Scripts and trace analysis

TDTS11: Computer Networks and Internet Protocols

Carl Magnus Bruhner

February 2021

LINKÖPING UNIVERSITY

Lesson agenda

- Overview and walkthrough of assignment 4
- Scripts – how they work and what to use
- Questions

Assignment 4 overview

- Based on the concept of problem-based learning
- A total of 46 points, but you only need 16 to pass
- Help each other – it's helps your own learning and understanding
- Please don't hesitate to raise any concerns with the tasks (email or lab sessions)
- This year: more potential tasks—and less points required to pass!

Assignment 4 task categories

- Task 1, 7-8: Descriptions, reasoning, estimates (4 pt)
- Task 2-6: Data collection, traffic analysis and scripting (20 pt)
- Task 9-11: Programming, partly labs from TDTS04/06 (22 pt)

Review the tasks, and choose ≥16 points worth of tasks to tackle!

(Walkthrough of assignment 4)

Useful tips for Wireshark

- Export text files to analyze
    - All packets or specified range
    - Captured or displayed (combine with filter)
    - Details, and if so; collapsed, displayed or expanded?
- Follow HTTP and/or TCP stream to filter out main/secondary content
- Experiment with Statistics for the known protocols for summarized data
- Save the original trace as pcap(ng) to share with lab partner or university computers

Useful tips for scripts

- Experiment in terminal, but create bash scripts (.sh) for the tasks
- Most useful commands:
  awk, cat, grep, sed, sort, uniq, wc
- Some examples in the very end of the assignment description
- Use “man [command]” for more information

Commands (part 1)

- `awk`
  "Aho, Weinberger, and Kernighan"
  Interpreted programming language for processing text

- `cat`
  Reads data from a file and outputs the content.

- `grep`
  "global regular expression print"
  Searches text line by line with regular expressions and returns the line with matches. Can be used with -c flag to count lines instead.

Commands (part 2)

- `sed`
  “stream editor”
  Filters and transforms text. Useful for editing text lines. Example is **`sed 's/search/replace/g'`** with substitutes (s) *search* with *replace* for all non-overlapping matches (g).

- `sort`
  Sorts or merges the text lines. Can be numerically sorted with -n flag

- `uniq`
  Filter out repeated lines in a file

- `wc`
  Count the words (wc, word count) in a file. Can be used with -l flag to count the lines.

Alternatives

- Tasks is mainly adopted to Wireshark and bash scripts
- Most tasks can be answered both with scripts and by looking in Wireshark, but automating them with scripts saves a lot of time
- Programming is not required, but encouraged. Any language is ok!

Questions?

carbr307@student.liu.se (subject “TDTS11 […]”)

www.liu.se

LINKÖPING
UNIVERSITY