Module Final report

Marcus Sung

CS7NS1/CS4400 Module Final Report

- Student name: Marcus Sung
- Student number: 14335274
- Date:

InfernoBall team self-evaluation

- Team number: 11
- Team members:
  - Marcus Sung
  - John Aitling
  - Xuming Xiu
  - Sahir Sharma
- Insert team member names/initials in column headers of table below
- Effort must be high/medium/low/zero
- Effectiveness must be high/medium/low/zero

Use zero if someone really didn't participate at all. Include your evaluation of yourself as well as other team members. Be honest.

	Marcus	John	Xuming	Sahir
effort	high	medium	low	low
effectiveness	high	low	low	low

What I learned (1-2 pages)

I learned various things from going to lectures (normal and guest) and doing the written assignments, and the practicals.

Lectures

- What I learn about Scalable Computing (general):
  - Things to look out for in scalable computing: Scalability, adaptability, dispersibility, accessibility, affordability, and reliability.
  - This helped when reading the papers for the written assignments.
- What I learnt about Internet of Things:
  - IoT and its structure.
  - IoT nodes and their constraints (low resources, low computing power, low memory etc)
  - Architectures of IoT systems.
  - Applications and the benefits and challenges.
  - Industry 4.0
- What I learnt about processing units:
  - The differences between CPUs and GPUs.
  - The cases in which each one is good for.
  - Advantages and disadvantages of them.
  - CPU and GPU orientated programming (OpenCL).
  - OpenCL specific programming approach.
- What I learnt about passwords:
  - Advantages and disadvantages of passwords
  - Best practices in managing and creating passwords
  - Password alternatives e.g biometrics
  - Cryotographic hashes, how they work and the different kinds
  - Some ways to attack a hash and crack it
- What I learnt about failures:
- Different causes of failure
- Inevitable but can learn a lot from them
- What happens when a failure occurs and how they are dealt with
- How to prevent them

Assignments

hashcat This is what I learnt doing the written assignments:

- Assignment 1 (Scalable Computing):
  - Application of IoT in personal healthcare:
    - RFID technology
    - Environmentally passive sensors and body centric sensors
    - Privacy, safety and reliability concerns
  - Smart Cities:
    - Architecture
    - Features (possible)
    - Technology involved in realising the concept (Already done in Padova)
- Assignment 2 (CPU and GPU):
  - Language based approach
    - Parallelism and reliability integrated into a language (MISO)
    - Benefits of language based approach for low powered computing
  - Energy efficiency in clustered many-cores:
    - Programmable tightly-coupled clusters of processors.
    - Pros and cons of using it.
    - Applications.

- Assignment 3 (Cloud/Edge or Fog Computing):
  - Mobile edge computing:
    - How MEC works and the architecture.
    - Pros and cons of using MEC.
    - Limitations on research in particular security.
  - Fog Computing and IoT applications:
    - Security and privacy not researched enough.
    - Advantages of using fog computing.
    - Limitations of fog computing.
    - Architecture of fog computing.

Practicals

Doing the practicals I learnt the following things.

- Practical 1(Compiling John The Ripper):
  - How to register with rosettahub
  - How to create and stop an instance and how to SSH into it
  - How to clone into github and compile tools such as John The Ripper
  - How to run bash scripts (I had no previous knowledge on scripting of any sort)
- Practical 2(1k passwords):
  - Different hash types and formats
  - The various algortithms and methods that JTR could use
  - Methods such as brute-forcing, dictionary attacks, mask attacks, rules etc
  - Salted vs unsalted hashes differences more clearly
  - Sending files to the instance and retrieving files from the instance (scp)
- Practical 3(More hashes):
  - How to set up a GPU instance
  - How to save images on rosettahub
  - Compile and run hashcat
  - Various different hashcat modes
  - Difference between GPU and CPU cracking
  - Implementing rules attacks, combinator attacks, prince attacks and mask attacks on hashcat
  - Writing scripts in python and bash to automate tasks
  - Using various command line tools such as sed, grep, awk etc to edit text files in turn automating tasks
  - Using tmux to leave instances running in the background
  - Slow and fast hashes differences
  - Multiple GPU vs singl GPU differences
  - Ram management when dealing with different methods so you don't run out of memory
  - Optimising attack methods e.g using word list compiled of all mask attack iterations is a bit faster then using mask attack purely
  - Vertical scaling and horizontal scaling
  - Using Google cloud scaled vertically (e.g adding more power to a single instance)
  - Running AWS and Google Cloud at the same time offered horizontal scaling (running more machines)
  - Collobaration can help speed things up
  - Slow hashes such as argon were very hard to crack in a reasonable time since they could only be cracked using CPU attacks
- Practical 4 (Group assignment):
  - How Shamir sharing worked
  - How to be more efficient working in groups cracking hashes
  - How to work with a team that did not put much effort in or were not that knowledgable in the module
  - Sometimes it is quicker to do things yourself instead of waiting for others to do it (we dropped from the top 5 in the leaderboard to the bottom 5 whilst I waited for them to do some tasks)
  - Splitting hashes between many would be the ideal way to crack faster

What I did (2-4 pages)

Describe what you did during the module, i.e., how you solved the practicals. Include URLs for code/repos but not the code itself. If you have more text than fits in 4 pages, include a link to that additional text but do make the 4 pages self-contained (other than code).

Module evaluation (1 page)

Say what you liked/disliked about the module and why. There's no need to say that RosettaHub is creaky, we know that:-)