Implementation: Analyzer

DaCoPAn

Helsinki 31st May 2004

Software Engineering Project

UNIVERSITY OF HELSINKI
Department of Computer Science

UNIVERSITY OF PETROZAVODSK
Department of Computer Science

Course

581260 Software Engineering Project (6 cr)

Project Group

Carlos Arrastia Aparicio
Jari Aarniala
Alejandro Fernandez Rey
Vesa Vainio
Jarkko Laine
Jonathan Brown

Kirill Kulakov
Andrey Salo
Andrey Ananin
Mikhail Kryshen
Viktor Surikov

Customer

Markku Kojo

Project Masters

Juha Taina (Supervisor)
Yury Bogoyavlenskiy (Supervisor)

Turjo Tuohiniemi (Instructor)
Dmitry Korzun (Instructor)

Homepage

`http://www.cs.helsinki.fi/group/dacopan`

Change Log

Version	Date	Modifications
1.0	20.05.2004	First version

Contents

1 Introduction

This document defines Implementation for Analyzer subsystem of the DaCoPAn software according to [2]. It is focused on changes to the Design document [1] that were necessary for the implementation phase. All modifications that are included into the Design documents have a short descriptions which are included into the Implementation document.

Architecture modifications are described in section 2. Changes in data structures are presented in section 3. Module modifications are described in section 4. Changes in behavioral models are described in section 6.

This document is intended mainly for the project development team. Experts from the customer's side may analyze this document to be sure that the requirements are going to be implemented sufficiently and efficiently.

This specification may be changed during the testing phase. All such changes must be shortly described and grounded in a separate document — Test execution document.

2 Architecture

All changes was added in design document with corresponding descriptions.

Log reader
Log reader module check used application protocol. If one of port numbers is defined then sets corresponded application layer.

3 Data structures

All changes was added in design document with corresponding descriptions.

link
Added structure `link` for definition constant variables of connections between two hosts (see sections **5.5 "Events sequence", 6.5.2 "Calculator"**).

host
Added field `timealign` for each host. (see section **5.5 "Events sequence"**).

event
Field `variables` changed to `vars`. Added fields `unit_prev`, `host_prev`, `host_next`, `host_prev`. Added fields `child`, `last_parent`, `next_parent`, `index`. Changed type of `data` to `void`. Also `app_ports` structure was added. See section **5.5 "Events sequence"**.

PTU
Field `ts` changed to `timestamp`. Field `h` changed to `host`. `trans_event` and `app_event` are no longer needed. Unknown protocol constant for each layer was added (see section **5.1 “Packet trace unit”**).

variables
Field `TCPstate` changed to `tcp_state`. Field `droptime` was moved to separate event `UNIT_LOST`. Type of transition time was changed to `timeval` (see section **5.5 “Events sequence”**).

4 Modules

All changes was added in design document with corresponding descriptions.

Command line parser
`http` and `dns` options was added for specifying application layer protocols. `time-diff` option is change to `time-align`. `program_invocation_name` variable was added. Macros `const *char anlz_version` and `const *char anlz_help` is changed to `void` and used instead of string. (see section **6.2 “Command line parser”**).

Log reader
Linux cooked sockets support was added for reading. `data_link_type` field was added. `read_file` function was changed to read_log function and `file` argument was replaced by `host`. `raw_callback` and `ipv4_callback` functions was removed and one callback function `lr_callback` was added for all data link types. Packet processing scheme was changed in compliance with implementation. Functions `read_linux_ssl, read_ether, read_app_unknown` was added in the list of functions for processing corresponding protocols data. (see section **6.3 “Log reader”**).

Message mapper
`mapping_messages` is changed to `map_messages`. Direction of message link in `linking` functions is set from received packet to sended packet. (see section **6.4 “Message mapper”**).

Events calculator
Change direction of message links on the diagrams. `ptu_sequence, links, flows, hosts` arguments was added in the `split` routine. Calculator is also responsible for building and adding ’dropped’ events in the events sequence, for filling links and flows lists. Field `id` was added in `link` structure. Type of the `tt` was changed to `timeval`. In `calculate` routine `ptu_seqeunce, links, flows, add_log_var` arguments was added. `droptime` variable is the timestamp of ’dropped’ event. Files in

Implementation notes was arranged in four group of files: common group, group of files for each submodule, and files, which are used by Layer splitter submodule. (see section **6.5 “Events calculator”**).

PEF writer
`links, flows` arguments was added in the `pef_write` and `pef_writef` routines. `pefwrite_http.c, pefwrite_http.h` routines was added in the implementation notes. (see section **6.6 “PEF writer”**).

5 Behavioral model

Command line parser
The command line parser can finish Analyzer work if user specified option `--help` or `--version`, or used wrong command line syntax (see sections **6.2 “Command line parser”, 7.2 “Get usage info”, 7.3 “Get program info”, 7.4.2 “Wrong command line syntax”**).

6 Features

In this section features are described, which are not implemented in this version of the Analyzer.

Events calculator
Splitting algorithm is implemented only fo 2 hosts. Reading of the additional logs of variables are not implemented. Calculating HTTP states and other protocol variables is not implemented.

References

1 DaCoPAn Software Engineering project, *Design: Analyzer*. Release 1.0. Universities of Helsinki and Petrozavodsk, April 2004.

2 Taina J., Korzun D., Tuohiniemi T., Alanko T., Bogoyavlenskiy Y., *Software Engineering Project: Distributed Approach*. Release 1.0. Universities of Helsinki and Petrozavodsk, January 2004.