BinPAC++
A Next-Generation Parser Generator

Robin Sommer
International Computer Science Institute, &
Lawrence Berkeley National Laboratory

robin@icsi.berkeley.edu
http://www.icir.org/robin
Bro’s Architecture

Network

Protocol Decoding

Analysis Logic

Policy Script Interpreter

Event Engine

Logs

Notification

Events

Packets

"User Interface"
Bro’s Event Model

Web Client 1.2.3.4/

Request for /index.html

Status OK plus data

Web Server 5.6.7.8/80

1.2.3.4/4321
Bro’s Event Model

Web Client 1.2.3.4/

Request for /index.html

Status OK plus data

Web Server 5.6.7.8/80

1.2.3.4/4321

Stream of TCP packets

SYN SYN ACK ... ACK ACK ACK ... ACK FIN FIN
Bro’s Event Model

Web Client 1.2.3.4/

1.2.3.4/4321

Request for /index.html

Status OK plus data

Web Server 5.6.7.8/80

Stream of TCP packets

Event

connection_established(1.2.3.4/4321 ⇒ 5.6.7.8/80)
Bro’s Event Model

Event

connection_established(1.2.3.4/4321 ⇒ 5.6.7.8/80)

TCP stream reassembly for originator

Event

http_request(1.2.3.4/4321 ⇒ 5.6.7.8/80, “GET”, “/index.html”)
Bro’s Event Model

Stream of TCP packets

- SYN
- SYN
- ACK
- ...
- ACK
- ACK
- ...
- ACK
- FIN
- FIN

Event
connection_established(1.2.3.4/4321→5.6.7.8/80)

TCP stream reassembly for originator

Event
http_request(1.2.3.4/4321→5.6.7.8/80, “GET”, “/index.html”)

TCP stream reassembly for responder

Event
http_reply(1.2.3.4/4321→5.6.7.8/80, 200, “OK”, data)
Bro’s Event Model

Request for `/index.html`

Status OK plus data

Stream of TCP packets

connection_established(1.2.3.4/4321 ⇒ 5.6.7.8/80)

http_request(1.2.3.4/4321 ⇒ 5.6.7.8/80, “GET”, “/index.html”)

http_reply(1.2.3.4/4321 ⇒ 5.6.7.8/80, 200, “OK”, data)

collection_finished(1.2.3.4/4321, 5.6.7.8/80)
Bro’s Event Model

Web Client 1.2.3.4/

1.2.3.4/4321

Request for /index.html

Status OK plus data

Web Server 5.6.7.8/80

Stream of TCP packets

SYN SYN ACK ... ACK ACK ... ACK FIN FIN

Event connection_established(1.2.3.4/4321⇒5.6.7.8/80)

Event http_request(1.2.3.4/4321⇒5.6.7.8/80, “GET”, “/index.html”)

TCP stream reassembly for originator

Event http_reply(1.2.3.4/4321⇒5.6.7.8/80, 200, “OK”, data)

TCP stream reassembly for responder

Event connection_finished(1.2.3.4/4321, 5.6.7.8/80)
Writing Protocol Analyzers For Bro

Generation 1: Manually written C++ code.
Cumbersome and error-prone.
Writing Protocol Analyzers For Bro

Generation 1: Manually written C++ code.
Cumbersome and error-prone.

Generation 2: BinPAC - “yacc for protocols”.
Better, but solves only half the problem (syntax).
Still requires C++ for semantics and Bro interface.
Writing Protocol Analyzers For Bro

Generation 1: Manually written C++ code.
   Cumbersome and error-prone.

Generation 2: BinPAC - “yacc for protocols”.
   Better, but solves only half the problem (syntax).
   Still requires C++ for semantics and Bro interface.

Generation 3: BinPAC++ - A “closed” system.
   A single, comprehensive language to write analyzers.
   No more C++ needed; not even for the Bro interface.
   Still sufficiently efficient for real-time, high-volume parsing.
Demo
In the Background: HILTI

A High-Level Intermediary Language for Traffic Inspection

Host Application
- Application Core
- Analysis Compiler
- Analysis Specification

HILTI Machine Environment
- HILTI Machine Code
- HILTI Compiler
- Runtime Library

LLVM Toolchain
- C Interface Stubs
- LLVM Bitcode
- LLVM Compiler/Linker
- Native Machine Code

BinPAC++
In the Background: HILTI

A High-Level Intermediary Language for Traffic Inspection

BinPAC++
BinPAC++ Features

Integrates syntax and semantics.
Agnostic to *what* is being parsed.
Lots of constructs for controlling parsing.
Layering of analyzers.
Just-in-time compilation.
Supports stand-alone and integrated operation.
Generates fully incremental look-ahead parsers.
Debugging support.
Resynchronization after parse errors.
*Soon*: Dynamic format detection
Parsing is becoming part of the user interface.

In some way, BinPAC++ is really just another scripting language, tailored to the domain.
BinPAC++ Vision

Parsing is becoming part of the user interface.

In some way, BinPAC++ is really just another scripting language, tailored to the domain.

... and it works outside of Bro just as well. (Wireshark, anyone?)