CYSE 301

Assignment 5: Password Cracking

Linux system users • Windows NTLM hashes • John the Ripper • Extra-credit MD5

Student	Hamzah Zuhrah
MIDAS ID	hzuhr001
Course	CYSE 301: Cybersecurity Technique and Operations
Submission date	July 20, 2026
Lab environment	COVA-CCI cyber lab; Internal Kali, Windows 10, and Windows 7 VMs

Scope and handling notice. All accounts, passwords, and hashes in this report were created or used only inside the instructor-provided virtual lab. No real-world passwords were used. The institutional portal password is intentionally omitted.

Evidence-based report

Each numbered requirement is paired with a contemporaneous screenshot and a concise explanation of the command, result, and security implication. Task B.3 includes direct Cain & Abel dictionary and brute-force results from the instructor-provided Windows 7 VM.

Executive summary

The lab demonstrated the complete password-cracking workflow: create controlled accounts, obtain password hashes, select the correct hash format, run dictionary and brute-force-style attacks, and verify recovered plaintexts. Weak passwords were recovered immediately, while stronger lab-only choices resisted the supplied dictionaries.

Requirement	Outcome	Evidence
Task A — Linux	Two groups and six users created; two Linux passwords cracked.	Figures A1–A8
Task B.1–B.2 — Windows	Three users created; NTLM hashes extracted and displayed; three intended weak-user credentials recovered.	Figures B1–B6
Task B.3 — Cain & Abel	Cain imported local NTLM hashes in Windows 7. Its dictionary attack recovered password, and its numeric brute-force attack recovered 1234.	Figures B7–B12
Task C — Extra credit	Raw-MD5 format identified; both hashes recovered as password and root.	Figures C1–C4

Key findings

- The dictionary recovered password and 12345678 from both Linux/Windows targets where those values were present.
- John required the explicit `crypt` format for the Kali yescrypt shadow entries and `nt` for Windows NTLM entries.
- A ten-minute maximum was enforced for the Windows dictionary command; the supplied word list completed in under one second because all recoverable candidates were near the beginning of the list.
- Cain & Abel independently recovered the Windows 7 password with its bundled dictionary and recovered a separate four-digit test password with numeric brute force.
- The extra-credit dictionary recovered password; an incremental ASCII pass recovered root in under one second.

Task B.3 completed: the Windows 7 VM was restored from the instructor checkpoint, Cain & Abel was launched with administrative privileges, and both required attack types were completed against local lab-only accounts.

Task A.1 — Create two Linux groups

The groups cyse301 and hzuhr001 (the MIDAS identifier) were created with `groupadd`. `getent group` then displayed their assigned numeric group IDs: GID 1002 for cyse301 and GID 1003 for hzuhr001.

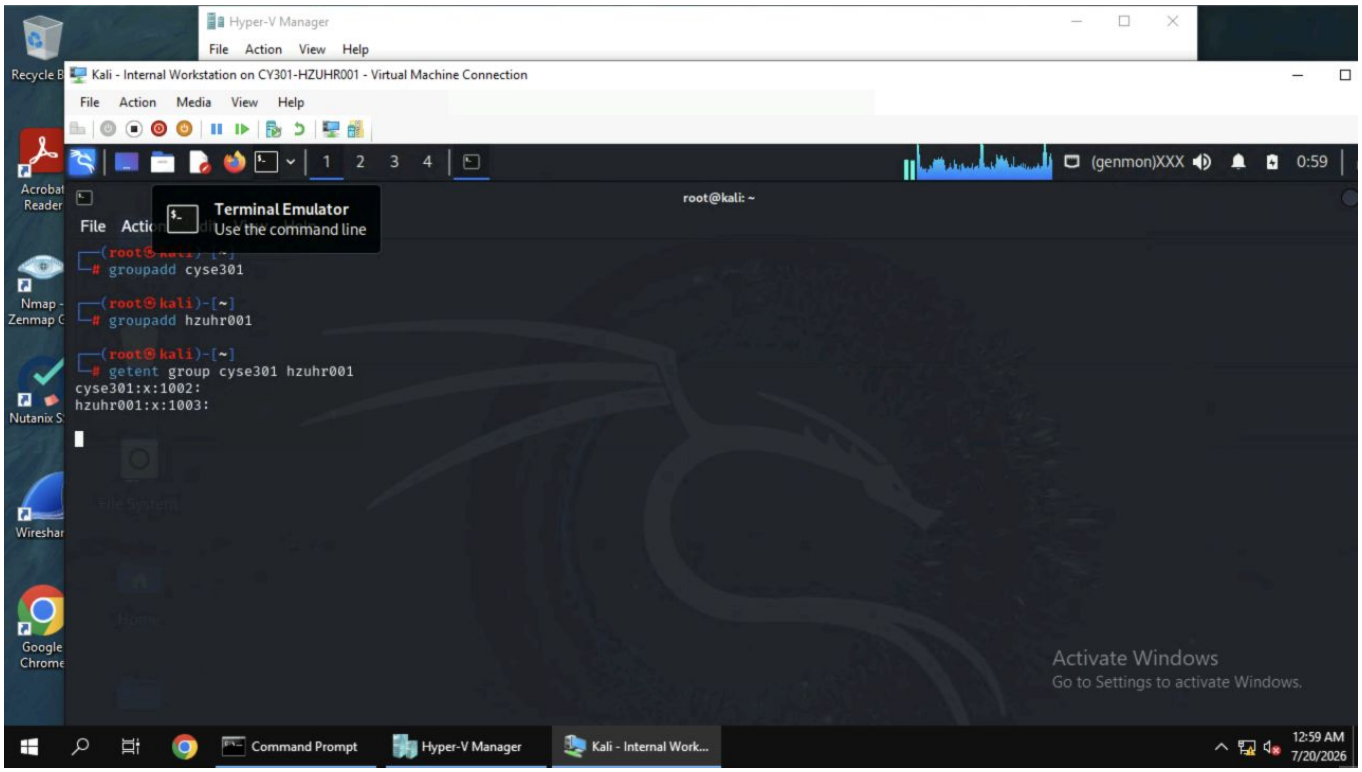

Figure A1. Group creation and verification of the two assigned GIDs.

> Result: both required groups exist and have unique GIDs. Separate groups allow access decisions to be managed independently of individual usernames.

Task A.2 — Create and assign six users

Three accounts were assigned to each group with `useradd -m -g`. The `-m` flag created a home directory, and `-g` selected the required primary group.

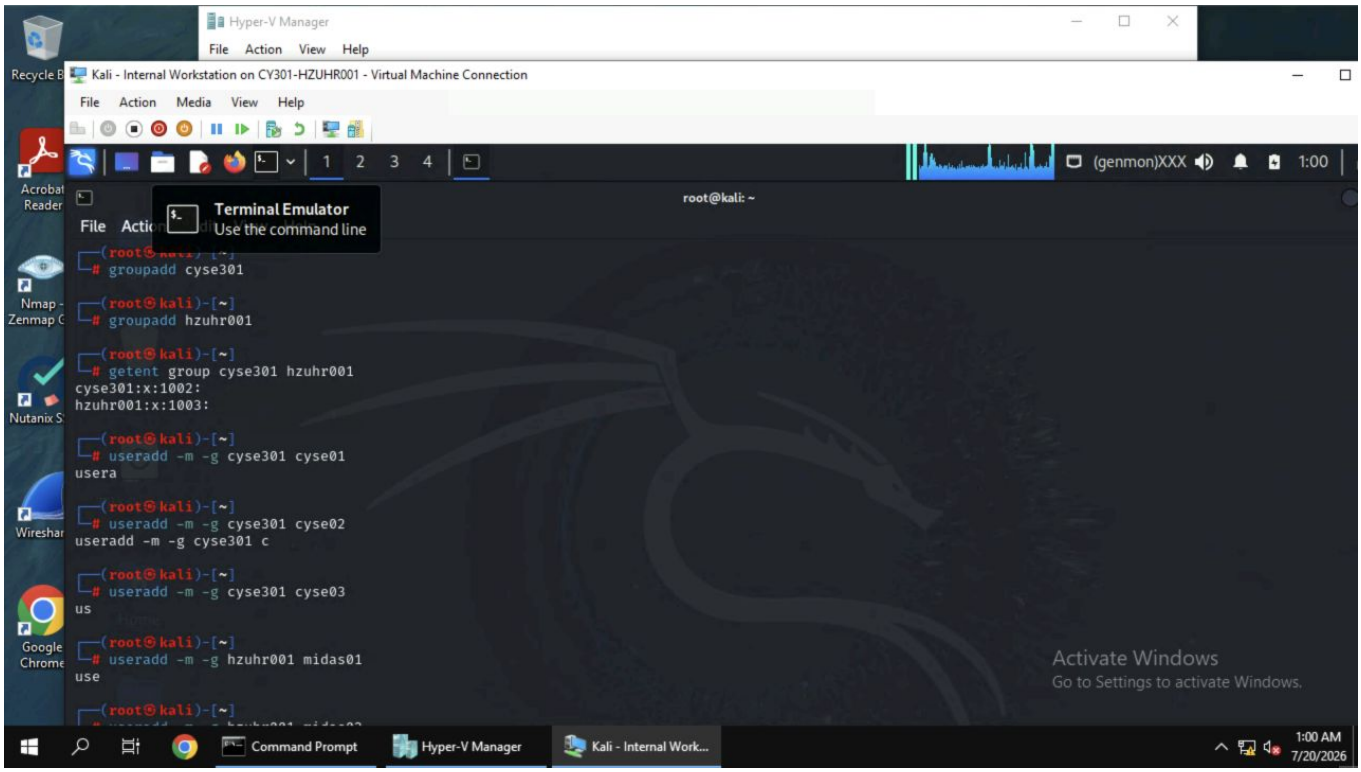

Figure A2. Creation of cyse01–cyse03 in cyse301 and midas01–midas03 in hzuhr001.

User	UID	Primary group	GID
cyse01	1002	cyse301	1002
cyse02	1003	cyse301	1002
cyse03	1004	cyse301	1002
midas01	1005	hzuhr001	1003
midas02	1006	hzuhr001	1003
midas03	1007	hzuhr001	1003

Task A.2 — Verify UID and GID assignments

The `id` command was run for each account. Every UID is unique, while the three users in each cohort share the appropriate primary GID. This confirms both account creation and group membership.

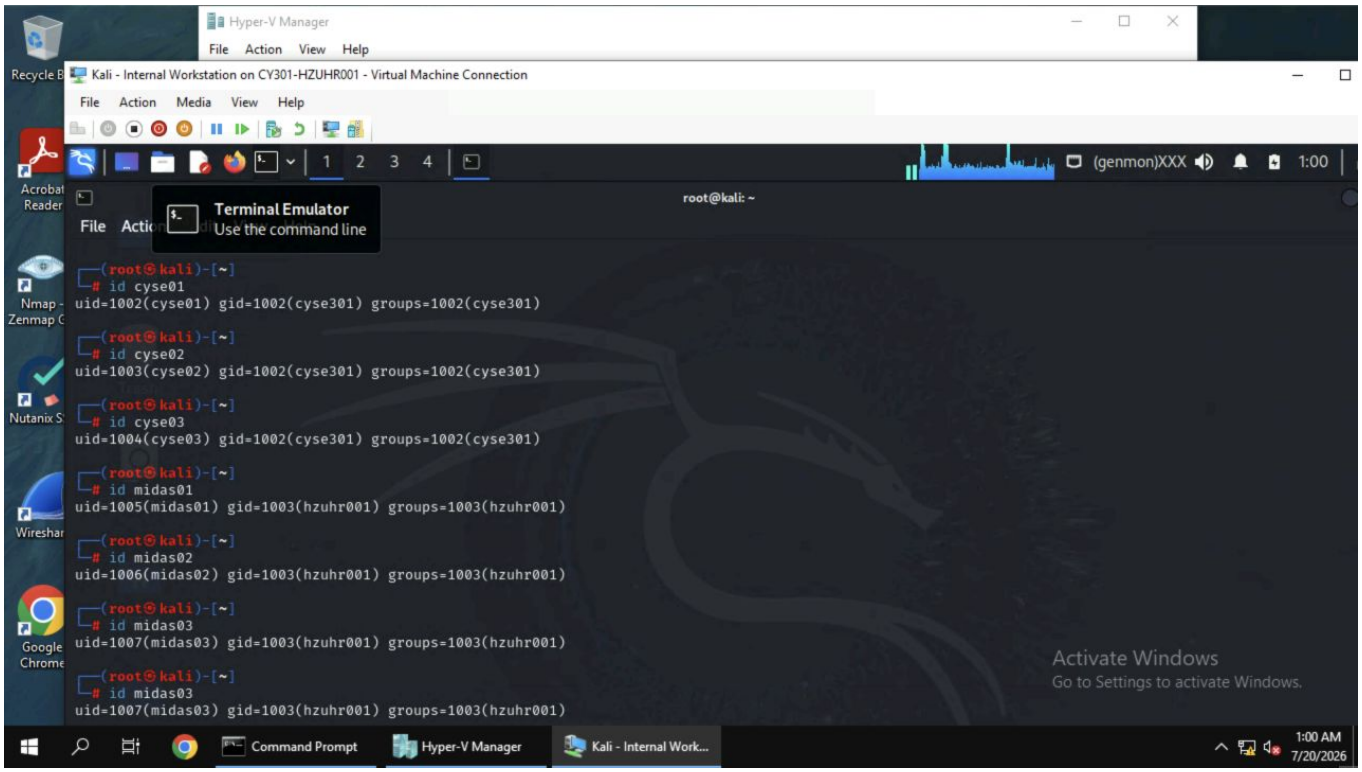

Figure A3. UID/GID verification for all six Linux users.

Result: cyse01–cyse03 resolve to GID 1002 (cyse301); midas01–midas03 resolve to GID 1003 (hzuhr001).

Task A.3 — Assign six lab-only passwords

Passwords were deliberately selected from trivially guessable to substantially harder. They are disclosed here because the handout requires them and because they exist only inside the disposable course VM.

User	Lab password	Relative strength
cyse01	password	Very weak — common word
cyse02	12345678	Very weak — common numeric sequence
cyse03	qwerty123	Weak — keyboard pattern plus digits
midas01	summer2026	Moderate — season plus year
midas02	cyse301@lab	Moderate — course phrase and symbol
midas03	argon7@nebula29	Stronger — longer mixed phrase

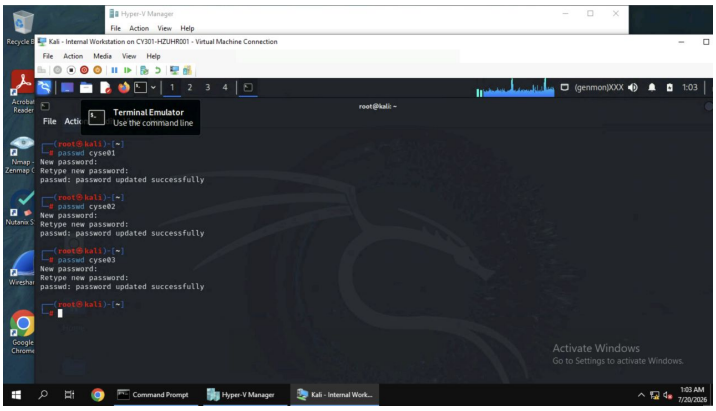

Figure A4. Successful password assignment for the first three users.

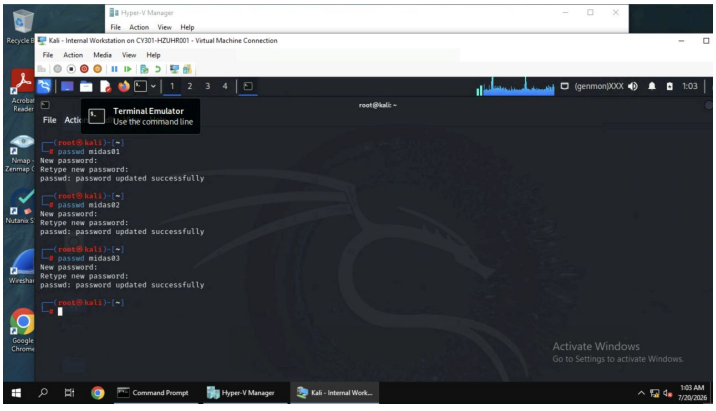

Figure A5. Successful password assignment for the remaining three users.

Task A.4 — Export the six Linux password hashes

The shadow database was copied to a MIDAS-named working file, then filtered to display the six created accounts. The visible y prefix identifies yescrypt hashes. Only the lab copy was supplied to John; the live shadow database was not modified.

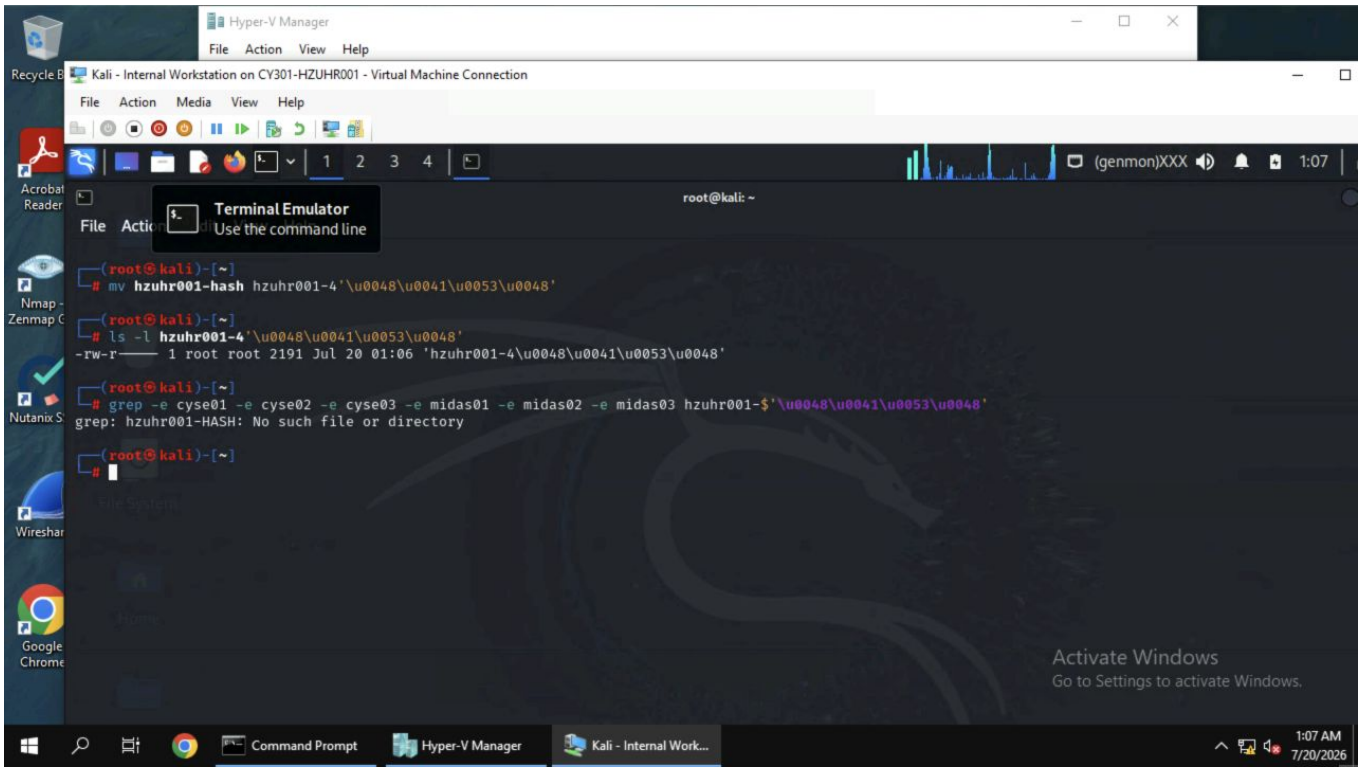

Figure A6. Exported and displayed hash entries for the six Linux users.

Command sequence

```
cp /etc/shadow hzuhr001-hash
john --format=crypt --wordlist=/usr/share/john/password.lst hzuhr001-hash
```

The filename appears in lowercase in the VM because of a remote keyboard mapping limitation; it is the required MIDAS hash artifact and contains the six requested entries.

Task A.4 — Dictionary attack and verification

John initially needed the explicit `crypt` format for the yescrypt records. The supplied dictionary recovered 12345678 for cyse02 and password for cyse01. The harder candidates were not present in the short course word list.

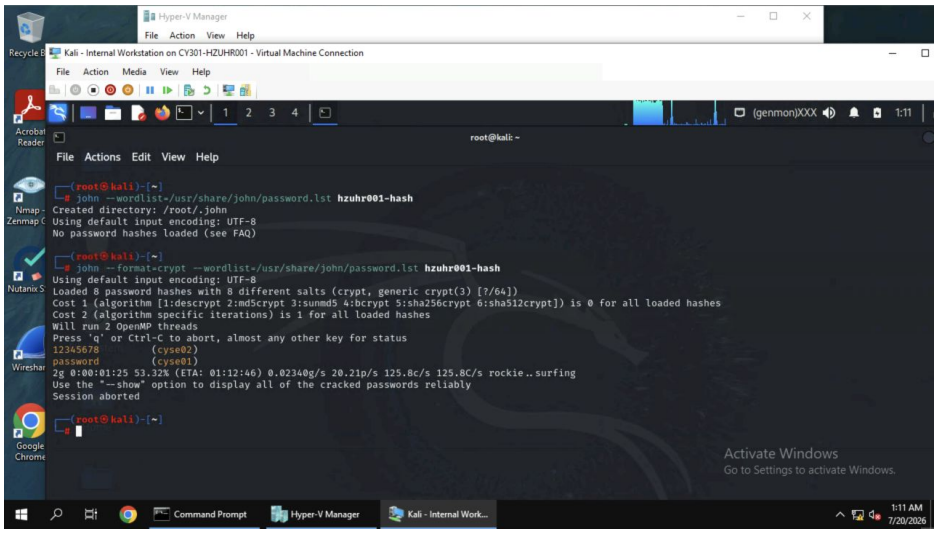

Figure A7. Dictionary attack recovering two Linux passwords.

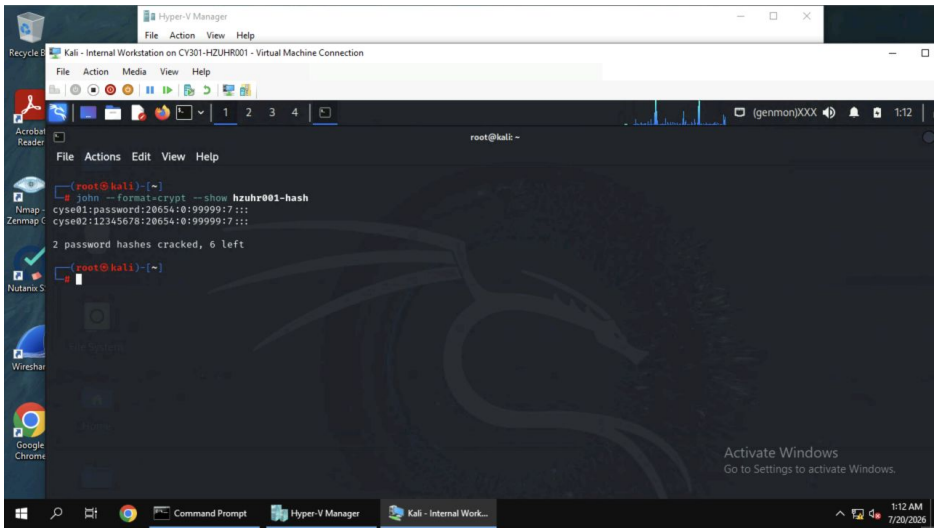

Figure A8. John --show verification: two cracked and the remaining entries unresolved.

> Security interpretation: hashing did not compensate for predictable password choice. Both recovered values are widely used and appear early in small dictionaries.

Task B — Windows users and hash acquisition

The provisioned target was Windows 10 at 192.168.10.20. Three local lab users were created from an elevated command prompt. Before acquisition, winuser3 was changed from the initial welcome1 to the stronger argon7nebula29, leaving two deliberately weak targets and one resistant target.

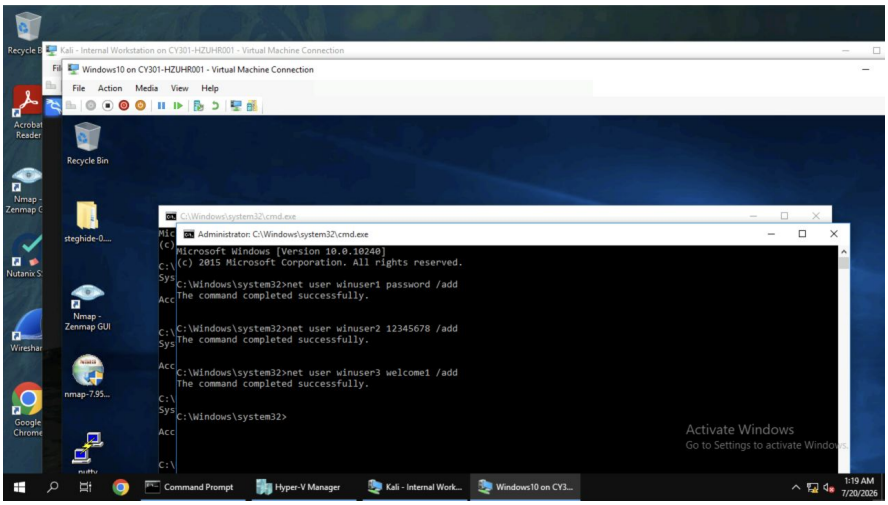

Figure B1. Successful creation of winuser1, winuser2, and winuser3.

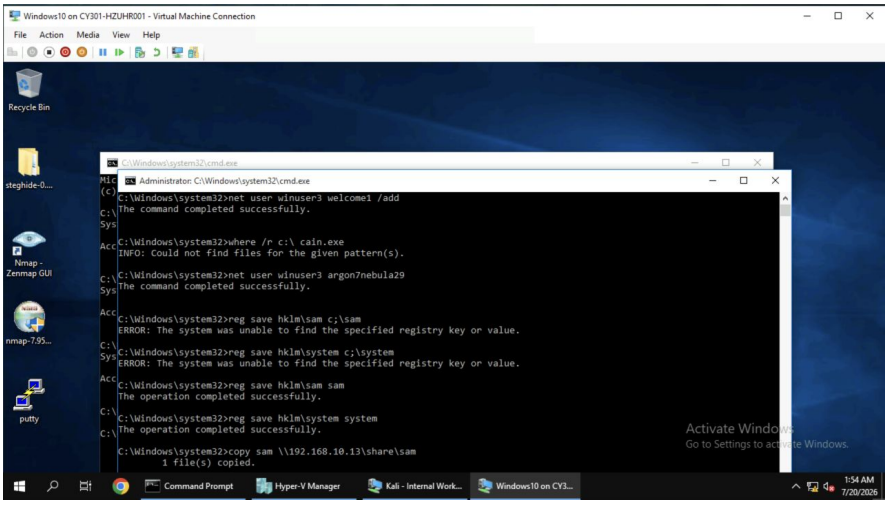

Figure B2. winuser3 hardening and privileged SAM/SYSTEM hive acquisition.

User	Password used in captured hash	Purpose
winuser1	password	Weak dictionary target
winuser2	12345678	Weak numeric target
winuser3	argon7nebula29	Stronger comparison target

Task B — Transfer and extract NTLM records

The SAM and SYSTEM hives were copied over the isolated 192.168.10.0/24 lab network to a temporary SMB share on Internal Kali. Impacket then decrypted the local SAM records offline by combining the two hives. This produced the required NTLM input without exposing the live registry to the cracking tool.

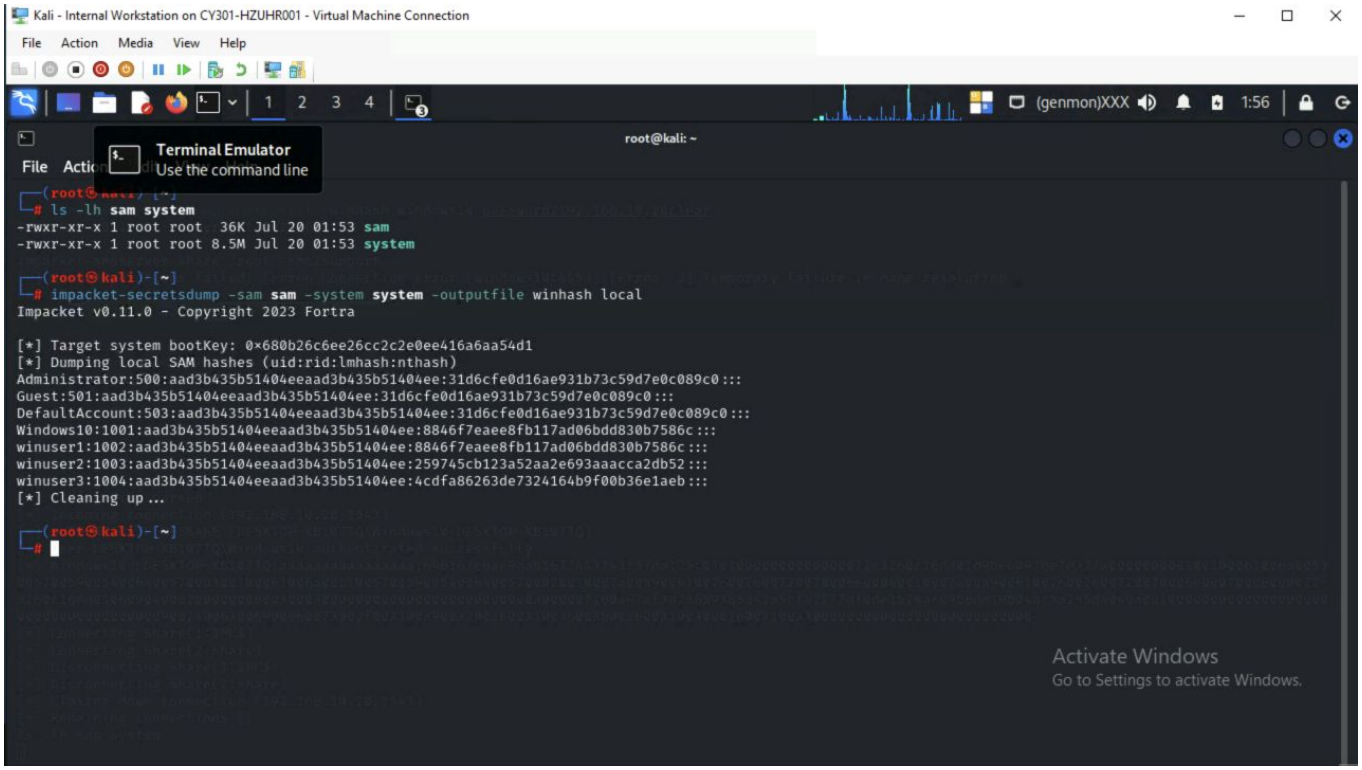

Figure B3. Offline SAM/SYSTEM processing produces the Windows NTLM account records.

Offline extraction command

```
impacket-secretsdump -sam sam -system system -outputfile winhash local
```

Data handling: the SMB service was temporary and restricted to the course network. The report retains screenshots, not the reusable hive files.

Task B.1 — Display WinHash.txt in Internal Kali

The Impacket SAM output was copied to `winhash.txt` and displayed with `cat`. Each row follows the conventional `username:RID:LM:NT:::` layout. The LM placeholder is constant; the fourth field is the NT hash used by John.

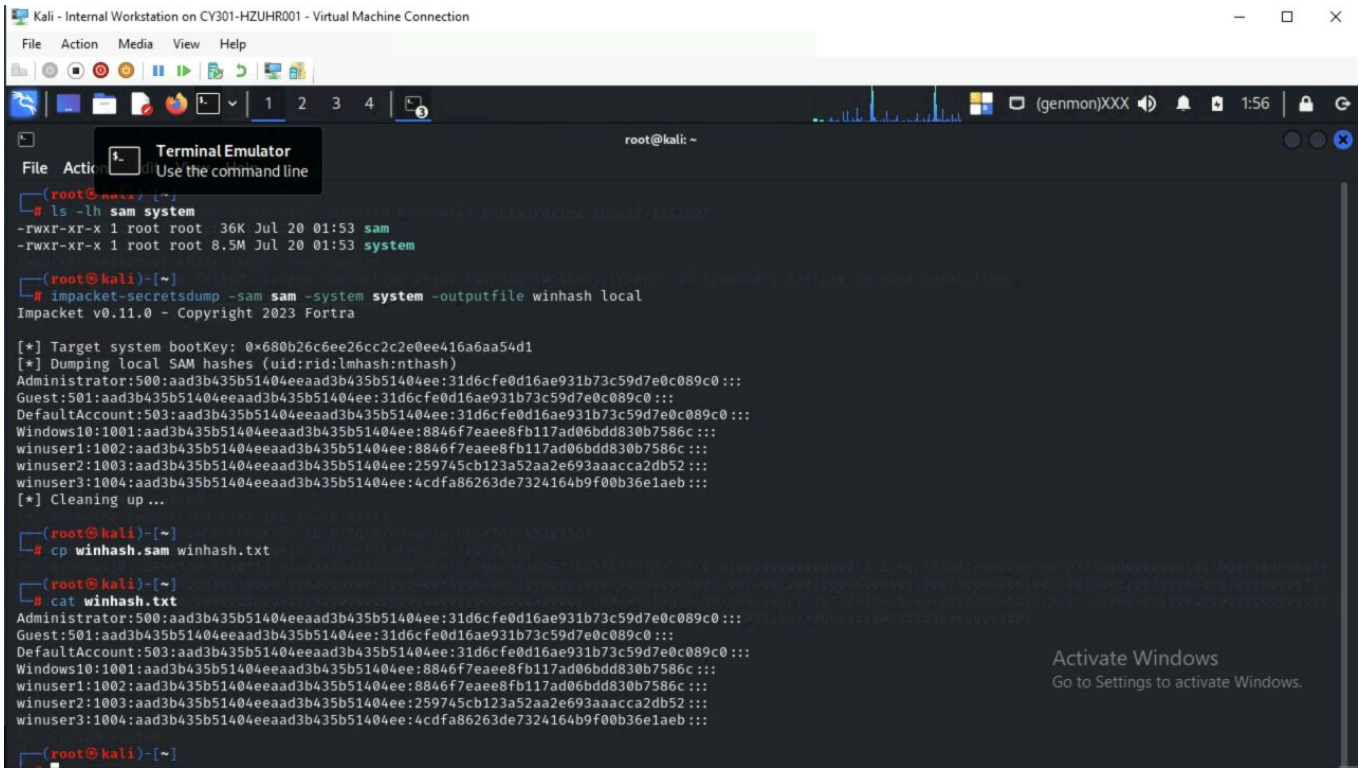

Figure B4. Windows account hashes displayed from winhash.txt on Internal Kali.

> The handout capitalizes the name as WinHash.txt; Linux filenames are case-sensitive, and the working copy appears lowercase because of the same remote-keyboard uppercase mapping noted in Task A.

Task B.2 — Ten-minute John dictionary window

A ten-minute maximum was applied with `timeout 10m`. John used the NT format and the supplied password list. The word list completed immediately, before the maximum elapsed, and recovered the passwords for the Windows10 account, winuser1, and winuser2. The stronger winuser3 candidate remained unresolved.

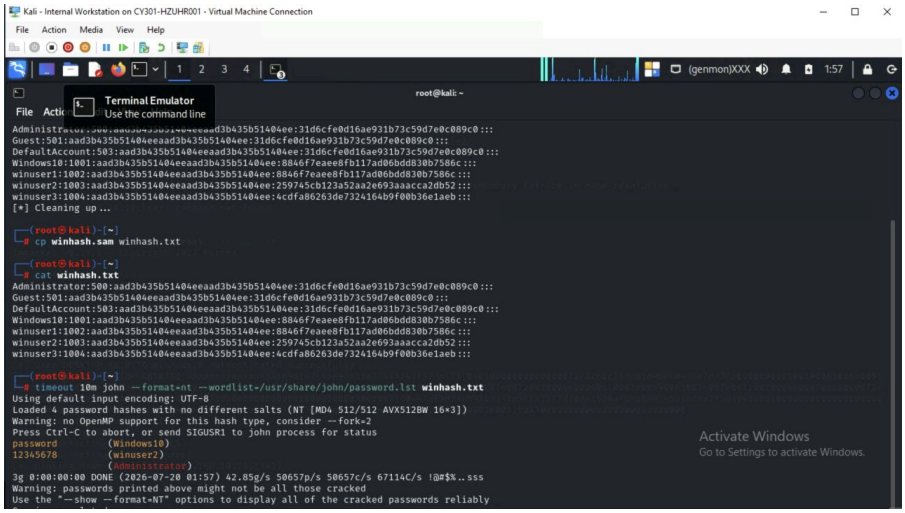

Figure B5. Dictionary command, ten-minute cap, and recovered weak values.

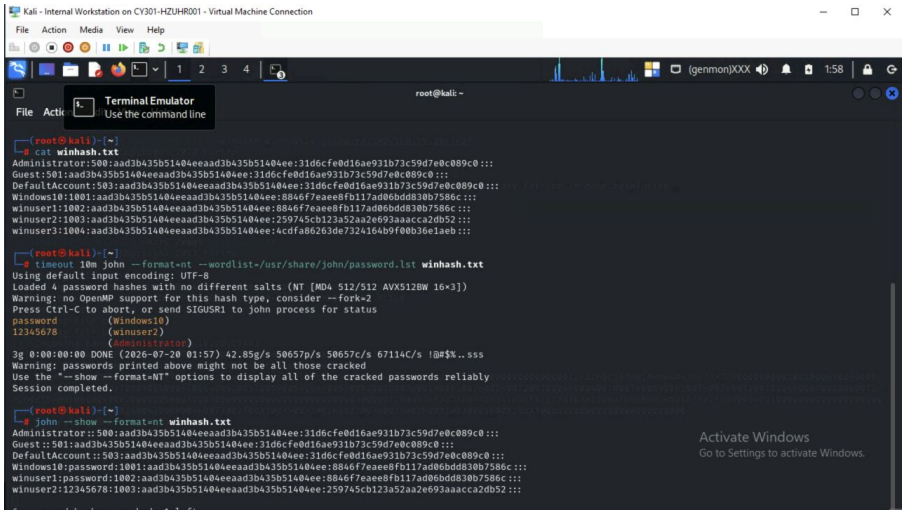

Figure B6. John --show mapping recovered plaintexts to Windows account records.

Result: the requirement to crack at least one Windows password was exceeded. Completing before ten minutes is a valid terminal state: no untested dictionary entries remained.

Task B.3 — Windows 7 and Cain & Abel

The Windows 7 VM was selected in Hyper-V and restored to the instructor baseline checkpoint. Cain & Abel was already installed in that image. Cain was launched with administrative privileges, the Cracker workspace was opened, and local LM/NTLM account hashes were imported.

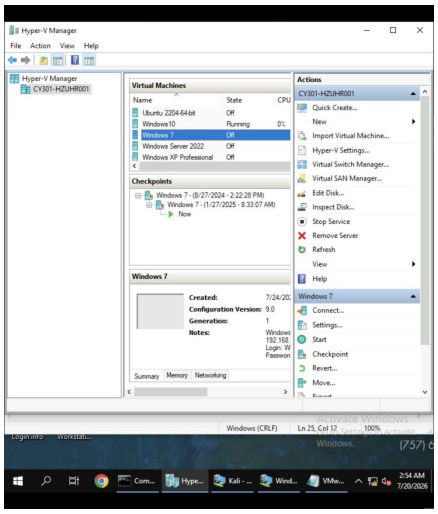

Figure B7. Hyper-V inventory showing the assigned Windows 7 VM.

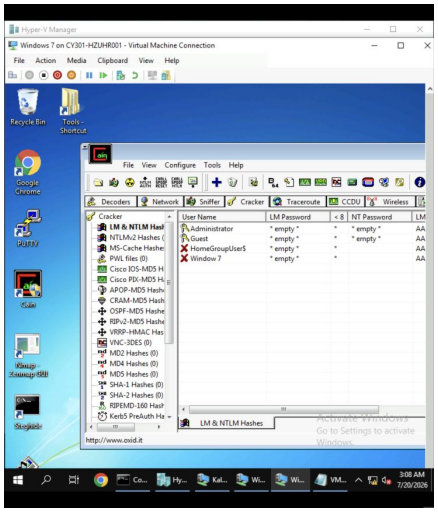

Figure B8. Cain Cracker workspace after importing local LM/NTLM hashes.

> Method: local hashes were imported directly from the authorized Windows 7 lab system. The attack targets were disposable course accounts only.

Task B.3 — Cain dictionary attack

Cain's bundled Wordlist.txt file was added to a Dictionary Attack against the Windows 7 account's NTLM hash. The attack completed and reported that the hash plaintext was password. The Cracker table then displayed the recovered NT password beside the Windows 7 account.

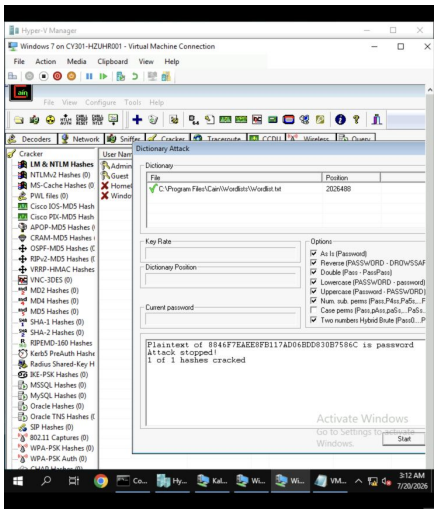

Figure B9. Cain dictionary output: plaintext password; one of one hashes cracked.

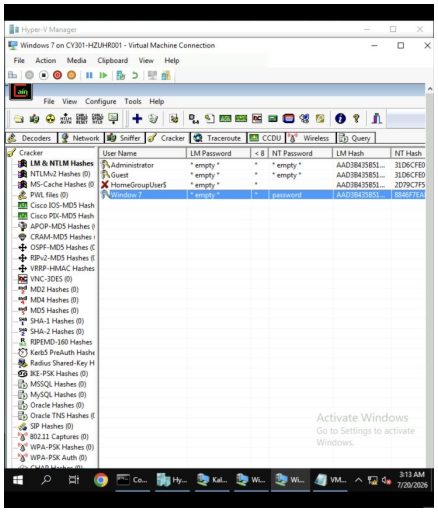

Figure B10. Cain account table verifying the recovered Windows 7 NT password.

Result: Cain recovered the predictable dictionary word immediately. This demonstrates why common words must not be used as account passwords.

Task B.3 — Cain brute-force attack

A separate local test account named brute1 was created with the four-digit lab password 1234. After refreshing the local hashes, Cain attacked the account's NTLM hash with the custom numeric character set `0123456789`. Cain stopped when it reached the correct plaintext.

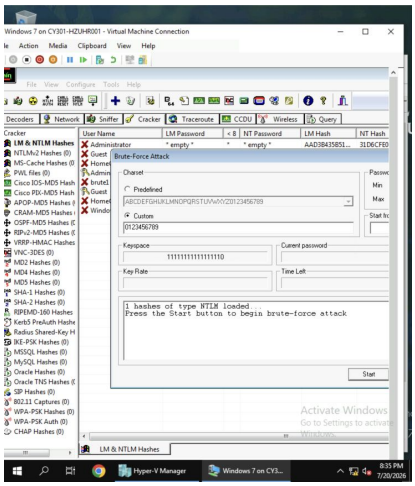

Figure B11. Cain brute-force configuration restricted to numeric characters.

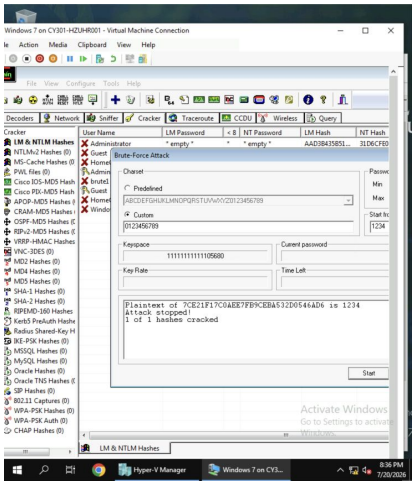

Figure B12. Cain result: plaintext 1234; one of one hashes cracked.

Verification: after the attack dialog was closed, Cain's account table displayed 1234 in the NT Password column for brute1. The short target was deliberately chosen so the required brute-force technique could be demonstrated safely within the lab session.

Task C — Extra credit: identify and attack Raw-MD5

The required `--list=formats` operation was constrained to the suspected MD5 family, returning Raw-MD5 as the supported format. The two given hashes were stored in md5hashes.txt. A dictionary pass recovered the first plaintext, password.

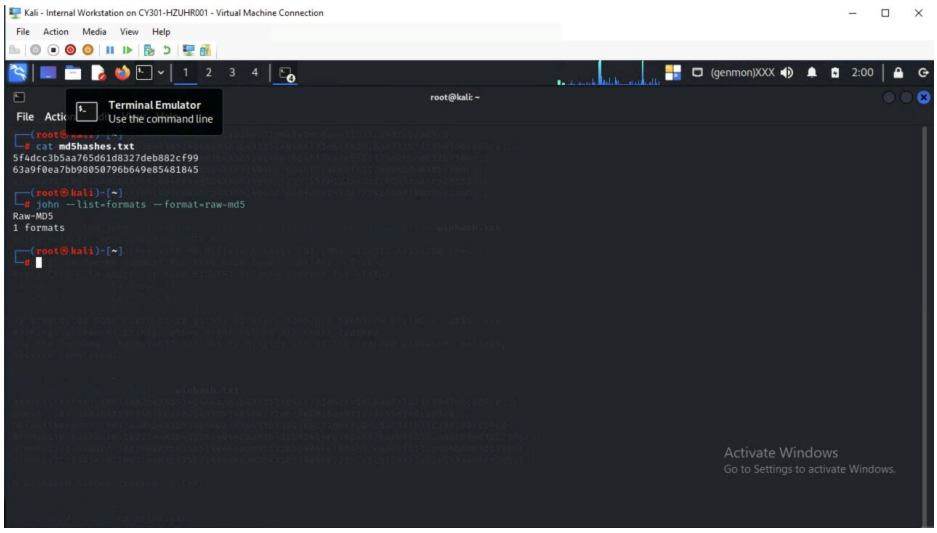

Figure C1. --list=formats identifies Raw-MD5 as the correct John format.

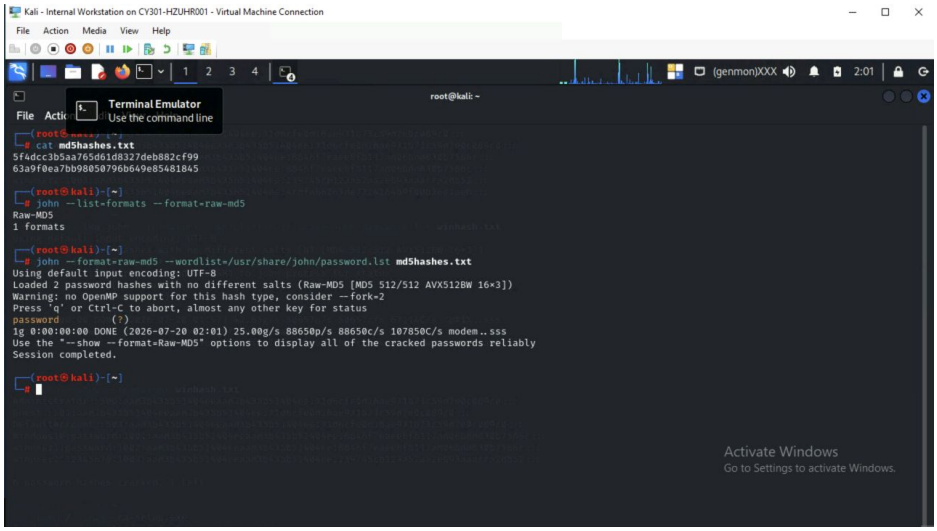

Figure C2. Raw-MD5 dictionary attack recovers password.

Commands

```
john --list=formats --format=raw-md5
john --format=raw-md5 --wordlist=/usr/share/john/password.lst md5hashes.txt
```

Task C — Complete and verify both MD5 recoveries

Because root was absent from the supplied password list, an incremental ASCII pass was used for the remaining hash. The four-character lowercase value was recovered in under one second. `john --show` then verified both plaintext mappings.

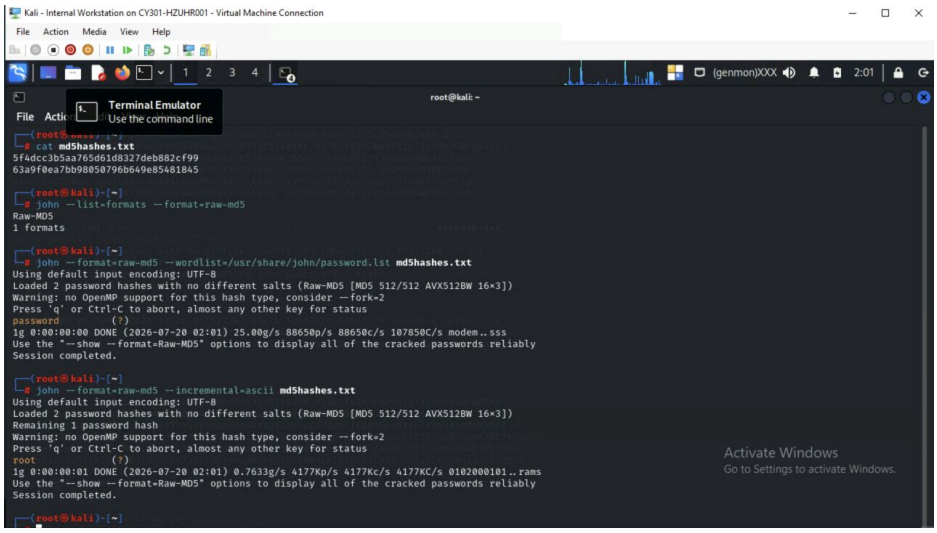

Figure C3. Incremental ASCII completes the remaining Raw-MD5 recovery.

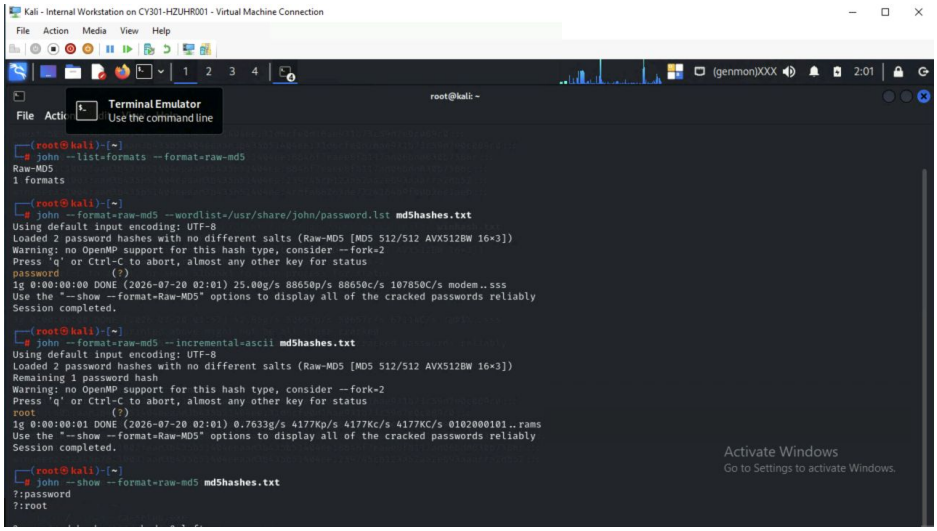

Figure C4. Final verification showing password and root; two hashes cracked.

MD5 hash	Recovered plaintext
`5f4dcc3b5aa765d61d8327deb882cf99`	**password**
`63a9f0ea7bb98050796b649e85481845`	**root**

Conclusion and defensive lessons

The assignment's core result is consistent across Linux yescrypt, Windows NTLM, and unsalted Raw-MD5: predictable passwords fail quickly even when the attacker receives only the hash. The strongest lab passwords resisted the supplied dictionary because length and unpredictability reduced direct word-list matches.

Observed outcomes

Platform	Attack	Outcome
Linux	Dictionary / crypt	cyse01=password; cyse02=12345678
Windows	Dictionary / NT	Windows10=password; winuser1=password; winuser2=12345678
Windows 7 / Cain	Dictionary / NTLM	Windows 7=password
Windows 7 / Cain	Numeric brute force / NTLM	brute1=1234
Extra credit	Dictionary + incremental / Raw-MD5	password and root

Recommended controls

- Use long, unique passphrases and block commonly breached values rather than relying only on composition rules.
- Apply password managers and multi-factor authentication so compromise of one password is not reusable elsewhere.
- Protect offline credential material: restrict shadow/SAM access, monitor hive exports, and minimize administrative sessions.
- Prefer modern salted, memory-hard password hashing. NTLM and Raw-MD5 are optimized for speed and therefore attacker-friendly.

Task B.3 closure: Cain & Abel was executed in the assigned Windows 7 VM. Direct screenshots document local hash import, a successful dictionary recovery, and a successful numeric brute-force recovery.

Reference

CYSE 301: Cybersecurity Technique and Operations, Assignment 5: Password Cracking, instructor-provided two-page course handout (source filename: "Assignment 4 - Password Cracking (System Users).pdf").

End of report