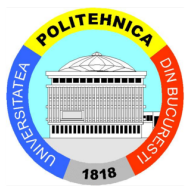


Session 8
Stateful Fuzzing

Security Summer School
July 17th 2014

ACS/Ixia/Hexcellents

Stateful Fuzzing

- Yes, it involves a state machine...

Stateful Fuzzing

- Yes, it involves a state machine...
- How do you represent a state machine ?

Stateful Fuzzing

- Yes, it involves a state machine...
- How do you represent a state machine ?
- As a directed graph, with nodes representing message requests and edges representing sequencing between the messages

Stateful Fuzzing

- Yes, it involves a state machine...
- How do you represent a state machine ?
- As a directed graph, with nodes representing message requests and edges representing sequencing between the messages
- The fuzzer walks all the paths in the graph, incrementally sending the correct messages to reach a certain node

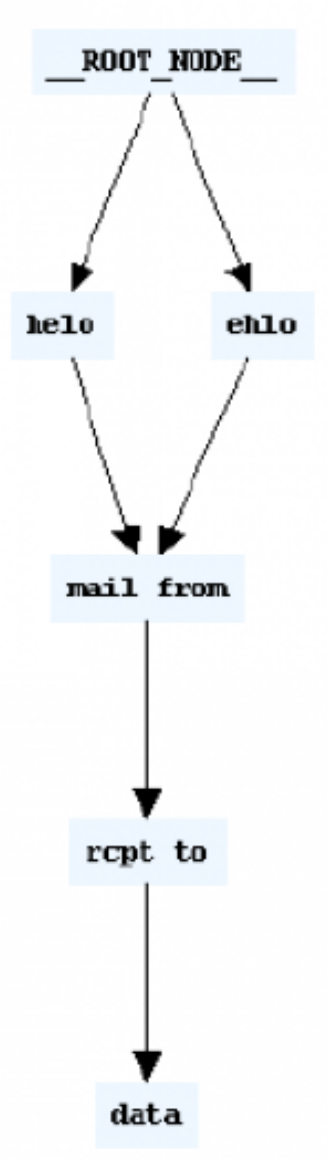
Stateful Fuzzing

- Yes, it involves a state machine...
- How do you represent a state machine ?
- As a directed graph, with nodes representing message requests and edges representing sequencing between the messages
- The fuzzer walks all the paths in the graph, incrementally sending the correct messages to reach a certain node
- Usually Depth-First-Search

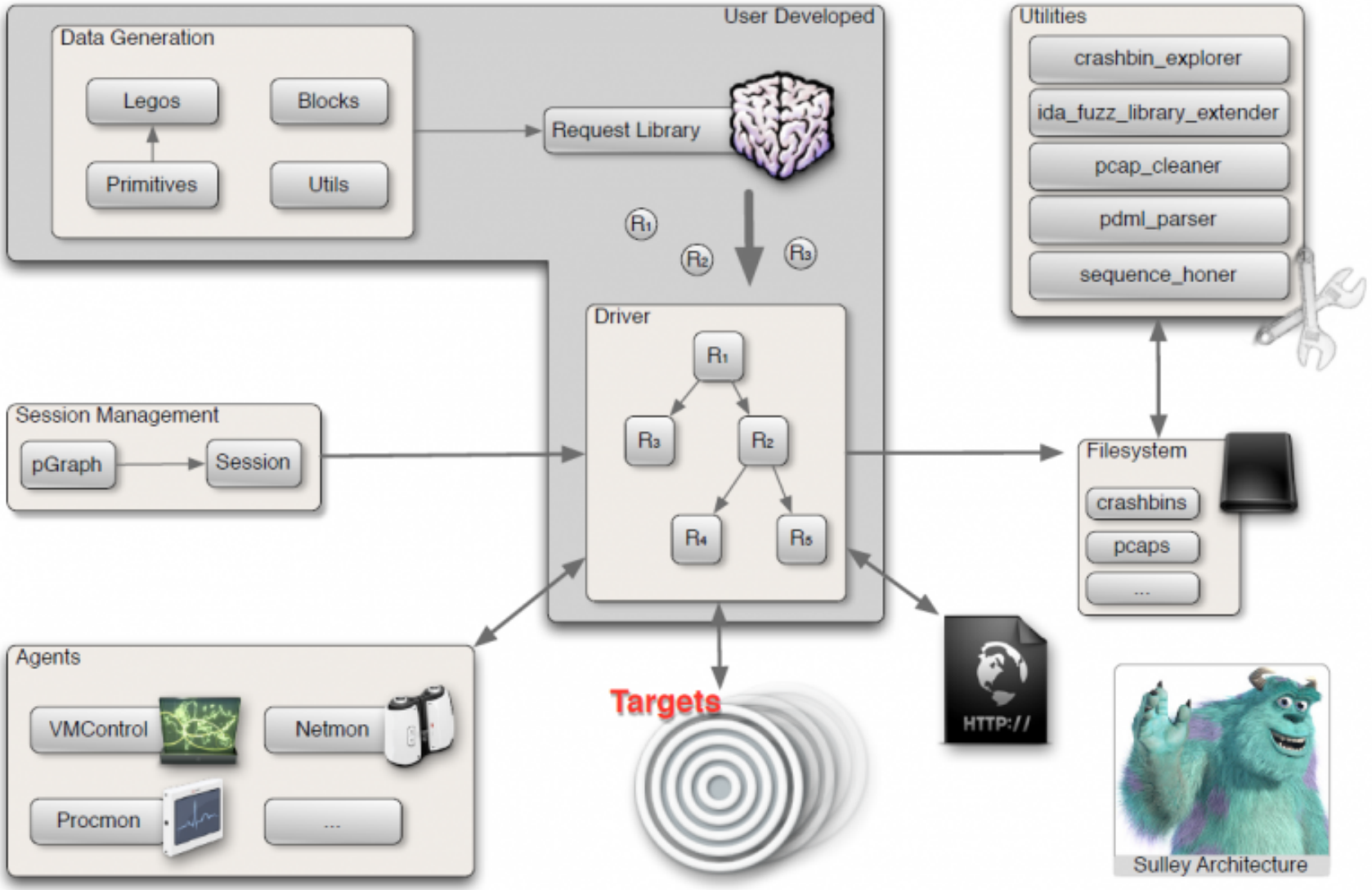
Stateful Fuzzing

- Yes, it involves a state machine...
- How do you represent a state machine ?
- As a directed graph, with nodes representing message requests and edges representing sequencing between the messages
- The fuzzer walks all the paths in the graph, incrementally sending the correct messages to reach a certain node
- Usually Depth-First-Search
- The fuzzing process is complete when all paths have been explored

Fuzzing Graph



Sulley Recap



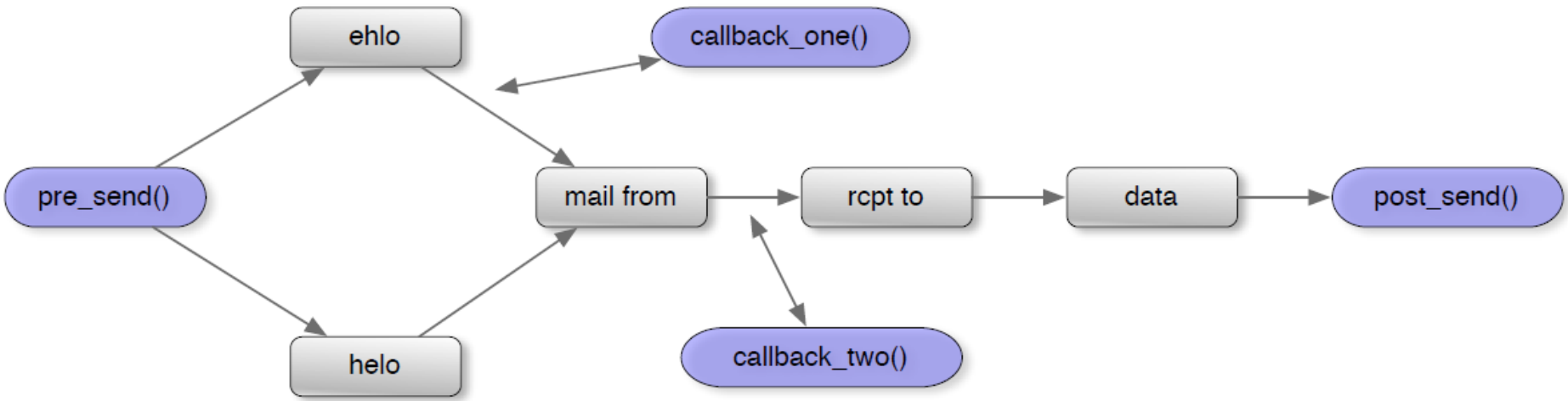
Sulley Session

- Links Sulley requests together in a graph
- Session.connect function is used for linking requests
- Session graphs maybe dumped in uDraw format to be visually rendered
- Several different options may be specified when instantiating a session. Some important ones:
  - session_filename: (string, default=None) Filename to serialize persistent data to. Specifying a filename allows you to stop and resume the fuzzer.
  - sleep_time: (float, default=1.0) Time to sleep in between transmission of test cases.
  - proto: (string, default="tcp") Communication protocol.
  - timeout: (float, default=5.0) Seconds to wait for a send() / recv() to return prior to timing out.
  - crash_threshold: (integer, default=3) Maximum number of crashes allowed before a node is exhausted

Sulley Session Callbacks

- Ability to register callbacks on every edge defined within the protocol graph structure
- Allows us to register a function to call between node transmissions to implement functionality such as challenge response systems
- def callback(node, edge, last_recv, sock)
  - 'node' is the node about to be sent
  - 'edge' is the last edge along the current fuzz path to 'node'
  - 'last_recv' contains the data returned from the last socket transmission
  - 'sock' is the live socket.

Where you can register callbacks



Resources

1. Fuzzing.org: `www.fuzzing.org/`
2. Sulley: `github.com/OpenRCE/sulley`
3. Sulley User Manual: `www.fuzzing.org/wp-content/SulleyManual.pdf`