INTI
LAUREATE INTERNATIONAL UNIVERSITIES®

University of Wollongong

FACULTY OF INFORMATICS

ASSIGNMENT(2) SPECIFICATION – 10%

Part One:

1. Describe how a honeypot could be used to detect spam, and aid spam filters on ”real systems”.

[2Marks]

2. In what way can personalised login screens provide protection? Explain your answer carefully

[2Marks]

Part Two: Logging and intrusion detection systems **[36 Marks]**

You are to implement, in Java or C/C++, an intrusion detection system that raises alerts in terms of information found in log files which are generated from “raw” traffic files. There are two parts, the Filter and the Analyser, and you should include a Readme.txt containing compilation instructions allowing Filter and Analyser to be produced. A makefile or script for compilation would be nice.

The Readme.txt file needs to contain about the formats you choose to use for representing the filter and analysis policies.

You will be provided with sample files Port4000.txt up to Port4020.txt, and your program should be designed to work on a collection of files such as those. Each file represents traffic on particular ports.

Each file is of the form,

AxKrLxLwZwCrXwFx...

where the capital letters A–M represent users accessing the port associated with the file, and the action they carry out being represented as r (read), w (write) or x (execute). A capital letter X indicates an anonymous user, and Z indicates no connection to the port in that particular time block.

Each file will have the same number of entries. Each day has 30 pairs of (user, action) associated it, and the 60 symbols will appear on the same line in each of the port files.

You are to write a file filterPolicy.txt containing filter policies that your code Filter can use to filter the traffic represented in the port files and produce a file logs.txt. The filter should run as

Filter start end

with the start and end being the lowest and highest port numbers. Only port files in that range need to be considered. The port files will always be named Port[number].txt.

You are to use your own format/grammer for the file filterPolicy.txt to implement the following filter policies:

Rule 1: Record when ports 4010 or 4011 have been inactive for 5 time blocks in a row.

Rule 2: Record all actions relating to the anonymous user.

Rule 3: Record when any user has attached 4 or more times to a port in a day.

Rule 4: Record when a user has accessed 3 or more ports at the same time.

Rule 5: Record any execute actions on port 4013.

These policies must not be hard coded into the source for Filter. It should be possible to change the numerical values, including the port numbers and thresholds, and the relevant actions (read, write or execute) being considered, all within the filterPolicy.txt. The number of policies could also be changed.

You are also to write a file alertPolicy.txt containing policies that your code Analyser can use to analyse the file log.txt and produce a file alerts.txt, containing "abnormal events". You are to use your own format/grammer for alertPolicy.txt to represent the following policies requiring alerts:

Case 1: When there have been 5 or more executes on 4013 in a day.

Case 2: When the anonymous user has written on port 4017 3 or more times in a day.

Case 3: When Rule 3 is triggered 2 or more times in a day.

Case 4: When Rule 4 is triggered 2 or more times in a day.

Note that for Case 3 and Case 4 they need to explicitly state they relate to the third Rule or fourth Rule in the filterPolicy.txt file. If a Case relates to a Rule that doesn't exist a note should be made. Note also that the Cases could contain statements relating to information which the Rules do not capture. You do not need to check for such consistency, simply generate the logs.txt file in accordance with the filter policies, and process the logs.txt file in accordance with the alert policies.

Again it should be possible to change the entries in the alertPolicy.txt file to vary the numbers in the cases, or the number of cases.

The file alerts.txt should be updated with the alerts for each day, with the day clearly denoted.

Due date: 1/10/2014