Computer & Network Forensics

Lab 1

Software Write-Blocker

Questions

1. Why are write blockers essential in a forensics investigation?

   Write blockers are very important in digital forensics because this basically protects the original evidence. When investigators plug in a suspect's hard drive or USB, we need to look at everything on it without accidentally changing a single thing. Even something small, like the system updating a file timestamp, could mess up the case and make evidence unusable in court. Write blockers stop the computer from writing anything back to the device, so investigators can safely read the data without worrying about altering it.

2. What are the main types of write blockers?

   There are two main types: hardware and software. Hardware write blockers are physical devices you plug the suspect's drive into before connecting it to the forensic machine; they block all writes at the hardware level. Software write blockers are programs that run on the computer to stop write commands.

3. What are the main challenges of write blocking for forensics investigators?

   One challenge is compatibility – some drives or file systems don't play nice with certain write blockers. Another issue is reliability: investigators need to be absolutely sure the blocker is actually preventing writes, because if something slips through, the evidence could be compromised. There's also the cost factor, since good hardware blockers can be expensive. Plus, investigators need to stay updated with technology changes – new storage devices come out all the time, and blockers need to keep up.

4. Discuss the implications of using open-source technologies for write blocking.

   Open-source tools are appealing because they're free and customizable, but they also come with risks. Since the code is public, it can be inspected for flaws, which is good for transparency. But at the same time, it might not always meet strict forensic standards, and defense lawyers could challenge its reliability in court. Also, open-source tools often don't get the same level of official certification as commercial hardware blockers. So, while open-source can be great for learning or smaller cases, most professionals stick with certified tools when they know evidence might go to trial.

Computer & Network Forensics

Lab 1a

Extracting Volatile Data (Manually)

Commands

1. Command to Capture System Information

   ```
   systeminfo >> VolatileDataFile.txt
   ```

   → Capture the system information and save it into a text file for later analysis

2. Checking Active Network Connections

   ```
   netstat -nao >> VolatileDataFile.txt
   ```

   → Using the netstat (network statistics) command, we can capture details about all active network connections

3. Routing Configuration

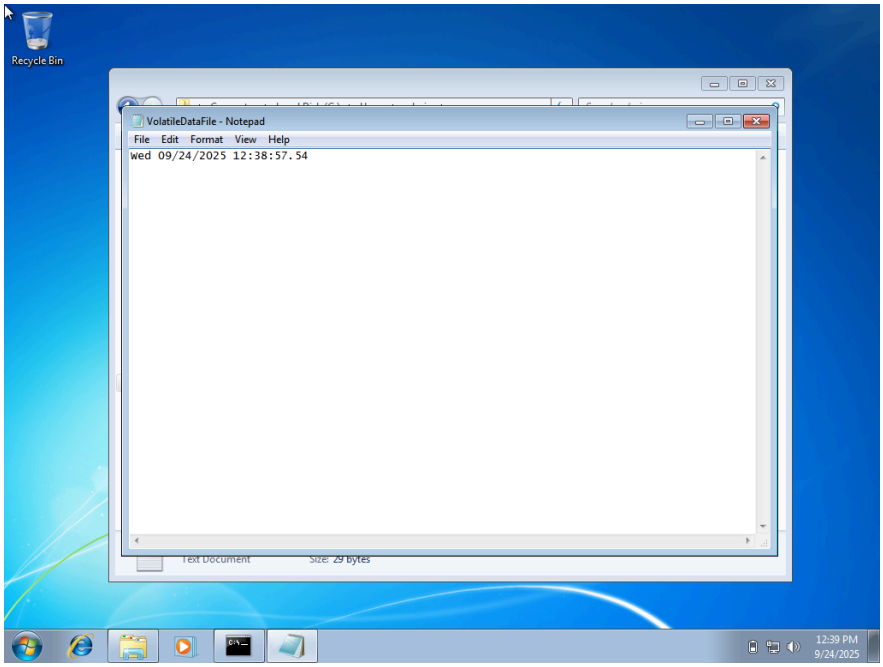
   ```
   route print >> VolatileDataFile.txt
   ```

   → Routing configuration refers to the setup of IP addresses, gateway settings, and network routes that direct data traffic between devices in a network. This information is crucial in digital forensics because improper or malicious routing could indicate unauthorized access or a compromised system.

1. Date and Time

```
echo %date% %time% > VolatileDataFile.txt
```

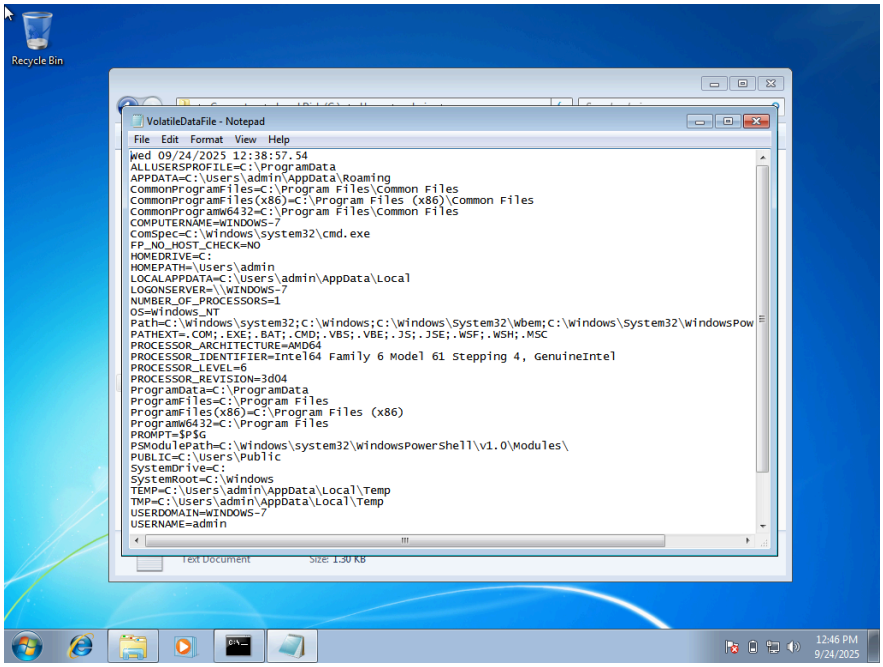
→ Records the current system date and time. Useful for timestamping forensic collection.



2. System Variables

```
set >> VolatileDataFile.txt
```

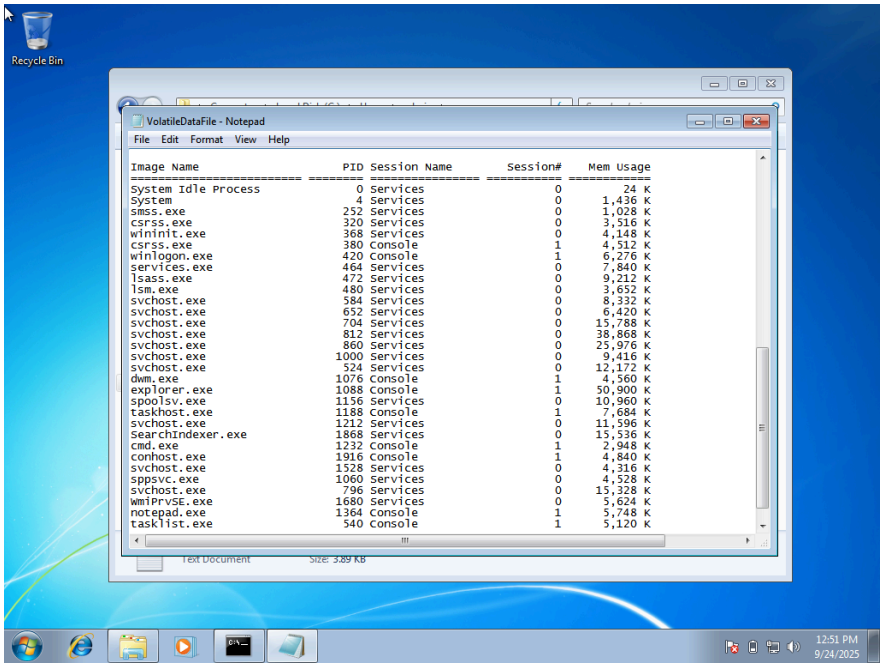
→ Lists all environment variables. Helps investigators see system paths, user variables, and possible malware persistence points.



3. Task List

```
tasklist >> VolatileDataFile.txt
```

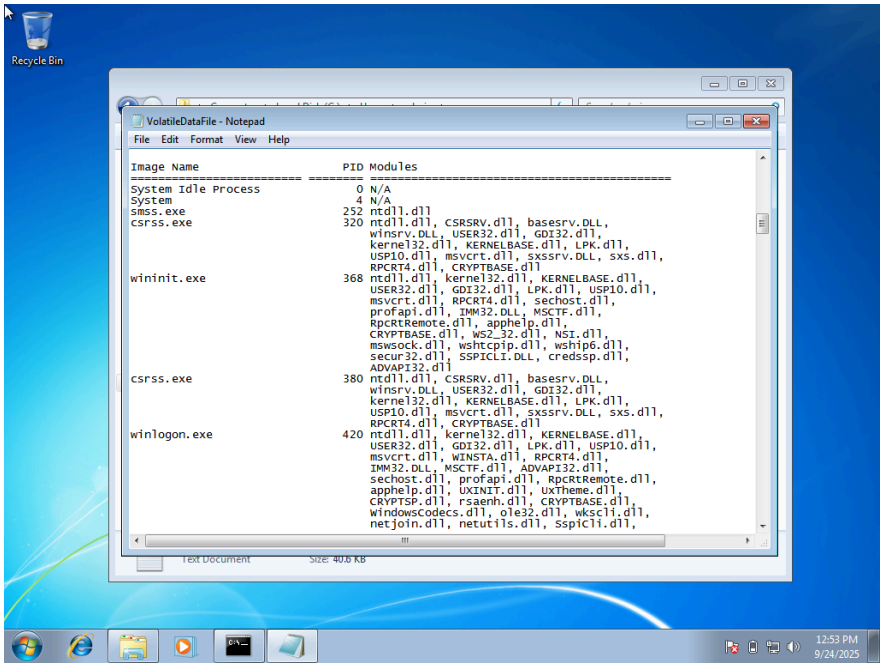
→ Shows all running processes with PID and memory usage. Detects suspicious programs.



4. Task List with Modules

```
tasklist /m >> VolatileDataFile.txt
```

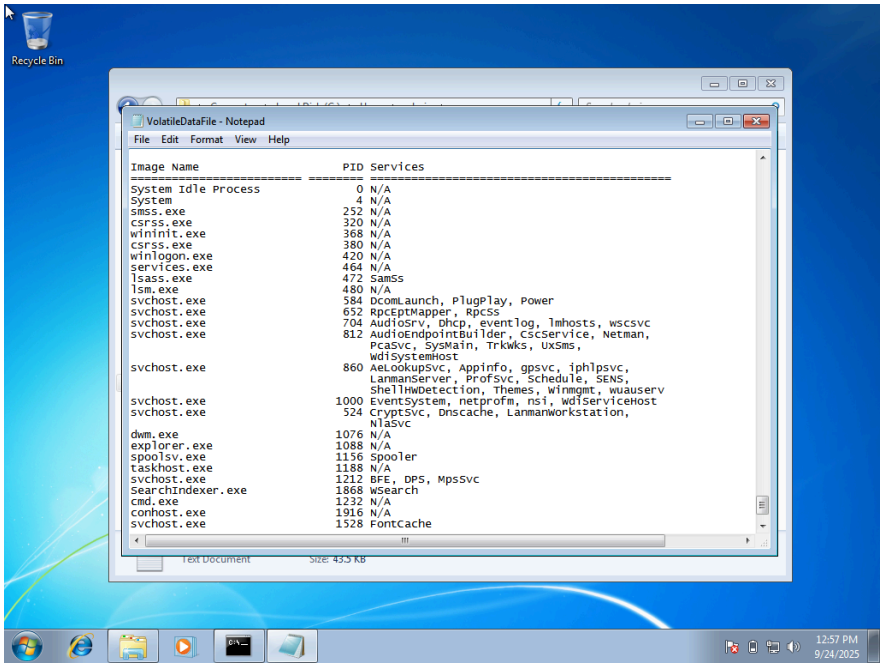
→ Shows processes with loaded DLL modules. Can reveal injected malicious DLLs.



5. Task List with Services

```
tasklist /svc >> VolatileDataFile.txt
```

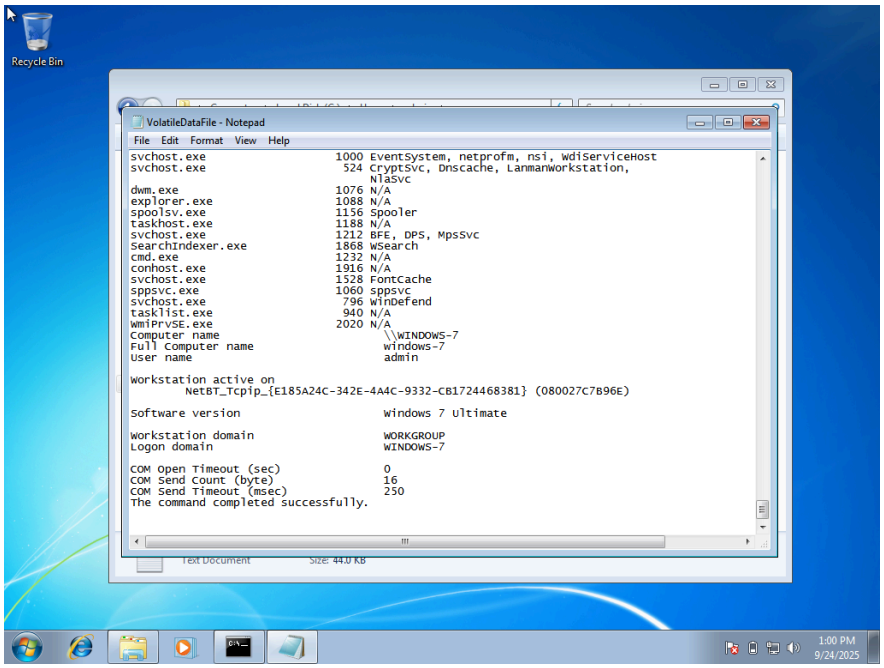
→ Links processes to services they host. Useful for spotting rogue services.



6. Workstation Information

```
net config workstation >> VolatileDataFile.txt
```

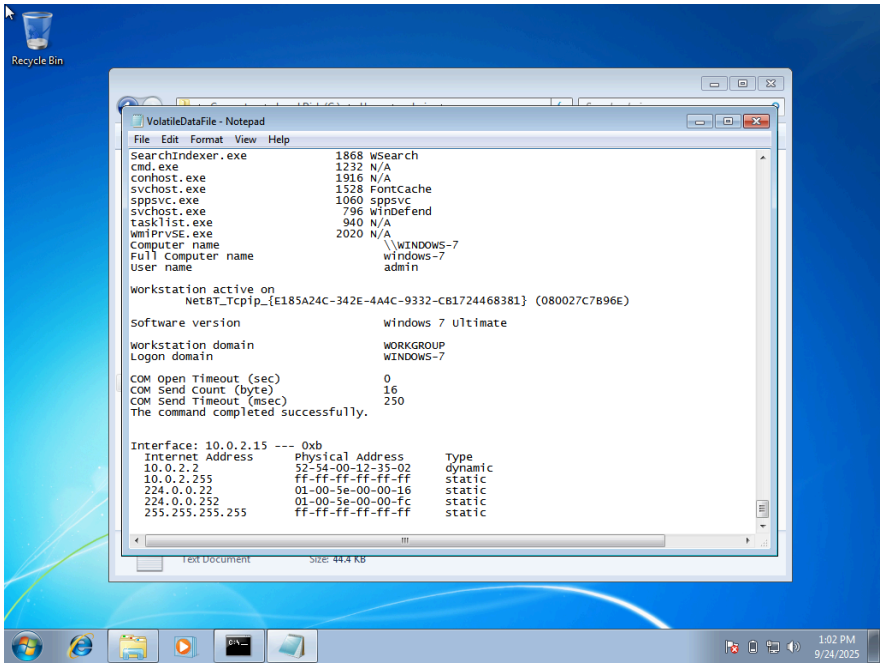
→ Displays workstation settings like computer name, domain, and logon details.



7. MAC Address saved in System ARP Cache

```
arp -a >> VolatileDataFile.txt
```

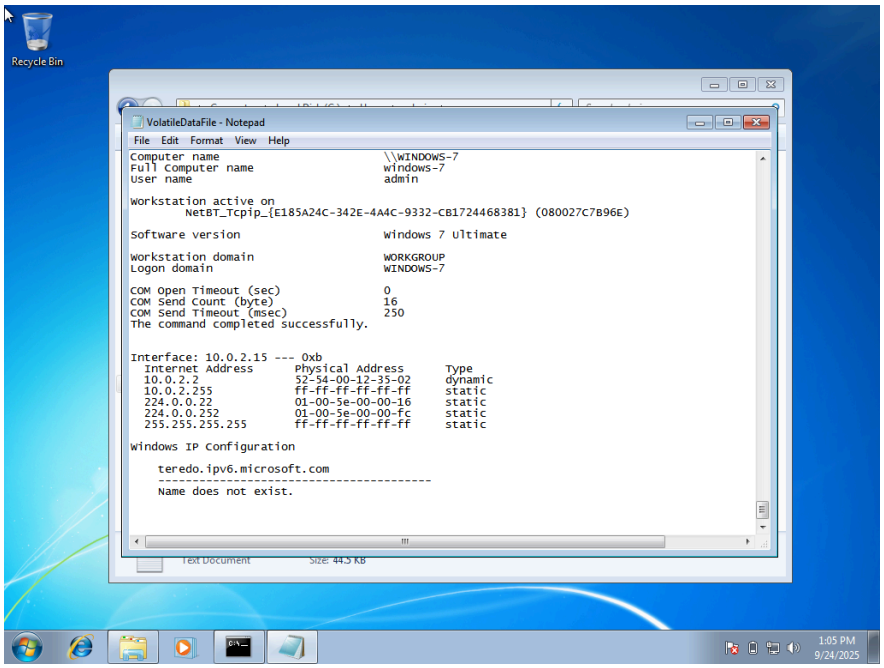
→ Shows IP-to-MAC address mappings. Helps trace active devices on the local network.



8. DNS Configuration

```
ipconfig /displaydns >> VolatileDataFile.txt
```

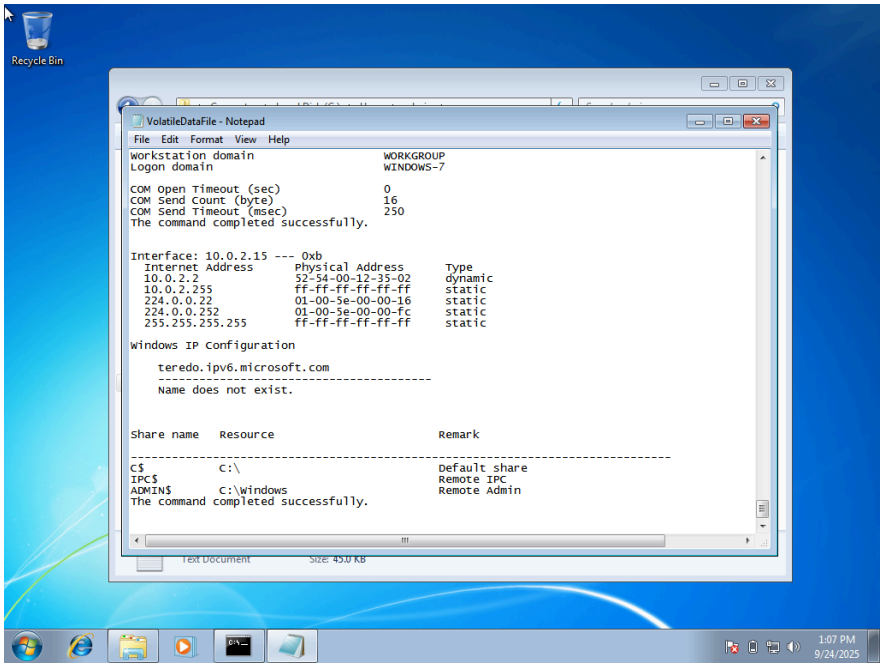
→ Lists cached DNS records. Useful for spotting suspicious domains visited.



9. System network shares

```
net share >> VolatileDataFile.txt
```

→ Displays shared folders. Attackers might create hidden shares for data theft.



10. Network Configuration

```
ipconfig /all >> VolatileDataFile.txt
```

→ Shows detailed network adapter settings, IPs, gateways, and DNS servers. Can reveal anomalies like rogue DNS or static routes.