or id:pass:realm
       -list-plugins       List all available plugins
       -output+            Write output to this file
       -nossl              Disables using SSL
       -no404              Disables 404 checks
       -Plugins+           List of plugins to run (default: ALL)
       -port+              Port to use (default 80)
       -root+              Prepend root value to all requests, format is /directory
       -ssl                Force ssl mode on port
       -Tuning+            Scan tuning
       -timeout+           Timeout for requests (default 10 seconds)
       -update             Update databases and plugins from CIRT.net
       -Version            Print plugin and database versions
       -vhost+             Virtual host (for Host header)
                + requires a value

        Note: This is the short help output. Use -H for full help text.
```

- Sanning a website:
  nikto www.example.com

Vega: Vega is a free and open-source web security scanner and web security testing platform to test the security of web applications. Vega can help you find and validate SQL Injection, Cross-Site Scripting (XSS), inadvertently disclosed sensitive information, and other vulnerabilities. It is written in Java, GUI based, and runs on Linux, OS X, and Windows. Vega can help you find vulnerabilities such as: reflected cross-site scripting, stored cross-site scripting, blind SQL injection, remote file includes, shell injection, and others. Vega also probes for TLS / SSL security settings and identifies opportunities for improving the security of your TLS servers. Vega includes an automated scanner for quick tests and an intercepting proxy for tactical inspection. The Vega scanner finds XSS (cross-site scripting), SQL injection, and other vulnerabilities. Vega can be extended using a powerful API in the language of the web: Javascript.

5. Conclusion: In this way we how to detect and analyse vulnerabilities of ecommerce sites.

D Y PATIL UNIVERSITY — RAMRAO ADIK — INSTITUTE OF TECHNOLOGY NAVI MUMBAI

RAMRAO ADIK INSTITUTE OF TECHNOLOGY
D. Y. PATIL VIDYANAGAR, SECTOR - 7, NERUL, NAVI MUMBAI - 400 706
WEBSITE: http://www.dypatil.edu/mumbai/rait/

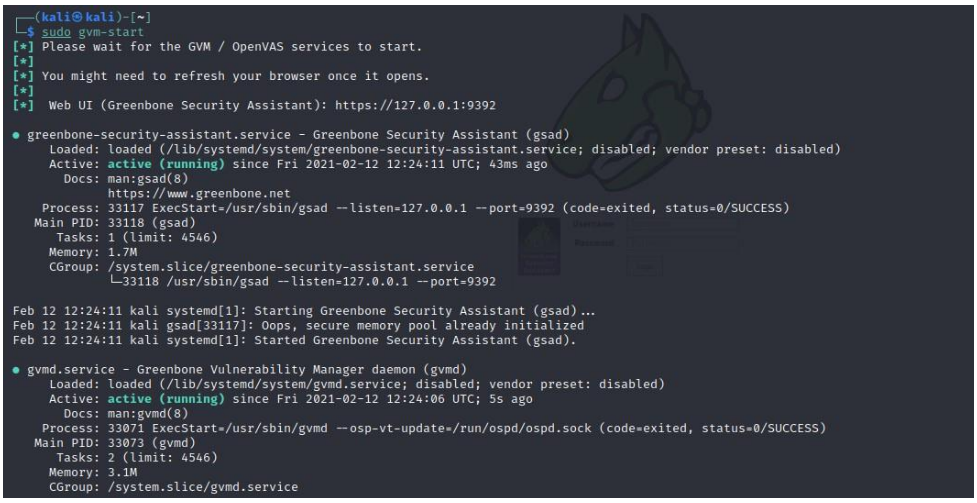
Experiment No. 7

1. **Aim:** Detect vulnerabilities of windows system and analyse corresponding attacks.
2. **What will you learn by performing this experiment?**
   - Detection and analyse the vulnerabilities of Windows.
   - Tools required for detecting the vulnerabilities.
3. **Software Required:** Windows, Vmware, Kali Linux, OpenVas
4. **Theory:**

OpenVAS is a full-featured vulnerability scanner. Its capabilities include unauthenticated and authenticated testing, various high-level and low-level internet and industrial protocols, performance tuning for large-scale scans and a powerful internal programming language to implement any type of vulnerability test.

Demonstration of vulnerability assessment using OpenVas:

- Installation of OpenVas
  sudo apt-get update
  sudo apt-get dist-upgrade
  sudo apt-get install openvas
- Set up gvm for SQLite Database
  sudo gvm-setup *//Note: Remember to note down the password generated during the setup process*
  sudo gvm-start

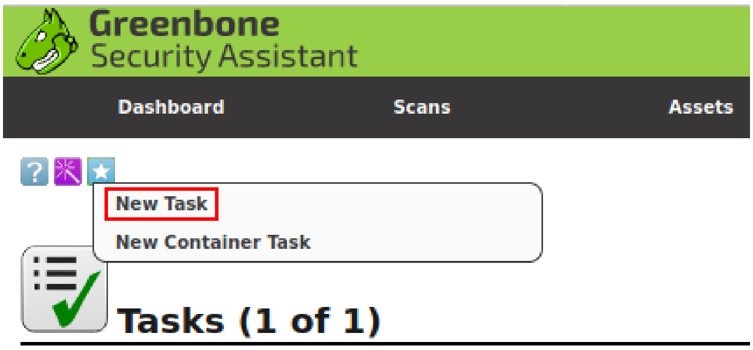


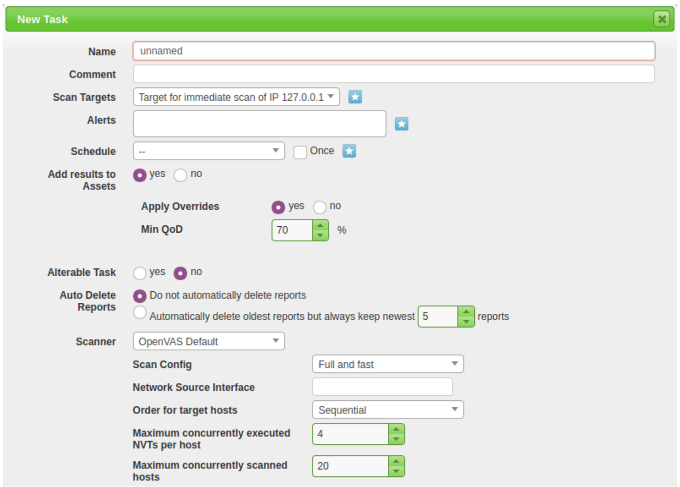
```
┌──(kali㉿kali)-[~]
└─$ sudo gvm-start
[*] Please wait for the GVM / OpenVAS services to start.
[*]
[*] You might need to refresh your browser once it opens.
[*]
[*]  Web UI (Greenbone Security Assistant): https://127.0.0.1:9392

● greenbone-security-assistant.service - Greenbone Security Assistant (gsad)
     Loaded: loaded (/lib/systemd/system/greenbone-security-assistant.service; disabled; vendor preset: disabled)
     Active: active (running) since Fri 2021-02-12 12:24:11 UTC; 43ms ago
       Docs: man:gsad(8)
             https://www.greenbone.net
    Process: 33117 ExecStart=/usr/sbin/gsad --listen=127.0.0.1 --port=9392 (code=exited, status=0/SUCCESS)
   Main PID: 33118 (gsad)
      Tasks: 1 (limit: 4546)
     Memory: 1.7M
     CGroup: /system.slice/greenbone-security-assistant.service
             └─33118 /usr/sbin/gsad --listen=127.0.0.1 --port=9392

Feb 12 12:24:11 kali systemd[1]: Starting Greenbone Security Assistant (gsad)...
Feb 12 12:24:11 kali gsad[33117]: Oops, secure memory pool already initialized
Feb 12 12:24:11 kali systemd[1]: Started Greenbone Security Assistant (gsad).

● gvmd.service - Greenbone Vulnerability Manager daemon (gvmd)
     Loaded: loaded (/lib/systemd/system/gvmd.service; disabled; vendor preset: disabled)
     Active: active (running) since Fri 2021-02-12 12:24:06 UTC; 5s ago
       Docs: man:gvmd(8)
    Process: 33071 ExecStart=/usr/sbin/gvmd --osp-vt-update=/run/ospd/ospd.sock (code=exited, status=0/SUCCESS)
   Main PID: 33073 (gvmd)
      Tasks: 2 (limit: 4546)
     Memory: 3.1M
     CGroup: /system.slice/gvmd.service
```

- Open your browser and navigate to http://localhost:9392
- Use the **username as admin and the password generated** in the setup process**.**

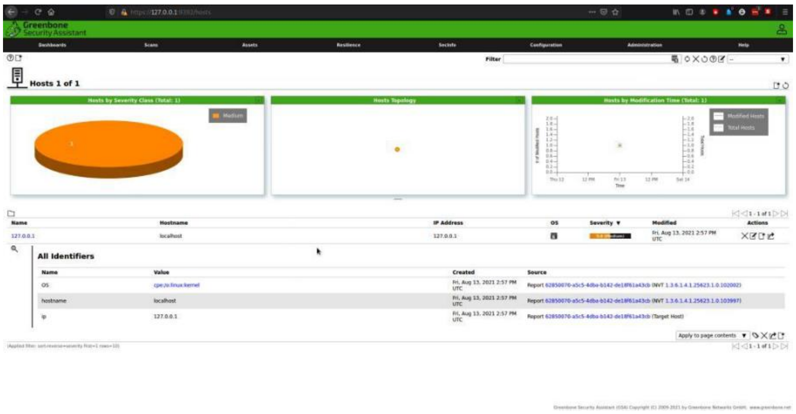
- Scanning Windows system

navigating to **Scans** > **Tasks**>**New Task**




New Task

Name	unnamed
Comment	
Scan Targets	Target for immediate scan of IP 127.0.0.1
Alerts	
Schedule	-- □ Once
Add results to Assets	● yes ○ no
Apply Overrides	● yes ○ no
Min QoD	70 %
Alterable Task	○ yes ● no
Auto Delete Reports	● Do not automatically delete reports ○ Automatically delete oldest reports but always keep newest 5 reports
Scanner	OpenVAS Default
Scan Config	Full and fast
Network Source Interface	
Order for target hosts	Sequential
Maximum concurrently executed NVTs per host	4
Maximum concurrently scanned hosts	20



Name	Hostname	IP Address	OS	Severity	Modified
127.0.0.1	localhost	127.0.0.1		(Medium)	Fri, Aug 13, 2021 2:57 PM UTC

All Identifiers

Name	Value	Created	Source
OS	cpe:/o:linux:kernel	Fri, Aug 13, 2021 2:57 PM UTC	Report 62850070-a5c5-4dba-b142-de18f61a43cb (NVT 1.3.6.1.4.1.25623.1.0.102002)
hostname	localhost	Fri, Aug 13, 2021 2:57 PM UTC	Report 62850070-a5c5-4dba-b142-de18f61a43cb (NVT 1.3.6.1.4.1.25623.1.0.103997)
ip	127.0.0.1	Fri, Aug 13, 2021 2:57 PM UTC	Report 62850070-a5c5-4dba-b142-de18f61a43cb (Target Host)

Greenbone Security Assistant (GSA) Copyright (C) 2009-2021 by Greenbone Networks GmbH, www.greenbone.net

5. **Conclusion**: In this way we how to detect and analyse vulnerabilities in Windows using OpenVas.

Experiment No. 8

1. **Aim:** Implement IDS log using snort.
2. **What will you learn by performing this experiment?**
   - Working of Snort.
   - Analysis of Snort logs.
3. **Software Required:** Windows, Vmware, Ubuntu16.
4. **Theory:**

Snort is the foremost Open Source Intrusion Prevention System (IPS) in the world. Snort IPS uses a series of rules that help define malicious network activity and uses those rules to find packets that match against them and generates alerts for users.

Snort can be deployed inline to stop these packets, as well. Snort has three primary uses: As a packet sniffer like tcpdump, as a packet logger — which is useful for network traffic debugging, or it can be used as a full-blown network intrusion prevention system. Snort can be downloaded and configured for personal and business use alike.

Implementation of Snort in Ubuntu:

- Installation of the required libraries for Snort
```
sudo apt install -y gcc libpcre3-dev zlib1g-dev libluajit-5.1-dev libpcap-dev openssl libssl-dev libnghttp2-dev libdumbnet-dev bison flex libdnet autoconf libtool
```
- Installing from the source
```
mkdir ~/snort_src && cd ~/snort_src
wget https://www.snort.org/downloads/snort/daq-2.0.7.tar.gz
tar -xvzf daq-2.0.7.tar.gz
cd daq-2.0.7
autoreconf -f -i
./configure && make && sudo make install
cd ~/snort_src
```
- Download latest Snort package from https://www.snort.org/downloads
```
wget https://www.snort.org/downloads/snort/snort-2.9.16.tar.gz
```
- Extract the source and change into the new directory
```
tar -xvzf snort-2.9.16.tar.gz
cd snort-2.9.16
```
- Configure the installation with sourcefire enabled, run make and make install.
```
./configure --enable-sourcefire && make && sudo make install
```

- Update the shared libraries.

  ```
  sudo ldconfig
  ```

- Create the folder structure to house the Snort configuration

  ```
  sudo mkdir -p /etc/snort/rules
  sudo mkdir /var/log/snort
  sudo mkdir /usr/local/lib/snort_dynamicrules
  ```

- Create new files for local rules

  ```
  sudo touch /etc/snort/rules/local.rules
  ```

- Copy the configuration files

  ```
  sudo cp ~/snort_src/snort-2.9.16/etc/*.conf* /etc/snort
  sudo cp ~/snort_src/snort-2.9.16/etc/*.map /etc/snort
  ```

- Configuring the network and rule sets

  ```
  sudo nano /etc/snort/snort.conf
  ```

  Find these sections shown below in the configuration file and change the parameters to reflect the examples here.

  ```
  # Setup the network addresses you are protecting
  ipvar HOME_NET server_public_ip/24
  # Set up the external network addresses. Leave as "any" in most situations
  ipvar EXTERNAL_NET !$HOME_NET
  # Path to your rules files (this can be a relative path)
  var RULE_PATH /etc/snort/rules
  var SO_RULE_PATH /etc/snort/so_rules
  var PREPROC_RULE_PATH /etc/snort/preproc_rules
  # Set the absolute path appropriately
  #var WHITE_LIST_PATH /etc/snort/rules
  #var BLACK_LIST_PATH /etc/snort/rules
  ```

  In the same snort.conf file, scroll down to the section 6 and set the output for unified2 to log under filename of snort.log like below.

  ```
  # unified2
  # Recommended for most installs
  output unified2: filename snort.log, limit 128
  ```

  Lastly, scroll down towards the bottom of the file to find the list of included rule sets. You will need to uncomment the local.rules to allow Snort to load any custom rules.

  ```
  include $RULE_PATH/local.rules
  ```

  Once you are done with the configuration file, save the changes and exit the editor.

- Validating settings

  ```
  sudo snort -T -c /etc/snort/snort.conf
  ```

- Testing the configuration

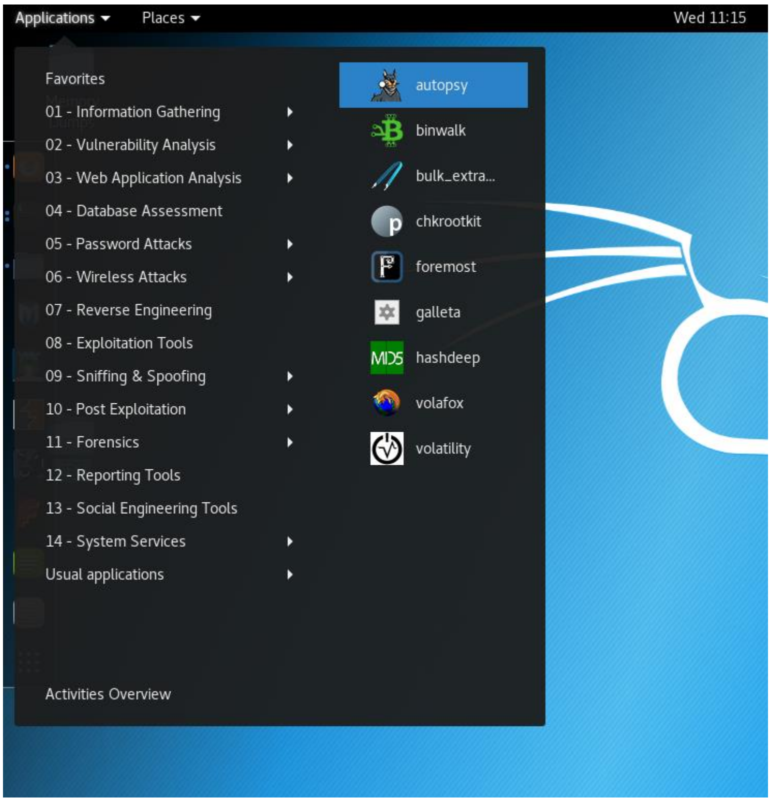
  ```
  sudo nano /etc/snort/rules/local.rules
  ```

- Add the following line to the file.
  alert icmp any any -> $HOME_NET any (msg:"ICMP test"; sid:10000001; rev:001;)
- Start Snort with -A console
  sudo snort -A console -i eth0 -u snort -g snort -c /etc/snort/snort.conf

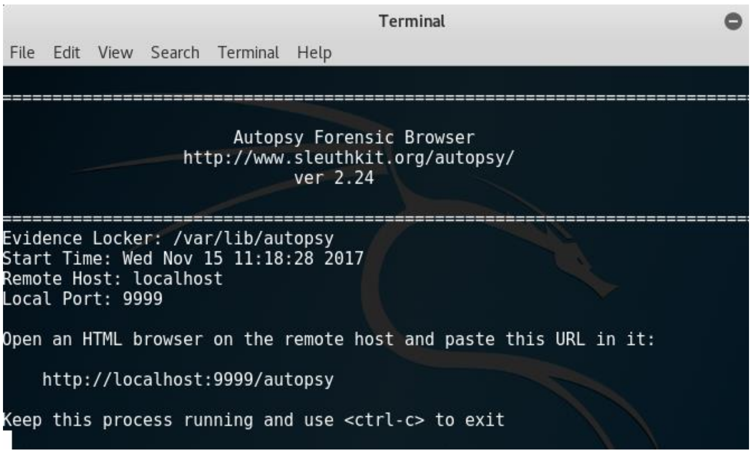
5. **Conclusion**: In this way we learn how to implement Snort and detection of ICMP packets using Snort.

Experiment No. 9

1. **Aim:** Evidence Collection and forensic analysis using Autopsy.
2. **What will you learn by performing this experiment?**
   - Detection and analyse of evidences.
   - Working of Autopsy.
3. **Software Required:** Windows, Autopsy
4. **Theory:** Autopsy® is a digital forensics platform and graphical interface to The Sleuth Kit® and other digital forensics tools. It is used by law enforcement, military, and corporate examiners to investigate what happened on a computer. You can even use it to recover photos from your camera's memory card.
   Demonstration of Autopsy:

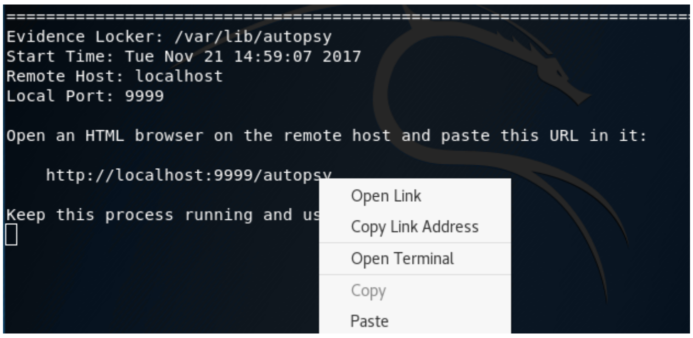


Alternatively, we can click on the Show applications icon (last item in the side menu) and type autopsy into the search bar at the top-middle of the screen and then click on the autopsy icon:



```
============================================================================

                        Autopsy Forensic Browser
                   http://www.sleuthkit.org/autopsy/
                               ver 2.24

============================================================================
Evidence Locker: /var/lib/autopsy
Start Time: Wed Nov 15 11:18:28 2017
Remote Host: localhost
Local Port: 9999

Open an HTML browser on the remote host and paste this URL in it:

    http://localhost:9999/autopsy

Keep this process running and use <ctrl-c> to exit
```

To open the Autopsy browser, position the mouse over the link in the terminal, then right-click and choose Open Link, as seen in the following screenshot:

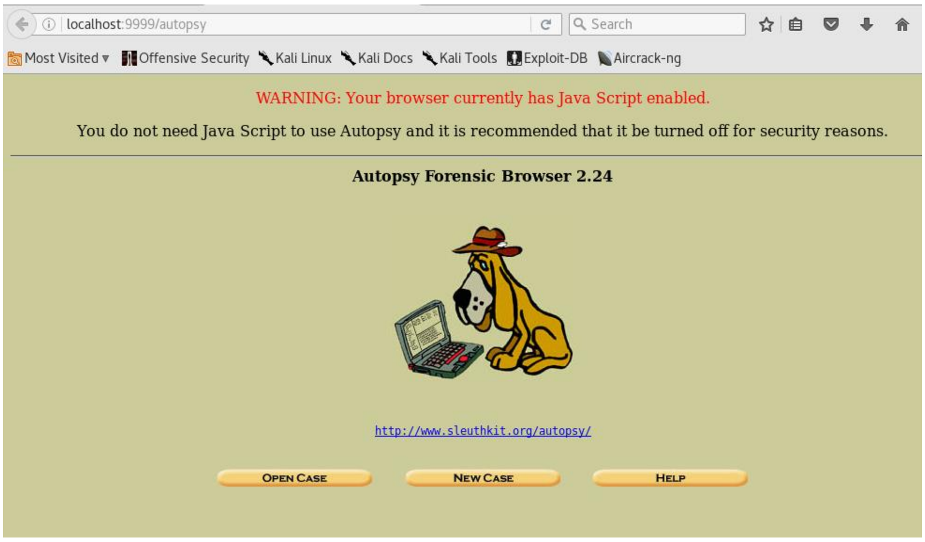


```
======================================================================
Evidence Locker: /var/lib/autopsy
Start Time: Tue Nov 21 14:59:07 2017
Remote Host: localhost
Local Port: 9999

Open an HTML browser on the remote host and paste this URL in it:

    http://localhost:9999/autopsy

Keep this process running and us
```

Creating a new case

To create a new case, follow the given steps:

1. When the Autopsy Forensic Browser opens, investigators are presented with three options.
2. Click on NEW CASE:



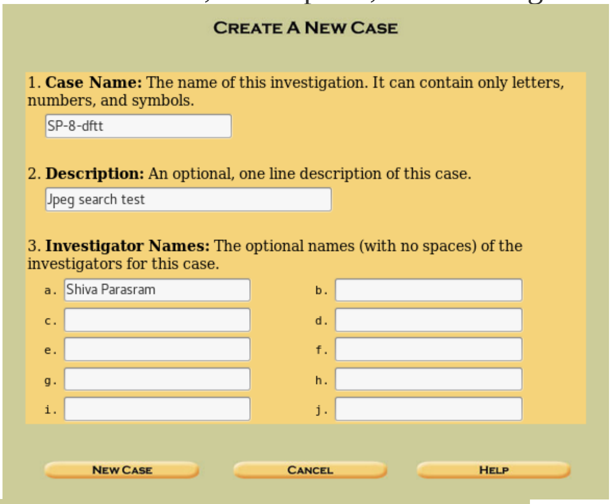
WARNING: Your browser currently has Java Script enabled.

You do not need Java Script to use Autopsy and it is recommended that it be turned off for security reasons.

Autopsy Forensic Browser 2.24

http://www.sleuthkit.org/autopsy/

3. Enter details for the Case Name, Description, and Investigator Names. the Case



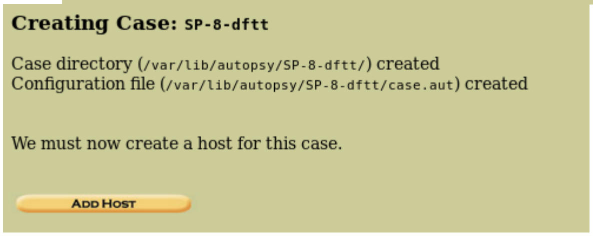
CREATE A NEW CASE

1. **Case Name:** The name of this investigation. It can contain only letters, numbers, and symbols.
SP-8-dftt

2. **Description:** An optional, one line description of this case.
Jpeg search test

3. **Investigator Names:** The optional names (with no spaces) of the investigators for this case.

a.	Shiva Parasram	b.	
c.		d.	
e.		f.	
g.		h.	
i.		j.	



Creating Case: SP-8-dftt

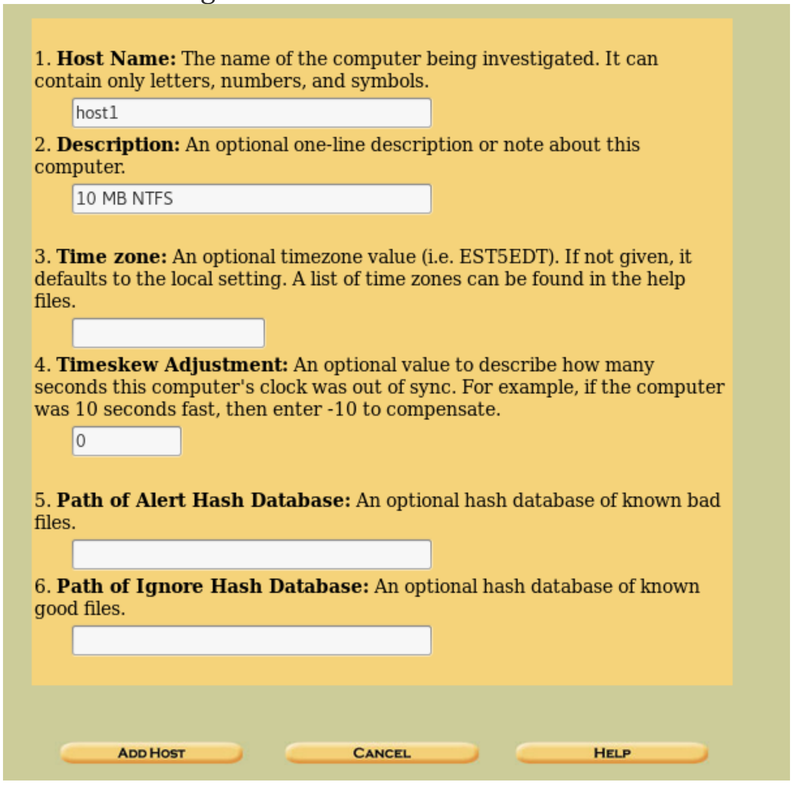
Case directory (/var/lib/autopsy/SP-8-dftt/) created
Configuration file (/var/lib/autopsy/SP-8-dftt/case.aut) created

We must now create a host for this case.

4. Enter the details for the Host Name (name of the computer being investigated) and the Description of the host.

5. Optional settings:
    - Time zone: Defaults to local settings, if not specified
    - Timeskew Adjustment: Adds a value in seconds to compensate for time differences
    - Path of Alert Hash Database: Specifies the path of a created database of known bad hashes

- Path of Ignore Hash Database: Specifies the path of a created database of known good hashes similar to the NIST NSRL:



1. **Host Name:** The name of the computer being investigated. It can contain only letters, numbers, and symbols.
host1

2. **Description:** An optional one-line description or note about this computer.
10 MB NTFS

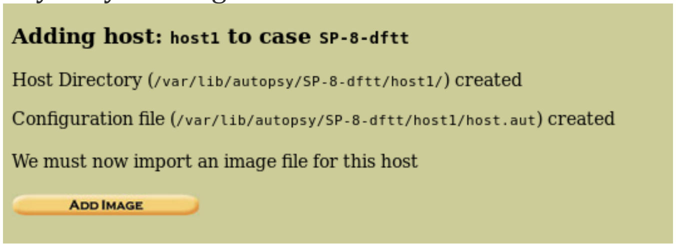
3. **Time zone:** An optional timezone value (i.e. EST5EDT). If not given, it defaults to the local setting. A list of time zones can be found in the help files.

4. **Timeskew Adjustment:** An optional value to describe how many seconds this computer's clock was out of sync. For example, if the computer was 10 seconds fast, then enter -10 to compensate.
0

5. **Path of Alert Hash Database:** An optional hash database of known bad files.

6. **Path of Ignore Hash Database:** An optional hash database of known good files.

6. Click on the ADD HOST button to continue.
7. Once the host is added and directories are created, we add the forensic image we want to analyze by clicking the ADD IMAGE button:



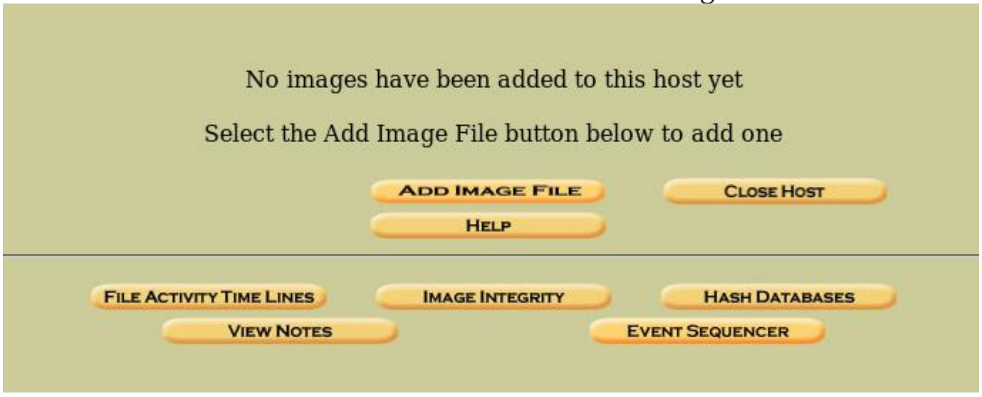
Adding host: `host1` to case `SP-8-dftt`

Host Directory (`/var/lib/autopsy/SP-8-dftt/host1/`) created

Configuration file (`/var/lib/autopsy/SP-8-dftt/host1/host.aut`) created

We must now import an image file for this host

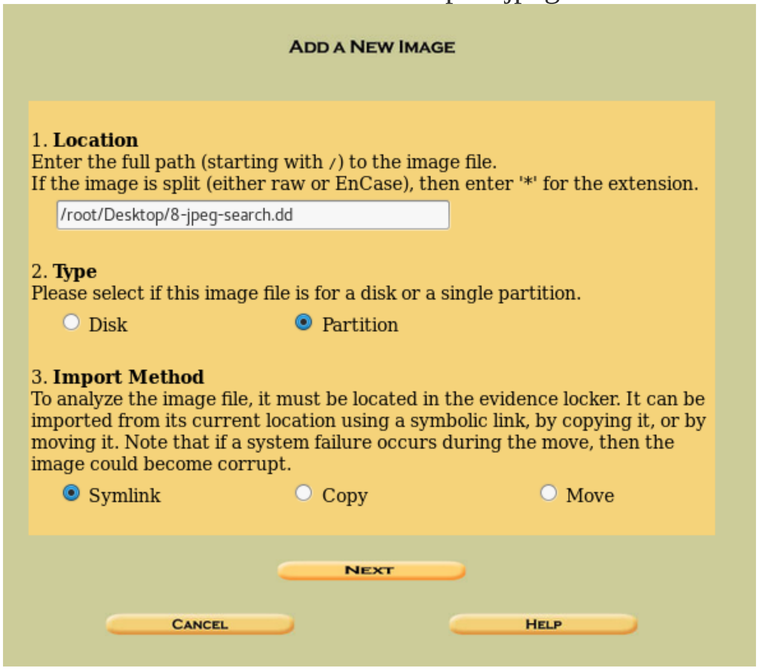
8. Click on the ADD IMAGE FILE button to add the image file:



No images have been added to this host yet

Select the Add Image File button below to add one

9. To import the image for analysis, the full path must be specified. On my machine, I've saved the image file (8-jpeg-search.dd) to the Desktop folder. As such, the location of the file would be /root/Desktop/ 8-jpeg-search.dd.



ADD A NEW IMAGE

1. **Location**
Enter the full path (starting with /) to the image file.
If the image is split (either raw or EnCase), then enter '*' for the extension.
/root/Desktop/8-jpeg-search.dd

2. **Type**
Please select if this image file is for a disk or a single partition.
○ Disk ● Partition

3. **Import Method**
To analyze the image file, it must be located in the evidence locker. It can be imported from its current location using a symbolic link, by copying it, or by moving it. Note that if a system failure occurs during the move, then the image could become corrupt.
● Symlink ○ Copy ○ Move

11. Upon clicking **Next**, the Image File Details are displayed. To verify the integrity of the file, select the radio button for Calculate the hash value for this image, and select the checkbox next to Verify hash after importing?

12. The File System Details section also shows that the image is of a ntfs partition.

13. Click on the ADD button to continue:

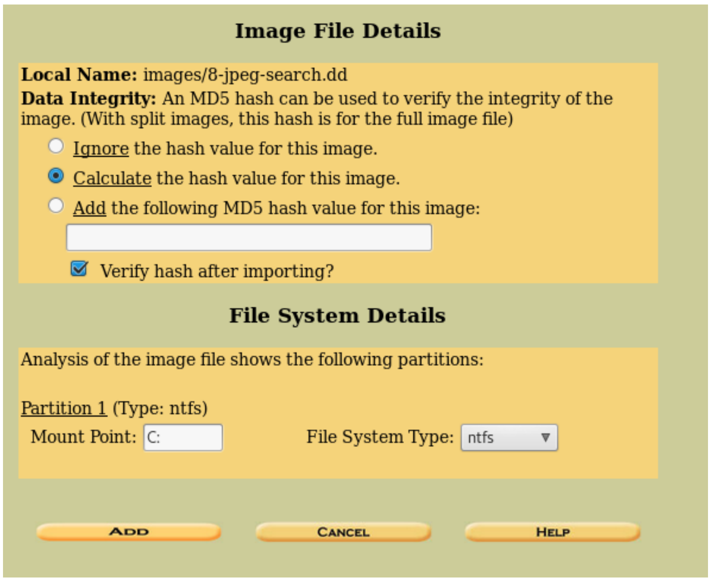


Image File Details

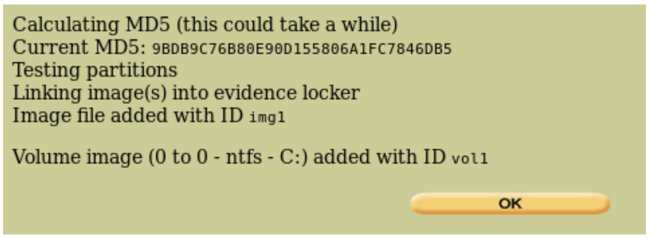
Local Name: images/8-jpeg-search.dd
Data Integrity: An MD5 hash can be used to verify the integrity of the image. (With split images, this hash is for the full image file)
○ Ignore the hash value for this image.
● Calculate the hash value for this image.
○ Add the following MD5 hash value for this image:
☑ Verify hash after importing?

File System Details

Analysis of the image file shows the following partitions:

Partition 1 (Type: ntfs)
Mount Point: C: File System Type: ntfs

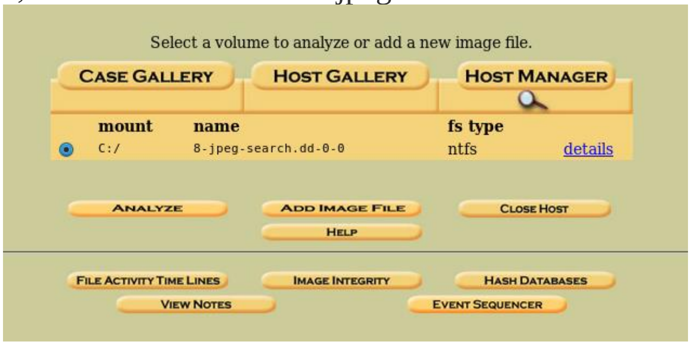
14. After clicking the ADD button in the previous screenshot, Autopsy calculates the MD5 hash and links the image into the evidence locker. Press OK to continue:



Calculating MD5 (this could take a while)
Current MD5: 9BDB9C76B80E90D155806A1FC7846DB5
Testing partitions
Linking image(s) into evidence locker
Image file added with ID img1

Volume image (0 to 0 - ntfs - C:) added with ID vol1

15. At this point, we're just about ready to analyze the image file. If there are multiple cases listed in the gallery area from any previous investigations you may have worked on, be sure to choose the 8-jpeg-search.dd file and case:



Select a volume to analyze or add a new image file.

	mount	name	fs type	
●	C:/	8-jpeg-search.dd-0-0	ntfs	details

16. Before proceeding, we can click on the IMAGE DETAILS option. This screen gives detail such as the image name, volume ID, file format, file system, and also allows for the extraction of ASCII, Unicode, and unallocated data to enhance and provide faster keyword searches. Click on the back button in the browser to return to the previous menu and continue with the analysis:

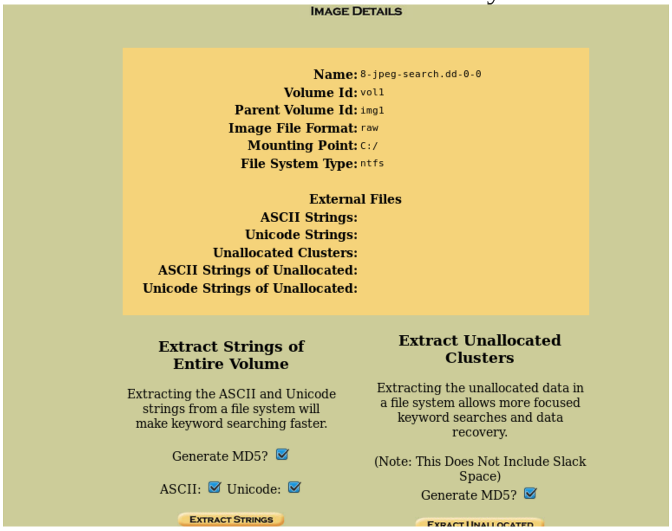


Image Details

Name: 8-jpeg-search.dd-0-0
Volume Id: vol1
Parent Volume Id: img1
Image File Format: raw
Mounting Point: C:/
File System Type: ntfs

External Files
ASCII Strings:
Unicode Strings:
Unallocated Clusters:
ASCII Strings of Unallocated:
Unicode Strings of Unallocated:

Extract Strings of Entire Volume

Extracting the ASCII and Unicode strings from a file system will make keyword searching faster.

Generate MD5? ☑

ASCII: ☑ Unicode: ☑

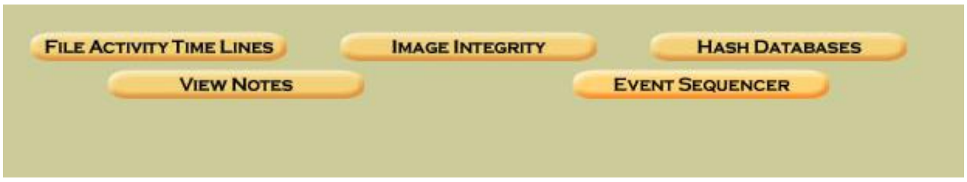
Extract Unallocated Clusters

Extracting the unallocated data in a file system allows more focused keyword searches and data recovery.

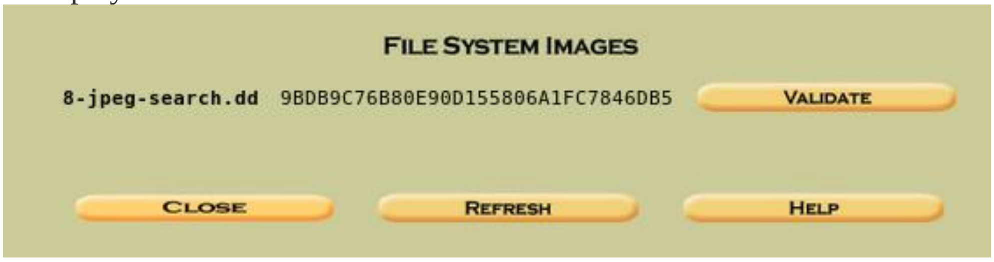
(Note: This Does Not Include Slack Space)

Generate MD5? ☑

17. Before clicking on the ANALYZE button to start our investigation and analysis, we can also verify the integrity of the image by creating an MD5 hash, by clicking on the IMAGE INTEGRITY button:



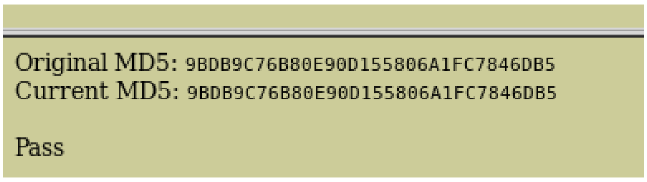
18. After clicking on the IMAGE INTEGRITY button, the image name and hash are displayed. Click on the VALIDATE button to validate the MD5 hash:



FILE SYSTEM IMAGES

8-jpeg-search.dd 9BDB9C76B80E90D155806A1FC7846DB5

19. The validation results are displayed in the lower-left corner of the Autopsy browser window:

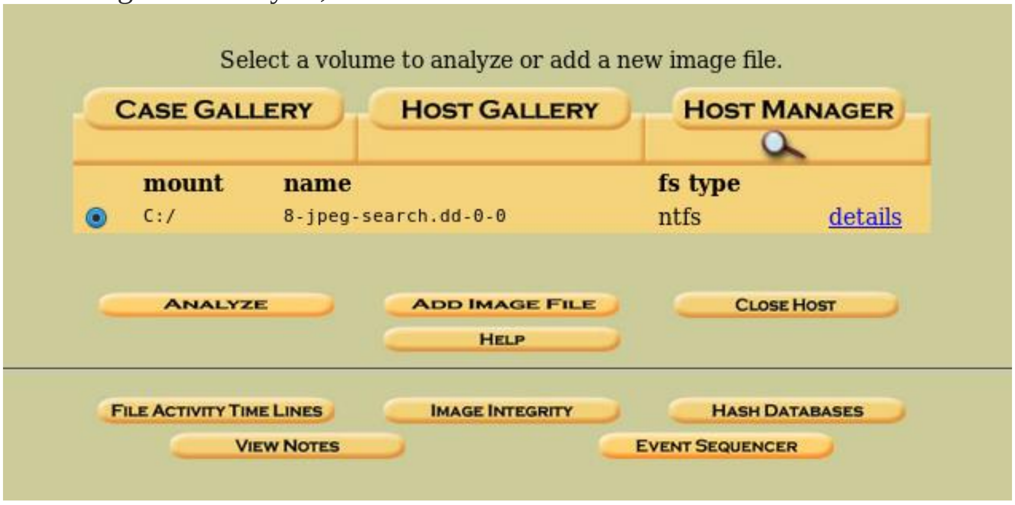


Original MD5: 9BDB9C76B80E90D155806A1FC7846DB5
Current MD5: 9BDB9C76B80E90D155806A1FC7846DB5

Pass

20. We can see that our validation was successful, with matching MD5 hashes displayed in the results. Click on the CLOSE button to continue.

21. To begin our analysis, we click on the ANALYZE button:



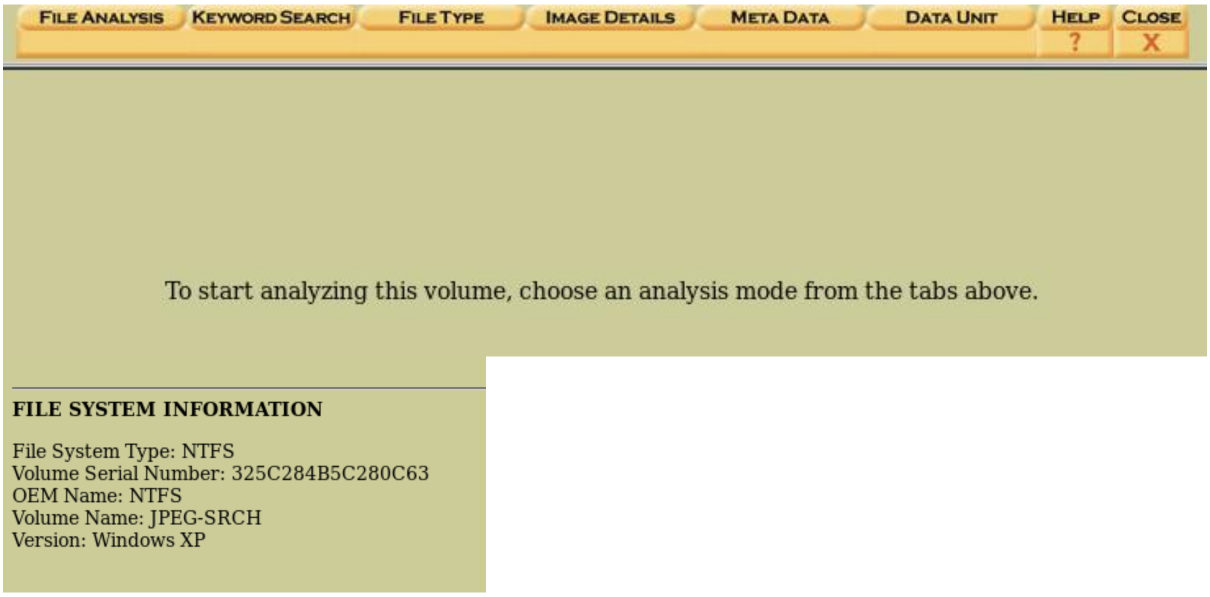
Select a volume to analyze or add a new image file.

	mount	name	fs type	
●	C:/	8-jpeg-search.dd-0-0	ntfs	details

Analysis using Autopsy

Now that we've created our case, added host information with appropriate directories, and added our acquired image, we get to the analysis stage.

After clicking on the ANALYZE button (see the previous screenshot), we're presented with several options in the form of tabs, with which to begin our investigation:

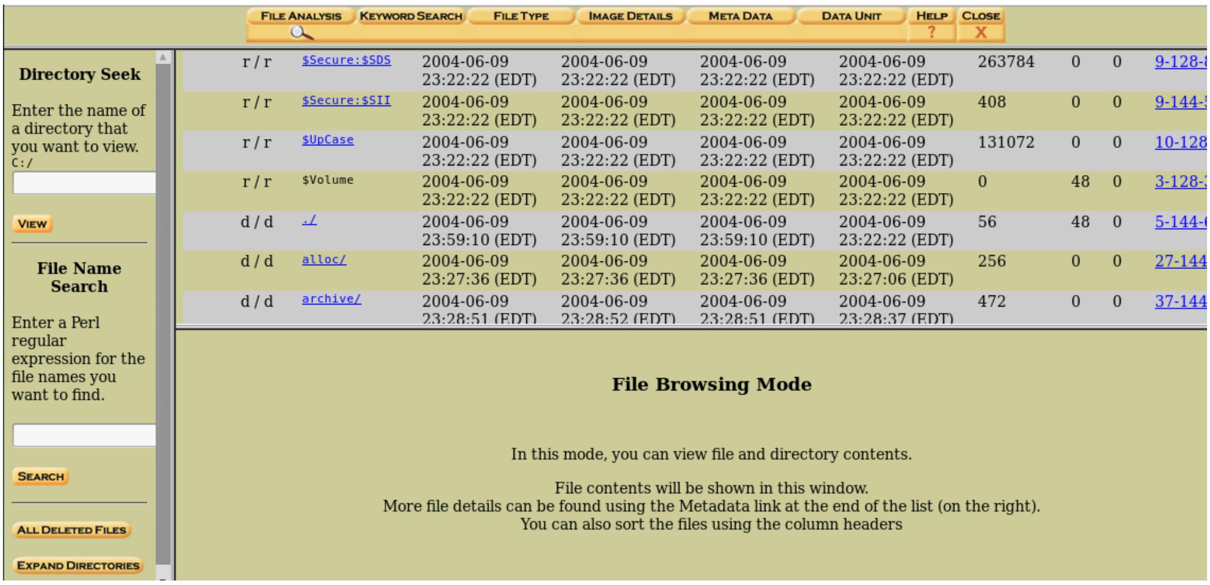


To start analyzing this volume, choose an analysis mode from the tabs above.

FILE SYSTEM INFORMATION

File System Type: NTFS
Volume Serial Number: 325C284B5C280C63
OEM Name: NTFS
Volume Name: JPEG-SRCH
Version: Windows XP

Next, we click on the FILE ANALYSIS tab. This mode opens into File Browsing Mode, which allows the examination of directories and files within the image. Directories within the image are listed by default in the main view area:



Directory Seek

Enter the name of a directory that you want to view.
C:/

File Name Search

Enter a Perl regular expression for the file names you want to find.

r / r	$Secure:$SDS	2004-06-09 23:22:22 (EDT)	2004-06-09 23:22:22 (EDT)	2004-06-09 23:22:22 (EDT)	2004-06-09 23:22:22 (EDT)	263784	0	0	9-128-
r / r	$Secure:$SII	2004-06-09 23:22:22 (EDT)	2004-06-09 23:22:22 (EDT)	2004-06-09 23:22:22 (EDT)	2004-06-09 23:22:22 (EDT)	408	0	0	9-144-
r / r	$UpCase	2004-06-09 23:22:22 (EDT)	2004-06-09 23:22:22 (EDT)	2004-06-09 23:22:22 (EDT)	2004-06-09 23:22:22 (EDT)	131072	0	0	10-128
r / r	$Volume	2004-06-09 23:22:22 (EDT)	2004-06-09 23:22:22 (EDT)	2004-06-09 23:22:22 (EDT)	2004-06-09 23:22:22 (EDT)	0	48	0	3-128-
d / d	./	2004-06-09 23:59:10 (EDT)	2004-06-09 23:59:10 (EDT)	2004-06-09 23:59:10 (EDT)	2004-06-09 23:22:22 (EDT)	56	48	0	5-144-
d / d	alloc/	2004-06-09 23:27:36 (EDT)	2004-06-09 23:27:36 (EDT)	2004-06-09 23:27:36 (EDT)	2004-06-09 23:27:06 (EDT)	256	0	0	27-144
d / d	archive/	2004-06-09 23:28:51 (EDT)	2004-06-09 23:28:52 (EDT)	2004-06-09 23:28:51 (EDT)	2004-06-09 23:28:37 (EDT)	472	0	0	37-144

File Browsing Mode

In this mode, you can view file and directory contents.

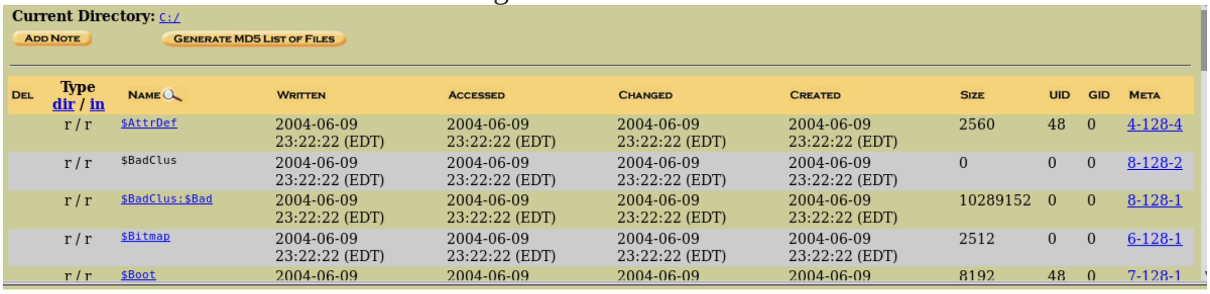
File contents will be shown in this window.
More file details can be found using the Metadata link at the end of the list (on the right).
You can also sort the files using the column headers

In File Browsing Mode, directories are listed with the Current Directory specified as C:/.

For each directory and file, there are fields showing when the item was WRITTEN, ACCESSED, CHANGED, and CREATED, along with its size and META data:WRITTEN: The date and time the file was last written to

- ACCESSED: The date and time the file was last accessed (only the date is accurate)
- CHANGED: The date and time the descriptive data of the file was modified
- CREATED: The data and time the file was created

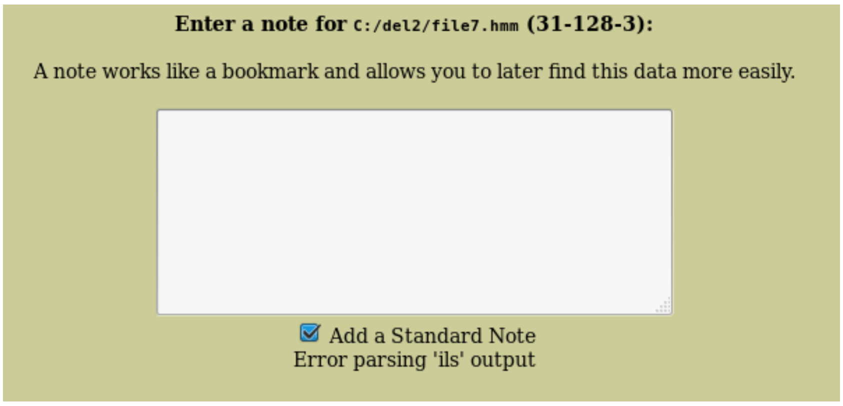
- META: Metadata describing the file and information about the file:



Current Directory: C:/

DEL	Type dir / in	NAME	WRITTEN	ACCESSED	CHANGED	CREATED	SIZE	UID	GID	META
	r / r	$AttrDef	2004-06-09 23:22:22 (EDT)	2004-06-09 23:22:22 (EDT)	2004-06-09 23:22:22 (EDT)	2004-06-09 23:22:22 (EDT)	2560	48	0	4-128-4
	r / r	$BadClus	2004-06-09 23:22:22 (EDT)	2004-06-09 23:22:22 (EDT)	2004-06-09 23:22:22 (EDT)	2004-06-09 23:22:22 (EDT)	0	0	0	8-128-2
	r / r	$BadClus:$Bad	2004-06-09 23:22:22 (EDT)	2004-06-09 23:22:22 (EDT)	2004-06-09 23:22:22 (EDT)	2004-06-09 23:22:22 (EDT)	10289152	0	0	8-128-1
	r / r	$Bitmap	2004-06-09 23:22:22 (EDT)	2004-06-09 23:22:22 (EDT)	2004-06-09 23:22:22 (EDT)	2004-06-09 23:22:22 (EDT)	2512	0	0	6-128-1
	r / r	$Boot	2004-06-09	2004-06-09	2004-06-09	2004-06-09	8192	48	0	7-128-1

For integrity purposes, MD5 hashes of all files can be made by clicking on the GENERATE MD5 LIST OF FILES button.Investigators can also make notes about files, times, anomalies, and so on, by clicking on the ADD NOTE button:



Enter a note for C:/del2/file7.hmm (31-128-3):

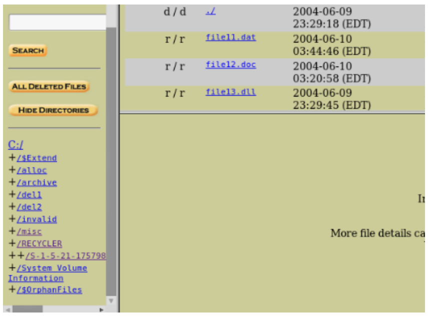
A note works like a bookmark and allows you to later find this data more easily.

☑ Add a Standard Note
Error parsing 'ils' output

The left pane contains four main features that we will be using:

- **Directory Seek**: Allows for the searching of directories
- **File Name Search**: Allows for the searching of files by Perl expressions or filenames
- **ALL DELETED FILES**: Searches the image for deleted files
- **EXPAND DIRECTORIES**: Expands all directories for easier viewing of contents

By clicking on EXPAND DIRECTORIES, all contents are easily viewable and accessible within the left pane and main window. The + next to a directory indicates that it can be further expanded to view subdirectories (++) and their contents:



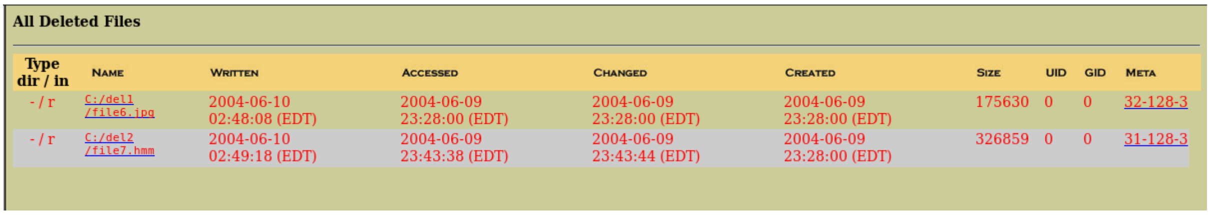
C:/
+/$Extend
+/alloc
+/archive
+/del1
+/del2
+/invalid
+/misc
+/RECYCLER
++/S-1-5-21-175798
+/System Volume Information
+/$OrphanFiles

d / d	./	2004-06-09 23:29:18 (EDT)
r / r	file1.dat	2004-06-10 03:44:46 (EDT)
r / r	file2.doc	2004-06-10 03:20:58 (EDT)
r / r	file3.dll	2004-06-09 23:29:45 (EDT)

In
More file details ca

To view deleted files, we click on the ALL DELETED FILES button in the left pane. Deleted files are marked in red and also adhere to the same format of WRITTEN, ACCESSED, CHANGED, and CREATED times.

From the following screenshot, we can see that the image contains two deleted files:



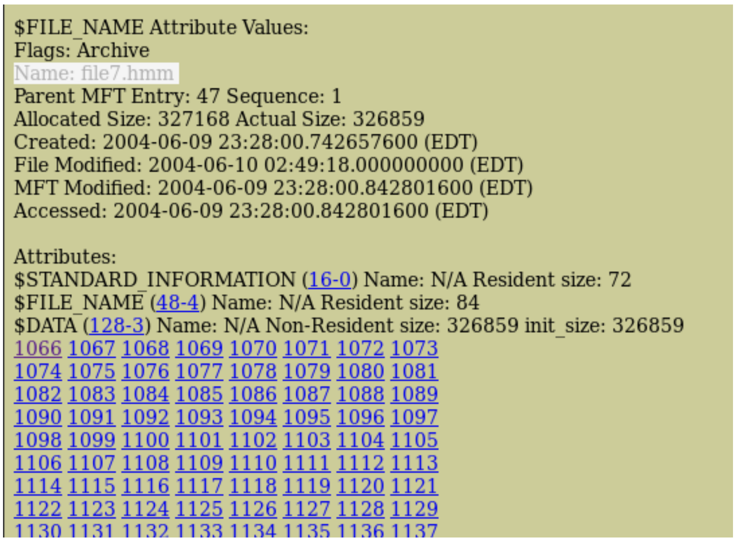
All Deleted Files

Type dir / in	Name	Written	Accessed	Changed	Created	Size	UID	GID	Meta
-/r	C:/del1 /file6.jpg	2004-06-10 02:48:08 (EDT)	2004-06-09 23:28:00 (EDT)	2004-06-09 23:28:00 (EDT)	2004-06-09 23:28:00 (EDT)	175630	0	0	32-128-3
-/r	C:/del2 /file7.hmm	2004-06-10 02:49:18 (EDT)	2004-06-09 23:43:38 (EDT)	2004-06-09 23:43:44 (EDT)	2004-06-09 23:28:00 (EDT)	326859	0	0	31-128-3

We can also view more information about this file by clicking on its META entry. By viewing the metadata entries of a file (last column to the right), we can also view the hexadecimal entries for the file, which may give the true file extensions, even if the extension was changed.

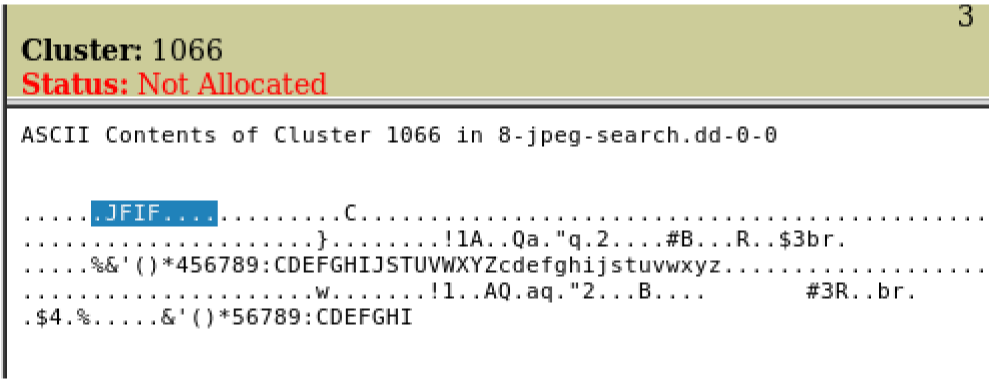
In the preceding screenshot, the second deleted file (file7.hmm) has a peculiar file extension of .hmm.

Click on the META entry (31-128-3) to view the metadata:



```
$FILE_NAME Attribute Values:
Flags: Archive
Name: file7.hmm
Parent MFT Entry: 47 Sequence: 1
Allocated Size: 327168 Actual Size: 326859
Created: 2004-06-09 23:28:00.742657600 (EDT)
File Modified: 2004-06-10 02:49:18.000000000 (EDT)
MFT Modified: 2004-06-09 23:28:00.842801600 (EDT)
Accessed: 2004-06-09 23:28:00.842801600 (EDT)

Attributes:
$STANDARD_INFORMATION (16-0) Name: N/A Resident size: 72
$FILE_NAME (48-4) Name: N/A Resident size: 84
$DATA (128-3) Name: N/A Non-Resident size: 326859 init_size: 326859
1066 1067 1068 1069 1070 1071 1072 1073
1074 1075 1076 1077 1078 1079 1080 1081
1082 1083 1084 1085 1086 1087 1088 1089
1090 1091 1092 1093 1094 1095 1096 1097
1098 1099 1100 1101 1102 1103 1104 1105
1106 1107 1108 1109 1110 1111 1112 1113
1114 1115 1116 1117 1118 1119 1120 1121
1122 1123 1124 1125 1126 1127 1128 1129
1130 1131 1132 1133 1134 1135 1136 1137
```

Under the Attributes section, click on the first cluster labelled 1066 to view header information of the file:



3

Cluster: 1066
Status: Not Allocated

```
ASCII Contents of Cluster 1066 in 8-jpeg-search.dd-0-0


......JFIF.............C............................................
.....................}........!1A..Qa."q.2....#B...R..$3br.
.....%&'()*456789:CDEFGHIJSTUVWXYZcdefghijstuvwxyz..................
.....................w.......!1..AQ.aq."2...B....       #3R..br.
.$4.%.....&'()*56789:CDEFGHI
```

We can see that the first entry is **.JFIF**, which is an abbreviation for **JPEG File Interchange Format**. This means that the `file7.hmm` file is an image file but had its extension changed to `.hmm`.

5. **Conclusion**: In this way we learn how to collect and analyse evidence using Autopsy.

Experiment No. 10

1. **Aim:** Study of Indian IT ACT 2000.
2. **What will you learn by performing this experiment?**

   Laws under the Indian IT ACT

3. **Software Required:** Linux, Java keytool and keystore
4. **Theory:** The Government of India enacted the Information Technology (I.T.) Act with some major objectives to deliver and facilitate lawful electronic, digital, and online transactions, and mitigate cyber-crimes.

   Salient Features of I.T Act

   The salient features of the I.T Act are as follows −

   - Digital signature has been replaced with electronic signature to make it a more technology neutral act.
   - It elaborates on offenses, penalties, and breaches.
   - It outlines the Justice Dispensation Systems for cyber-crimes.
   - It defines in a new section that cyber café is any facility from where the access to the internet is offered by any person in the ordinary course of business to the members of the public.
   - It provides for the constitution of the Cyber Regulations Advisory Committee.
   - It is based on The Indian Penal Code, 1860, The Indian Evidence Act, 1872, The Bankers' Books Evidence Act, 1891, The Reserve Bank of India Act, 1934, etc.
   - It adds a provision to Section 81, which states that the provisions of the Act shall have overriding effect. The provision states that nothing contained in the Act shall restrict any person from exercising any right conferred under the Copyright Act, 1957.

   Scheme of I.T Act

   The following points define the scheme of the I.T. Act −

   - The I.T. Act contains 13 chapters and 90 sections.
   - The last four sections namely sections 91 to 94 in the I.T. Act 2000 deals with the amendments to the Indian Penal Code 1860, The Indian Evidence Act 1872, The Bankers’ Books Evidence Act 1891 and the Reserve Bank of India Act 1934 were deleted.
   - It commences with Preliminary aspect in Chapter 1, which deals with the short, title, extent, commencement and application of the Act in Section 1. Section 2 provides Definition.
   - Chapter 2 deals with the authentication of electronic records, digital signatures, electronic signatures, etc.

- Chapter 11 deals with offences and penalties. A series of offences have been provided along with punishment in this part of The Act.
- Thereafter the provisions about due diligence, role of intermediaries and some miscellaneous provisions are been stated.
- The Act is embedded with two schedules. The First Schedule deals with Documents or Transactions to which the Act shall not apply. The Second Schedule deals with electronic signature or electronic authentication technique and procedure. The Third and Fourth Schedule are omitted.

Application of the I.T Act

As per the sub clause (4) of Section 1, nothing in this Act shall apply to documents or transactions specified in First Schedule. Following are the documents or transactions to which the Act shall not apply −

- Negotiable Instrument (Other than a cheque) as defined in section 13 of the Negotiable Instruments Act, 1881;
- A power-of-attorney as defined in section 1A of the Powers-of-Attorney Act, 1882;
- A trust as defined in section 3 of the Indian Trusts Act, 1882;
- A will as defined in clause (h) of section 2 of the Indian Succession Act, 1925 including any other testamentary disposition;
- Any contract for the sale or conveyance of immovable property or any interest in such property;
- Any such class of documents or transactions as may be notified by the Central Government.

Amendments Brought in the I.T Act

The I.T. Act has brought amendment in four statutes vide section 91-94. These changes have been provided in schedule 1-4.

- The first schedule contains the amendments in the Penal Code. It has widened the scope of the term "document" to bring within its ambit electronic documents.
- The second schedule deals with amendments to the India Evidence Act. It pertains to the inclusion of electronic document in the definition of evidence.
- The third schedule amends the Banker's Books Evidence Act. This amendment brings about change in the definition of "Banker's-book". It includes printouts of data stored in a floppy, disc, tape or any other form of electromagnetic data storage device. Similar change has been brought about in the expression "Certified-copy" to include such printouts within its purview.
- The fourth schedule amends the Reserve Bank of India Act. It pertains to the regulation of fund transfer through electronic means between the banks or between the banks and other financial institution.

5. Conclusion: In this way we have learnt about Indian IT ACT 2000.