

Homework

Three tasks covering GPU benchmarking, workload optimization, and performance documentation.

Complete all six lab exercises before starting the homework. These tasks build on the skills and observations from the lab.

Task 1 — Comprehensive GPU Benchmark

Objective

Benchmark your GPU across multiple hash types and create a performance reference sheet for your hardware.

Instructions

1. Run `hashcat -b` to benchmark all hash types (or select at least 10 common types)
2. Record the results in a table with columns: Hash Mode, Hash Name, Speed (H/s), and Relative Speed (compared to MD5)
3. Include at minimum these hash types:
 - MD5 (`-m 0`)
 - SHA-1 (`-m 100`)
 - SHA-256 (`-m 1400`)
 - NTLM (`-m 1000`)
 - bcrypt (`-m 3200`)
 - WPA-PBKDF2 (`-m 22000`)
 - SHA-512 (`-m 1700`)
 - MD5crypt (`-m 500`)

- sha512crypt (`-m 1800`)
 - PBKDF2-HMAC-SHA256 (`-m 10900`)
4. Calculate how long each hash type would take to exhaust a 6-character alphanumeric keyspace ($62^6 = \sim 56.8$ billion candidates)
 5. Write a brief analysis: which hash types are GPU-friendly and which are GPU-resistant?

Deliverable

A completed benchmark table with speed data, estimated exhaustion times, and a written analysis of GPU-friendly vs GPU-resistant algorithms.

Self-check: Your table has at least 10 hash types. Exhaustion time calculations are correct. Your analysis correctly identifies bcrypt and scrypt as GPU-resistant.

Task 2 — Workload Optimization for a Target Hash

Objective

Find the optimal Hashcat settings for cracking a specific hash type on your hardware.

Instructions

1. Choose a hash type that takes at least a few seconds to benchmark (e.g., bcrypt `-m 3200`, or WPA `-m 22000`)
2. Test all four workload profiles (`-w 1` through `-w 4`) and record the speed for each
3. Experiment with kernel tuning flags:
 - Try different `-n` values (e.g., 64, 128, 256, 512, 1024)
 - Try different `-u` values (e.g., 64, 128, 256, 512)
 - Record the speed for each combination
4. Find the combination that gives the highest speed without causing errors or overheating
5. Compare your optimal settings against Hashcat's auto-tuned defaults

6. Document the temperature behaviour under each configuration

Deliverable

A report showing all tested configurations with their speeds and temperatures. Identify the optimal configuration and explain why it outperforms (or matches) the defaults.

Self-check: You tested at least 8 different configurations. Your optimal settings are clearly identified. Temperature data is included for each test.

Task 3 — CPU vs GPU Performance Report

Objective

Create a detailed performance comparison between CPU and GPU cracking on your system.

Instructions

1. Benchmark at least 5 hash types on both CPU (`-D 1`) and GPU (`-D 2`)
2. For each hash type, calculate the speedup factor (GPU speed / CPU speed)
3. Run a real dictionary attack on both CPU and GPU for at least 2 hash types.
Record:
 - Wall-clock time to completion
 - Hashes per second
 - Power consumption (if measurable with `nvidia-smi`)
4. If you have access to multiple GPUs, test scaling: 1 GPU vs 2 GPUs. Calculate the scaling efficiency.
5. Write a summary answering:
 - What is the average GPU speedup on your system?
 - Which hash types benefit most from GPU acceleration?
 - Are there any hash types where CPU performance is competitive?
 - What are the diminishing returns of multi-GPU setups?

Deliverable

A performance report with benchmark tables, speedup calculations, and a written summary answering the questions above.

Self-check: At least 5 hash types compared. Speedup factors are calculated correctly. Your summary is supported by the data.

Submission Checklist

- Task 1: Benchmark table with 10+ hash types, exhaustion time estimates, and GPU-friendly analysis
- Task 2: Workload optimization report with tested configurations, speeds, and temperatures
- Task 3: CPU vs GPU comparison with speedup factors and written analysis
- All commands executed without errors; results are reproducible