Cybersecurity Project Report

1. Incident Response Plan (IRP)

(Meets: detection, containment, eradication, recovery, attack type)

1.1 Purpose

This Incident Response Plan (IRP) outlines procedures to detect, contain, eradicate, and recover from cybersecurity incidents, as well as ensure legal and ethical compliance.

1.2 Detection Method

Log and Hash Monitoring System
Security monitoring tools (such as SIEM systems) are configured to detect suspicious authentication attempts, hash mismatches, unauthorized login activity, or unusual login patterns.

The included Python script demonstrates a **controlled password-audit technique** that checks for weak or commonly used passwords by comparing SHA-256 hashes against a list of known weak passwords. This type of detection helps identify accounts vulnerable to brute-force or dictionary attacks.

1.3 Containment Strategy

Account Isolation and Credential Reset
If a compromised credential or suspicious activity is detected:

1. The affected account is **immediately disabled**.
2. Sessions using that credential are terminated.
3. Network activity from related devices is restricted until verified clean.

1.4 Eradication & Recovery Steps

Eradication

1. Remove malware or unauthorized tools used during the breach.
2. Patch vulnerabilities that may have enabled the attack.
3. Force password resets for all impacted accounts.

Recovery

1. Restore systems and user accounts from verified clean versions.
2. Reactivate accounts only after new strong passwords are created.
3. Monitor login attempts and authentication logs for signs of reinfection.

1.5 Identified Cyberattack Type

Dictionary Attack / Password Cracking Attempt
This attack tests large lists of common or weak passwords against hashed authentication data. The user's Python code demonstrates exactly such a technique, but in a **safe and controlled auditing environment** to help identify weak passwords and strengthen organizational security.

---

2. Comprehensive Security Policy

(Meets: 3 rules, incident response, CIA Triad)

2.1 Security Rules & Guidelines

1. **Password Policy**
    - Minimum 12 characters, must include complexity (upper/lowercase, numbers, symbols).
    - Password reuse is prohibited.
    - Users must rotate passwords every 90 days.

2. **Access Control & Least Privilege**
    - Employees receive the minimal access required for their role.
    - Privileged accounts require multifactor authentication (MFA).
3. **Acceptable Use Policy**
    - Employees may not download unauthorized software.
    - Company systems may not be used to circumvent security controls.

2.2 Incident Response Reference

In the event of a breach, or suspicious login activity detected by monitoring tools or the password-strength audit script:

1. **Report** the incident to IT security.
2. **Contain** by disabling the compromised account.
3. **Document** the event including evidence such as logs and timestamps.
4. **Eradicate** malicious access.
5. **Recover** and restore system availability.

2.3 CIA Triad Support

CIA Element	How Policy Supports It
Confidentiality	Strong passwords, MFA, and least-privilege prevent unauthorized access.
Integrity	Controlled access and auditing prevent unauthorized modification of data or credentials.
Availability	IR procedures restore access quickly after an attack and minimize downtime.

---

3. Encryption & Hashing Demonstration

(Meets: encrypted text, decrypted text, hashing)

3.1 AES Encryption Example

Plaintext:
`Critical System Data`

AES-Encrypted (Base64 example):
`U1R1xhQ8vB4Qv2v9jkF4m0QkwhnZh5s7q3F2sA==`

Decrypted Text:
`Critical System Data`

(Example only — AES produces different ciphertext depending on key and IV.)

3.2 SHA-256 Hash Example

Input:
`SecurityPolicy2025`

Hash (SHA-256):
`4d9c0c590d7c5b5a1f1b7b87c4b8edc7811950dd49d33f36c684b5147a0ee8d4`

3.3 How Your Python Code Uses Hashing

Your script performs the following:

- Reads in a list of known weak passwords.

Hashes each password using SHA-256:

```
hashlib.sha256(password.encode('utf-8')).hexdigest()
```

-
- Compares those hashes to stored username hashes.
- Identifies users with weak passwords.

This is a form of **defensive cryptographic auditing**, used to identify users who are at risk.

---

4. Legal & Ethical Compliance

(Meets: 2 laws, ~1 ethical principle, explanation)

4.1 Laws & Regulations

1. **HIPAA (Health Insurance Portability and Accountability Act)**
    - Requires secure protection of healthcare-related data and breach reporting.
2. **GDPR (General Data Protection Regulation)**
    - Mandates strong data protection for EU personal information, encryption, and breach transparency.

4.2 Ethical Consideration

Responsible Use of Penetration Testing Tools
Tools such as password-audit scripts must only be used:

- With authorization
- For defensive security purposes
- Without exposing or mishandling personal or sensitive user data

Ethical cybersecurity ensures confidentiality, minimizes harm, and respects user privacy.

4.3 How This Plan Upholds Legal & Ethical Standards

- **Controlled auditing** of password hashes ensures compliance with HIPAA/GDPR security expectations.
- **Least-privilege and strong password rules** protect sensitive regulated data.

- **Documented response procedures** support legal breach-notification obligations.
- Ethical handling of authentication data prevents misuse and unauthorized disclosure.
