Forensic Hash Algorithm Report: MD5, SHA-1, and SHA-256

Overview

This report provides a detailed analysis of three cryptographic hash algorithmsMD5, SHA-1, and SHA-256in the context of password cracking, as relevant to a forensic password cracking tool. The tool, designed for authorized forensic investigations, supports MD5 and SHA-1 for cracking passwords using a dictionary attack. This report covers the properties, security considerations, and forensic implications of these algorithms, including why SHA-256 is not supported in the provided script.

1. MD5 (Message Digest Algorithm 5)

1.1 Description

MD5 is a widely used cryptographic hash function designed by Ronald Rivest in 1991. It produces a 128-bit (16-byte) hash value, typically represented as a 32-character hexadecimal string (e.g., `5f4dcc3b5aa765d61d8327deb882cf99` for the input "password").

1.2 Properties

- **Output Length:** 128 bits (32 hexadecimal characters).
- **Block Size:** 512 bits.
- **Speed:** Fast, optimized for 32-bit processors.
- **Collision Resistance:** Broken; collisions can be generated efficiently (e.g., two different inputs producing the same hash).
- **Preimage Resistance:** Vulnerable to preimage attacks with sufficient computational resources.
- **Usage in Script:** Supported for dictionary attacks. The script validates MD5 hashes using the regex `^[0-9a-fA-F]{32}$` and computes hashes using `hashlib.md5()`.

1.3 Security Analysis

- **Vulnerabilities:**
  - MD5 is considered cryptographically broken due to its susceptibility to collision attacks. In 2004, researchers demonstrated practical collision attacks, making it unsuitable for secure applications like digital signatures or password hashing in modern systems.
  - Preimage attacks (finding an input for a given hash) are computationally feasible with tools like rainbow tables, especially for weak passwords.
- **Forensic Relevance:**
  - MD5 is commonly encountered in legacy systems, older password databases, or misconfigured applications, making it relevant for forensic investigations.

  - The scripts dictionary attack is effective for MD5 because of its speed and the availability of precomputed hash tables for common passwords.

- **Example:**
  - Input: password
  - MD5 Hash: `5f4dcc3b5aa765d61d8327deb882cf99`
  - The script successfully cracked this hash (as shown in prior output) when the wordlist contains "password".

1.4 Forensic Considerations

- **Ease of Cracking:** MD5s short hash length and fast computation make it susceptible to brute-force and dictionary attacks, especially with common passwords.
- **Log Analysis:** The script logs each attempt (e.g., `Attempting password: password (Hash: 5f4dcc3b5aa765d61d8327deb882cf99)`), providing a forensic audit trail.
- **Legal Note:** Cracking MD5 hashes requires legal authorization, as unauthorized access to password hashes violates privacy and data protection laws.

2. SHA-1 (Secure Hash Algorithm 1)

2.1 Description

SHA-1, developed by the NSA in 1993, is a cryptographic hash function producing a 160-bit (20-byte) hash value, represented as a 40-character hexadecimal string (e.g., `5baa61e4c9b93f3f0682250b6cf8331b7ee68fd8` for "password").

2.2 Properties

- **Output Length:** 160 bits (40 hexadecimal characters).
- **Block Size:** 512 bits.
- **Speed:** Slower than MD5 but still relatively fast.
- **Collision Resistance:** Broken; practical collision attacks demonstrated in 2017 (e.g., Googles SHAttered attack).
- **Preimage Resistance:** Weaker than modern algorithms but stronger than MD5.
- **Usage in Script:** Supported for dictionary attacks. The script validates SHA-1 hashes using the regex `^[0-9a-fA-F]{40}$` and computes hashes using `hashlib.sha1()`.

2.3 Security Analysis

- **Vulnerabilities:**
  - SHA-1 is no longer considered secure for cryptographic purposes due to collision vulnerabilities. In 2017, researchers generated two different PDF files with the same SHA-1 hash, proving its insecurity.

    - Preimage attacks are less practical than for MD5 but still feasible with significant resources, especially for weak passwords.

- **Forensic Relevance:**
    - SHA-1 is found in legacy systems, older Git repositories, and some password storage mechanisms, making it relevant for forensic analysis.
    - The scripts dictionary attack can crack SHA-1 hashes for common passwords, though its slightly slower than MD5 due to the longer hash length.

- **Example:**
    - Input: password
    - SHA-1 Hash: `5baa61e4c9b93f3f0682250b6cf8331b7ee68fd8`
    - The script would crack this hash if the wordlist contains "password".

2.4 Forensic Considerations

- **Ease of Cracking:** SHA-1s longer hash length makes it slightly harder to crack than MD5, but dictionary attacks remain effective for weak passwords.
- **Log Analysis:** Similar to MD5, the script logs each SHA-1 attempt, aiding forensic documentation.
- **Legal Note:** As with MD5, cracking SHA-1 hashes requires legal authorization.

3. SHA-256 (Secure Hash Algorithm 256)

3.1 Description

SHA-256, part of the SHA-2 family designed by the NSA in 2001, produces a 256-bit (32-byte) hash value, represented as a 64-character hexadecimal string (e.g., `5e884898da28047151d0e56f8dc` for "password").

3.2 Properties

- **Output Length:** 256 bits (64 hexadecimal characters).
- **Block Size:** 512 bits.
- **Speed:** Slower than MD5 and SHA-1 due to increased computational complexity.
- **Collision Resistance:** No practical collision attacks known as of August 2025.
- **Preimage Resistance:** Strong, making preimage attacks computationally infeasible with current technology.
- **Usage in Script:** Not supported. The script only handles MD5 and SHA-1, likely due to their prevalence in legacy systems and faster computation for dictionary attacks.

3.3 Security Analysis

- **Vulnerabilities:**
  - SHA-256 is considered cryptographically secure for most purposes, with no known collision or preimage attacks as of 2025.
  - Its longer hash length and complex design make it resistant to brute-force and dictionary attacks compared to MD5 and SHA-1.
- **Forensic Relevance:**
  - SHA-256 is widely used in modern systems, including password hashing (e.g., in bcrypt or PBKDF2), cryptocurrency (e.g., Bitcoin), and digital signatures.
  - Cracking SHA-256 hashes with a dictionary attack is significantly harder due to its computational cost and larger hash space, which is why the script excludes it.
- **Example:**
  - Input: password
  - SHA-256 Hash: `5e884898da28047151d0e56f8dc6292773603d0d6aabbdd62a11ef721d1542d`
  - A dictionary attack on SHA-256 would require significantly more time and resources, making it impractical for the scripts scope.

3.4 Forensic Considerations

- **Ease of Cracking:** SHA-256s security makes it resistant to dictionary attacks, especially for strong passwords. Specialized hardware (e.g., GPUs) and large wordlists are needed, but success is unlikely without weak passwords.
- **Why Not in Script:** The script focuses on MD5 and SHA-1 because:
  - These algorithms are common in older systems targeted by forensic investigations.
  - Their faster computation allows quicker dictionary attacks.
  - SHA-256s computational cost and stronger security make it less practical for the scripts purpose.
- **Recommendation:** For SHA-256, forensic investigators would need advanced tools like GPU-based crackers (e.g., Hashcat) and larger wordlists, or alternative methods like social engineering or keyloggers (with legal authorization).

4. Comparison of MD5, SHA-1, and SHA-256

Property	MD5	SHA-1	SHA-256
Hash Length	128 bits (32 hex chars)	160 bits (40 hex chars)	256 bits (64 hex chars)
Block Size	512 bits	512 bits	512 bits
Collision Resistance	Broken (2004)	Broken (2017)	Secure (no known attacks)
Preimage Resistance	Weak	Moderate	Strong
Speed	Very fast	Fast	Slower
Script Support	Yes	Yes	No
Forensic Relevance	Legacy systems, easy to crack	Legacy systems, moderately hard	Modern systems, hard to crack

5. Forensic Implications and Script Context

5.1 Why MD5 and SHA-1 in the Script?

- **Legacy Systems:** MD5 and SHA-1 are prevalent in older password databases, configuration files, and forensic artifacts, making them prime targets for password cracking in investigations.
- **Speed:** Both algorithms are computationally inexpensive, allowing rapid dictionary attacks, as implemented in the scripts `dictionary_attack` function.
- **Practicality:** The scripts design prioritizes speed and simplicity for forensic use, where weak passwords and legacy hashes are common.

5.2 Why SHA-256 is Excluded?

- **Computational Cost:** SHA- prioritization of modern systems, SHA-256 is slower to compute, reducing the efficiency of dictionary attacks.
- **Security:** Its stronger resistance to attacks makes it less likely to yield results in a simple dictionary attack, requiring more advanced tools.
- **Utility:** The script targets legacy hashes commonly encountered in forensic investigations, where SHA-256 is less prevalent.

5.3 Example from Script Execution

- **Command:** `python passwordcrack.py 5f4dcc3b5aa765d61d8327deb882cf99 md5 wordlist.txt`
- **Issue:** The script failed due to a missing `wordlist.txt`, producing:

```
=== Forensic Password Cracker Report ===
Hash: 5f4dcc3b5aa765d61d8327deb882cf99
Hash Type: Unknown
Status: Error
Message: Wordlist file not found: wordlist.txt
=== End of Report ===
```

- **Analysis:** If `wordlist.txt` existed with "password", the script would have cracked the MD5 hash successfully. For SHA-1 or SHA-256, similar logic applies, but SHA-256 would require a modified script with `hashlib.sha256()`.

6. Recommendations for Forensic Investigators

- **MD5 and SHA-1:**
  - Use the provided script for quick dictionary attacks on weak passwords.
  - Ensure a comprehensive wordlist (e.g., RockYou or custom lists) to increase success rates.
  - Verify legal authorization before attempting to crack hashes.
- **SHA-256:**
  - Use advanced tools like GPU-based crackers (e.g., Hashcat) and larger wordlists.
  - Consider alternative forensic methods (e.g., analyzing metadata or user behavior) if dictionary attacks fail.
- **General:**
  - Maintain detailed logs (as the script does) for audit trails.
  - Store reports securely, as they contain sensitive information (e.g., cracked passwords).
  - Update the script to support SHA-256 if modern systems are targeted, using `hashlib.sha256()` and a regex like `^[0-9a-fA-F]64`.

7. Conclusion

MD5 and SHA-1 are legacy hash algorithms supported by the forensic password cracking script due to their prevalence in older systems and susceptibility to dictionary attacks. SHA

SHA-256, while more secure, is not included due to its computational complexity and stronger resistance to cracking, making it less practical for the scripts scope. Forensic investigators should use the script for MD5 and SHA-1 with appropriate wordlists and legal authorization, while relying on advanced tools for SHA-256.

End of Report