StateFuzzVis:

State-coverage Guided Fuzzer Visualizer

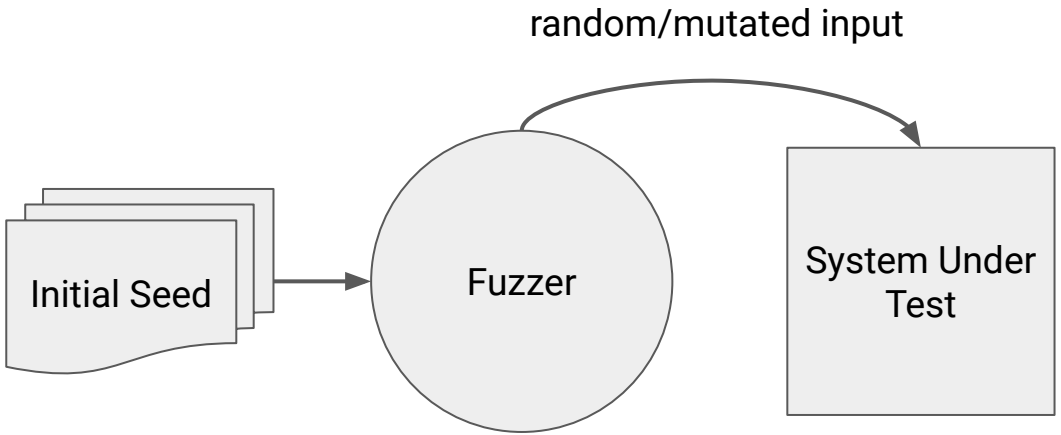
Steve Gustaman
stevegustaman@kaist.ac.kr

Motivation

- **Fuzzing** is an **effective** and **widely-used** vulnerability detection technique

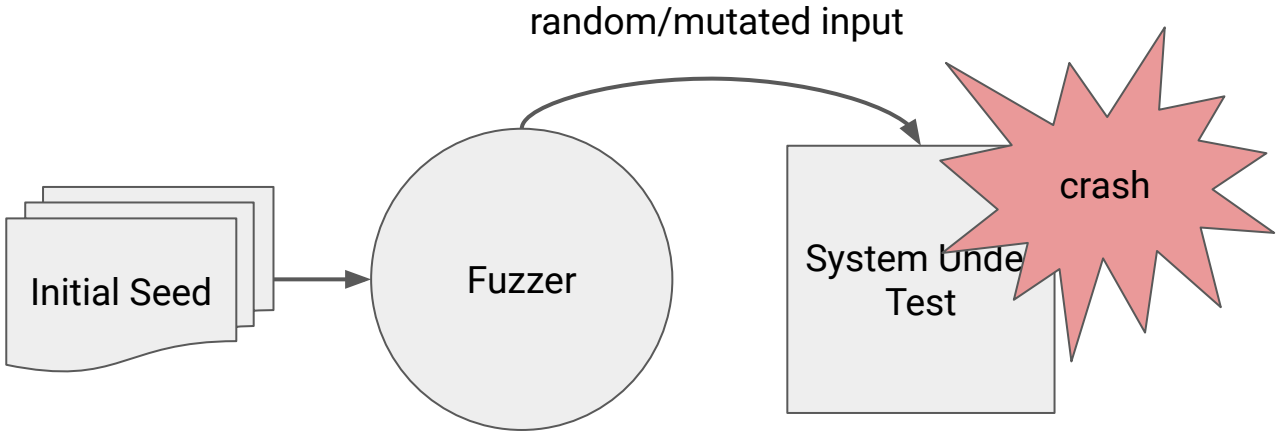
Motivation

- **Fuzzing** is an **effective** and **widely-used** vulnerability detection technique



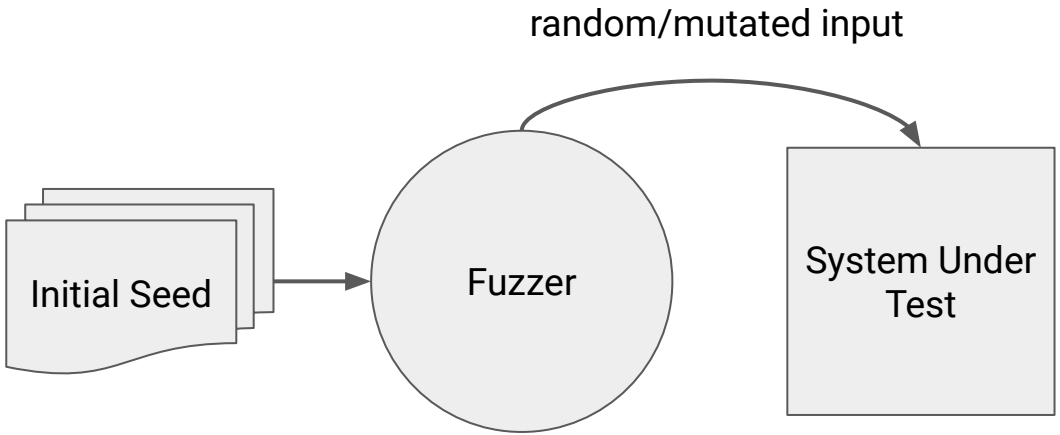
Motivation

- **Fuzzing** is an **effective** and **widely-used** vulnerability detection technique



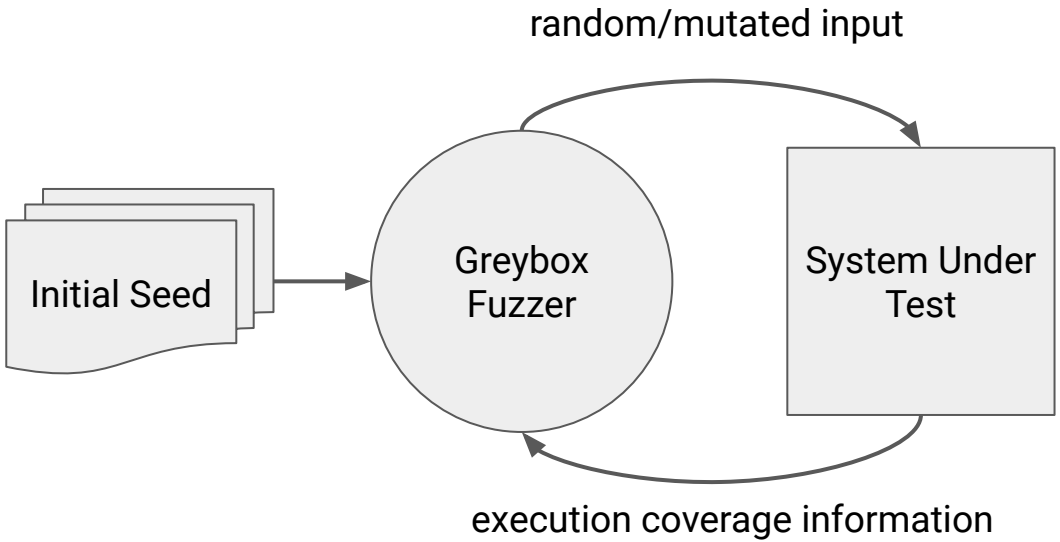
Motivation

- **Fuzzing** is an **effective** and **widely-used** vulnerability detection technique
- Greybox fuzzing uses **coverage information** to guide its fuzzing process



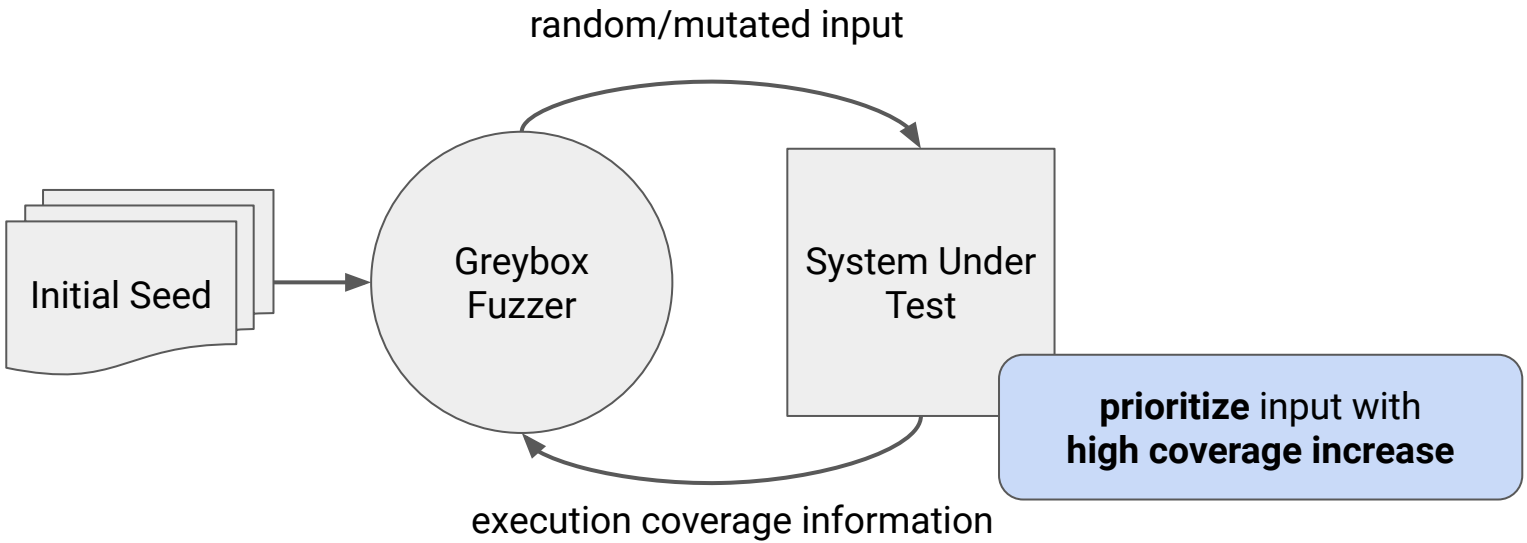
Motivation

- **Fuzzing** is an **effective** and **widely-used** vulnerability detection technique
- Greybox fuzzing uses **coverage information** to guide its fuzzing process



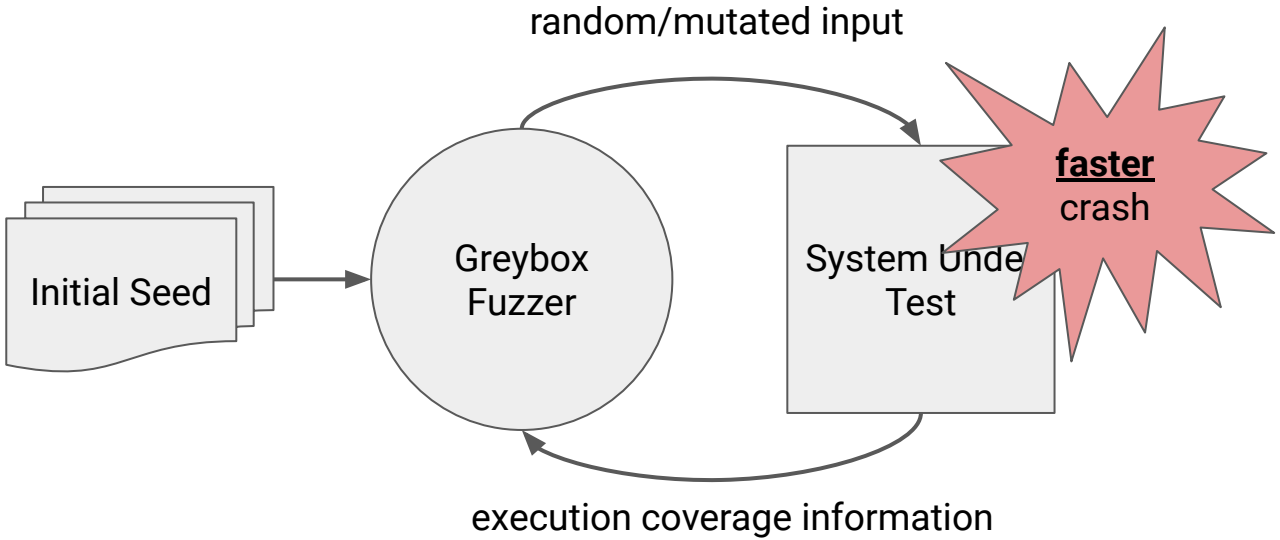
Motivation

- **Fuzzing** is an **effective** and **widely-used** vulnerability detection technique
- Greybox fuzzing uses **coverage information** to guide its fuzzing process



Motivation

- **Fuzzing** is an **effective** and **widely-used** vulnerability detection technique
- Greybox fuzzing uses **coverage information** to guide its fuzzing process



Motivation

- **Fuzzing** is an **effective** and **widely-used** vulnerability detection technique
- Greybox fuzzing uses **coverage information** to guide its fuzzing process
- Fuzzing in stateful environment such as network protocol is challenging

Motivation

- **Fuzzing** is an **effective** and **widely-used** vulnerability detection technique
- Greybox fuzzing uses **coverage information** to guide its fuzzing process
- Fuzzing in stateful environment such as network protocol is challenging

```
220 LightFTP server v2.0a ready
USER foo
331 User foo OK. Password required
PASS foo
230 User logged in, proceed.
MKD demo
257 Directory created.
CWD demo
250 Requested file action okay, completed.
STOR test.txt
150 File status okay
226 Transfer complete
QUIT
221 Goodbye!
```

Motivation

- **Fuzzing** is an **effective** and **widely-used** vulnerability detection technique
- Greybox fuzzing uses **coverage information** to guide its fuzzing process
- Fuzzing in stateful environment such as network protocol is challenging
  - Input is a **sequence of messages**

```
220 LightFTP server v2.0a ready
USER foo
331 User foo OK. Password required
PASS foo
230 User logged in, proceed.
MKD demo
257 Directory created.
CWD demo
250 Requested file action okay, completed.
STOR test.txt
150 File status okay
226 Transfer complete
QUIT
221 Goodbye!
```

Motivation

- **Fuzzing** is an **effective** and **widely-used** vulnerability detection technique
- Greybox fuzzing uses **coverage information** to guide its fuzzing process
- Fuzzing in stateful environment such as network protocol is challenging
  - Input is a **sequence of messages**

```
220 LightFTP server v2.0a ready
USER foo
331 User foo OK. Password required
PASS foo
230 User logged in, proceed.
MKD demo
257 Directory created.
CWD demo
250 Requested file action okay, completed.
STOR test.txt
150 File status okay
226 Transfer complete
QUIT
221 Goodbye!
```

Motivation

- **Fuzzing** is an **effective** and **widely-used** vulnerability detection technique
- Greybox fuzzing uses **coverage information** to guide its fuzzing process
- Fuzzing in stateful environment such as network protocol is challenging
  - Input is a **sequence of messages**
  - Server behavior depends on **current program state**

```
220 LightFTP server v2.0a ready
USER foo
331 User foo OK. Password required
PASS foo
230 User logged in, proceed.
MKD demo
257 Directory created.
CWD demo
250 Requested file action okay, completed.
STOR test.txt
150 File status okay
226 Transfer complete
QUIT
221 Goodbye!
```

Motivation

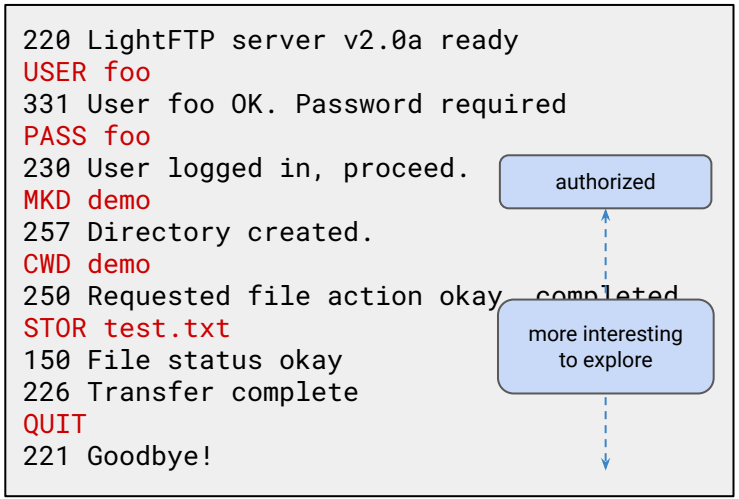
- **Fuzzing** is an **effective** and **widely-used** vulnerability detection technique
- Greybox fuzzing uses **coverage information** to guide its fuzzing process
- Fuzzing in stateful environment such as network protocol is challenging
  - Input is a **sequence of messages**
  - Server behavior depends on **current program state**

```
220 LightFTP server v2.0a ready
USER foo
331 User foo OK. Password required
PASS foo
230 User logged in, proceed.
MKD demo
257 Directory created.
CWD demo
250 Requested file action okay, completed.
STOR test.txt
150 File status okay
226 Transfer complete
QUIT
221 Goodbye!
```

authorized

Motivation

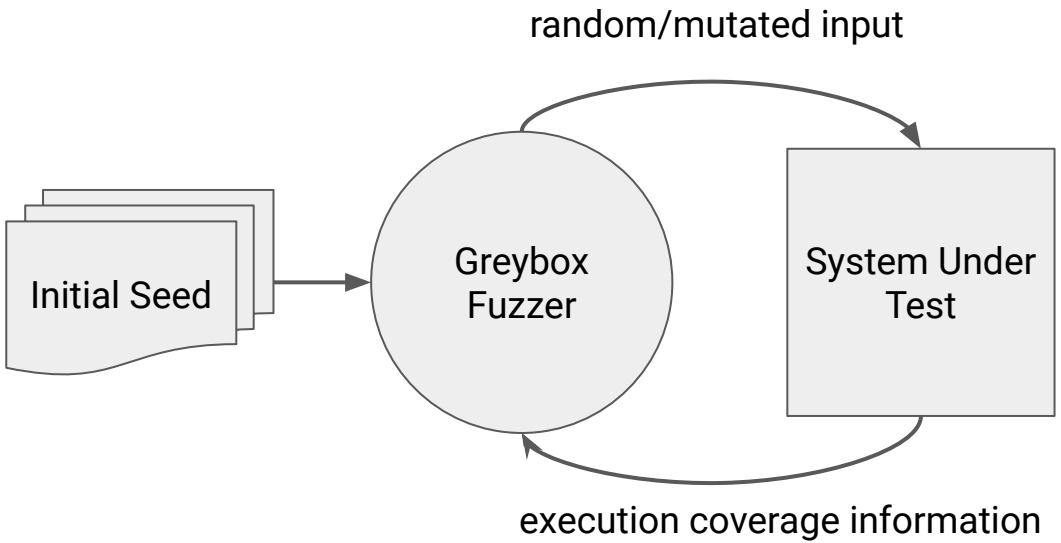
- **Fuzzing** is an **effective** and **widely-used** vulnerability detection technique
- Greybox fuzzing uses **coverage information** to guide its fuzzing process
- Fuzzing in stateful environment such as network protocol is challenging
  - Input is a **sequence of messages**
  - Server behavior depends on **current program state**



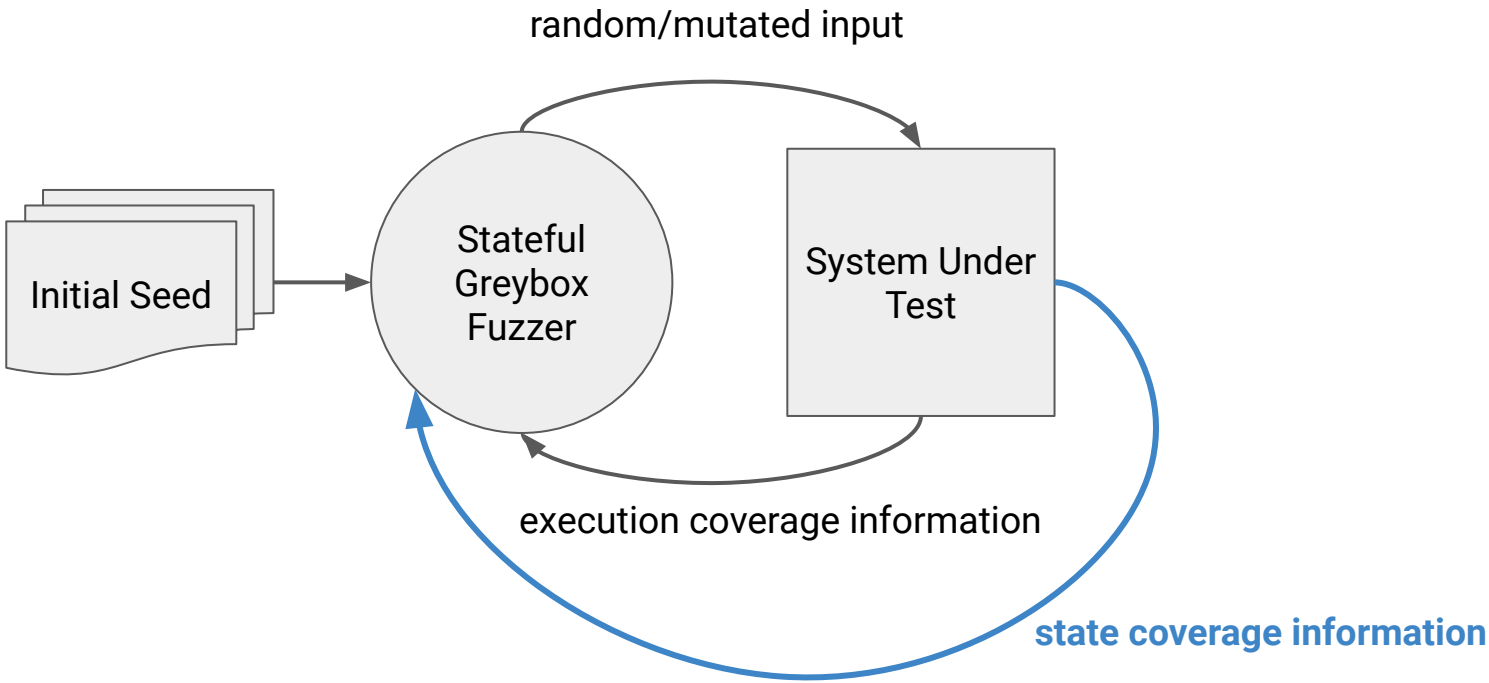
Motivation

- **Fuzzing** is an **effective** and **widely-used** vulnerability detection technique
- Greybox fuzzing uses **coverage information** to guide its fuzzing process
- Fuzzing in stateful environment such as network protocol is challenging
  - Input is a **sequence of messages**
  - Server behavior depends on **current program state**
- Much research effort on **state-coverage guided fuzzers**

State-coverage Guided Fuzzers



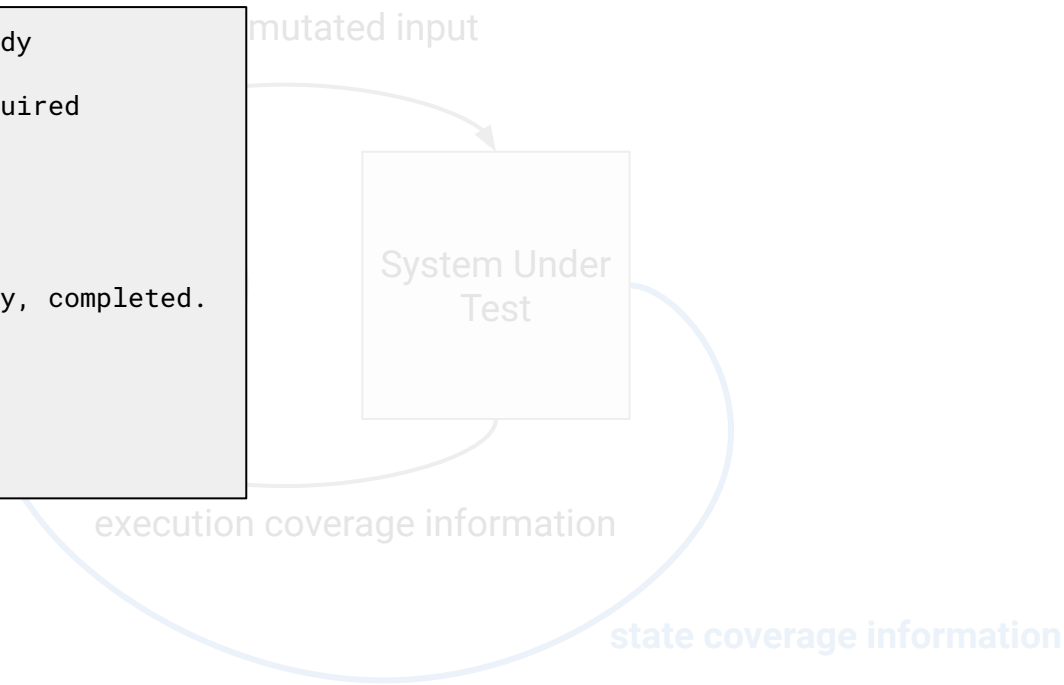
State-coverage Guided Fuzzers



State-coverage Guided Fuzzers

State identification

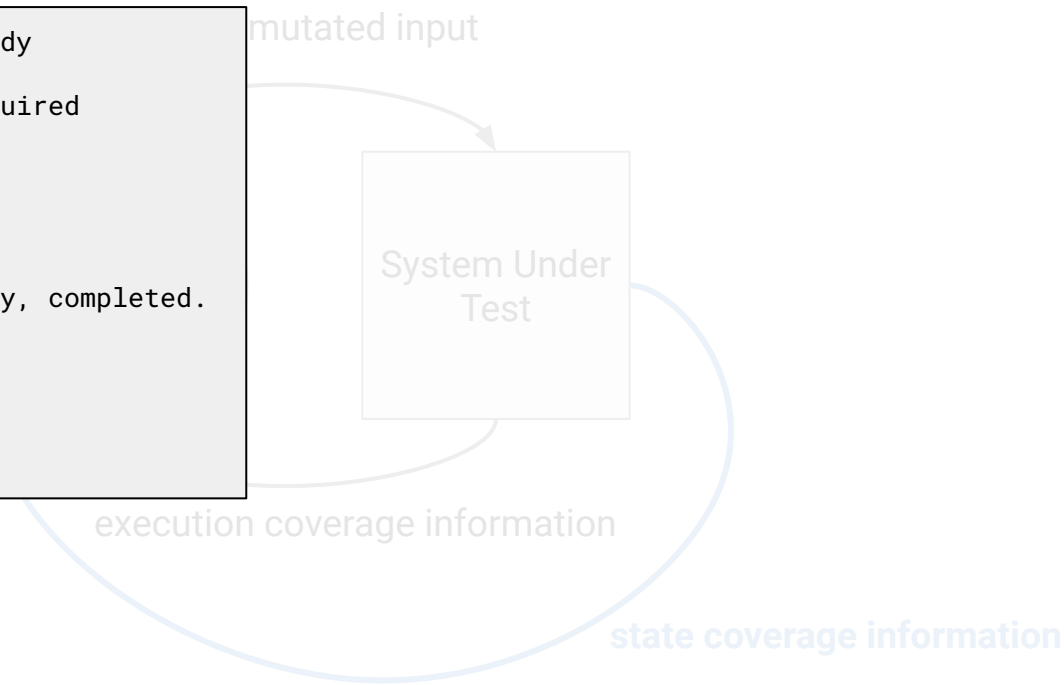
```
220 LightFTP server v2.0a ready
USER foo
331 User foo OK. Password required
PASS foo
230 User logged in, proceed.
MKD demo
257 Directory created.
CWD demo
250 Requested file action okay, completed.
STOR test.txt
150 File status okay
226 Transfer complete
QUIT
221 Goodbye!
```



State-coverage Guided Fuzzers

State identification

```
220 LightFTP server v2.0a ready
USER foo
331 User foo OK. Password required
PASS foo
230 User logged in, proceed.
MKD demo
257 Directory created.
CWD demo
250 Requested file action okay, completed.
STOR test.txt
150 File status okay
226 Transfer complete
QUIT
221 Goodbye!
```



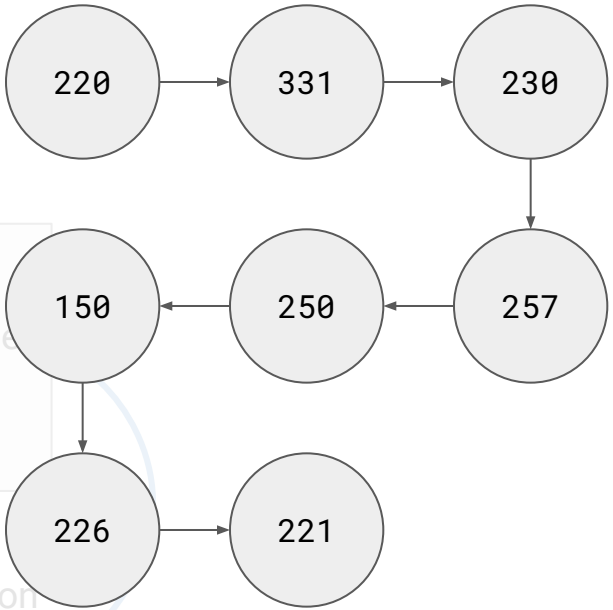
State-coverage Guided Fuzzers

State identification

```
220 LightFTP server v2.0a ready
USER foo
331 User foo OK. Password required
PASS foo
230 User logged in, proceed.
MKD demo
257 Directory created.
CWD demo
250 Requested file action okay, completed.
STOR test.txt
150 File status okay
226 Transfer complete
QUIT
221 Goodbye!
```

AFLNet

State coverage information

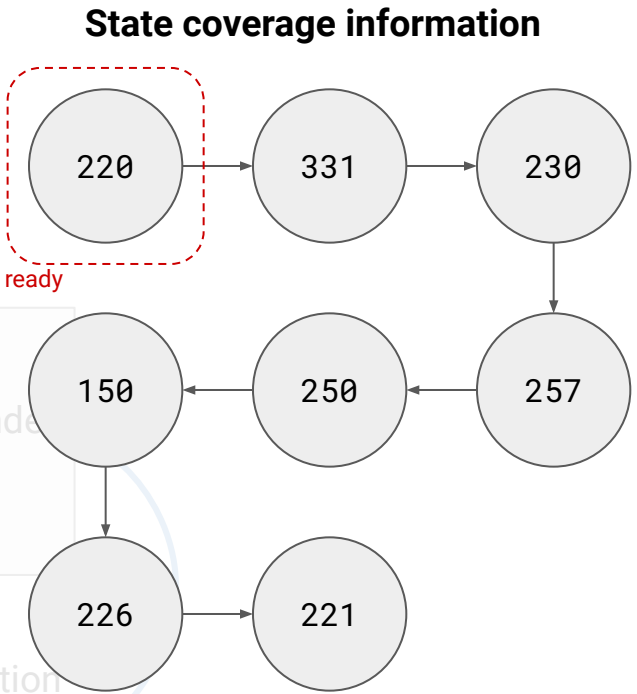


State-coverage Guided Fuzzers

State identification

```
220 LightFTP server v2.0a ready
USER foo
331 User foo OK. Password required
PASS foo
230 User logged in, proceed.
MKD demo
257 Directory created.
CWD demo
250 Requested file action okay, completed.
STOR test.txt
150 File status okay
226 Transfer complete
QUIT
221 Goodbye!
```

AFLNet

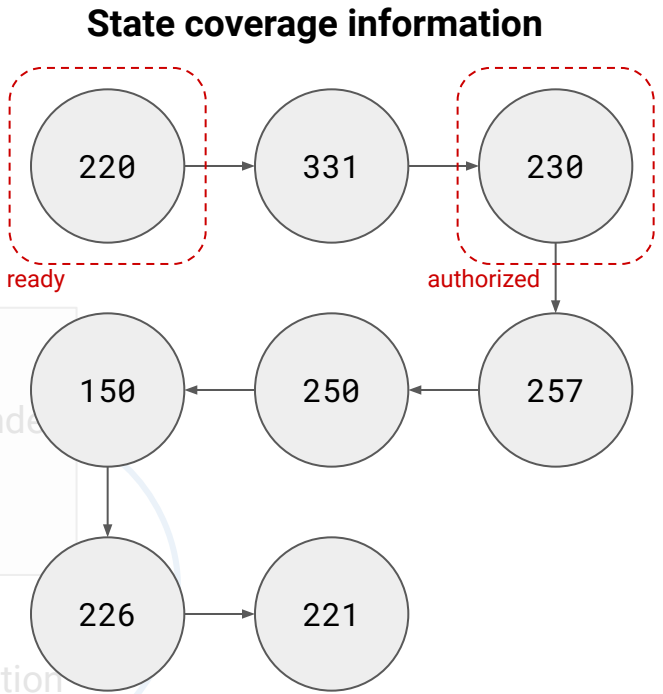


State-coverage Guided Fuzzers

State identification

```
220 LightFTP server v2.0a ready
USER foo
331 User foo OK. Password required
PASS foo
230 User logged in, proceed.
MKD demo
257 Directory created.
CWD demo
250 Requested file action okay, completed.
STOR test.txt
150 File status okay
226 Transfer complete
QUIT
221 Goodbye!
```

AFLNet



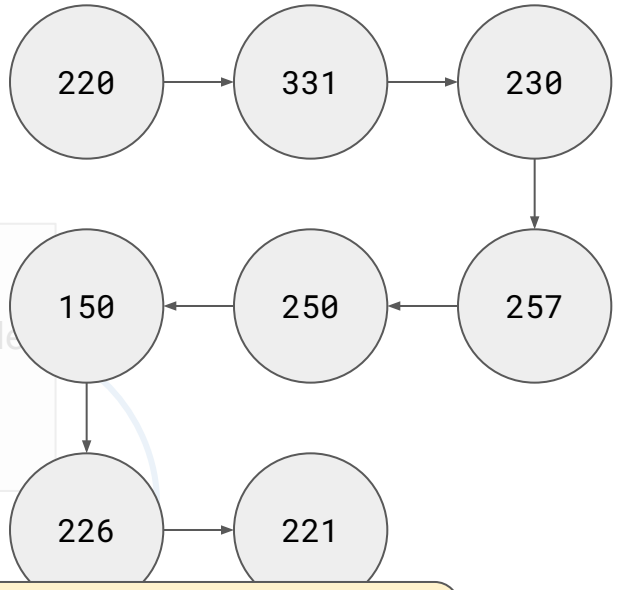
State-coverage Guided Fuzzers

State identification

```
220 LightFTP server v2.0a ready
USER foo
331 User foo OK. Password required
PASS foo
230 User logged in, proceed.
MKD demo
257 Directory created.
CWD demo
250 Requested file action okay, completed.
STOR test.txt
150 File status okay
226 Transfer complete
QUIT
221 Goodbye!
```

AFLNet

State coverage information

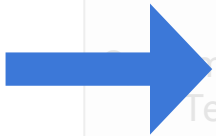


other fuzzers may use different state identification strategy

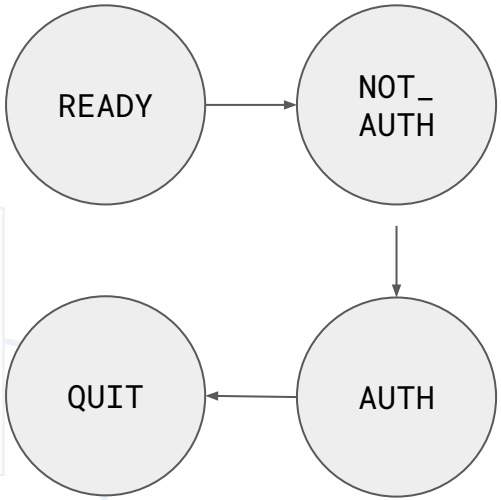
State-coverage Guided Fuzzers

State identification

```
220 LightFTP server v2.0a ready
USER foo
331 User foo OK. Password required
PASS foo
230 User logged in, proceed.
MKD demo
257 Directory created.
CWD demo
250 Requested file action okay, completed.
STOR test.txt
150 File status okay
226 Transfer complete
QUIT
221 Goodbye!
```



State coverage information



other fuzzers may use different state identification strategy

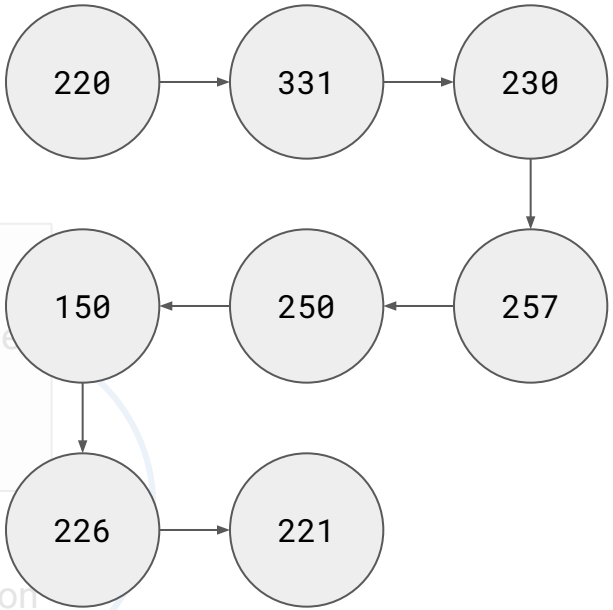
State-coverage Guided Fuzzers

State identification

```
220 LightFTP server v2.0a ready
USER foo
331 User foo OK. Password required
PASS foo
230 User logged in, proceed.
MKD demo
257 Directory created.
CWD demo
250 Requested file action okay, completed.
STOR test.txt
150 File status okay
226 Transfer complete
QUIT
221 Goodbye!
```

AFLNet

State coverage information



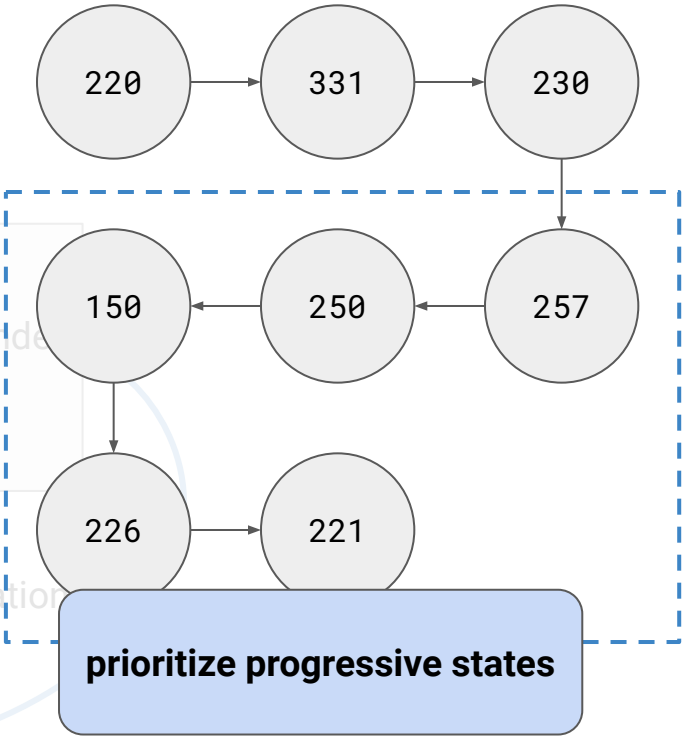
State-coverage Guided Fuzzers

State identification

```
220 LightFTP server v2.0a ready
USER foo
331 User foo OK. Password required
PASS foo
230 User logged in, proceed.
MKD demo
257 Directory created.
CWD demo
250 Requested file action okay, completed.
STOR test.txt
150 File status okay
226 Transfer complete
QUIT
221 Goodbye!
```

AFLNet

State coverage information



State-coverage Guided Fuzzers

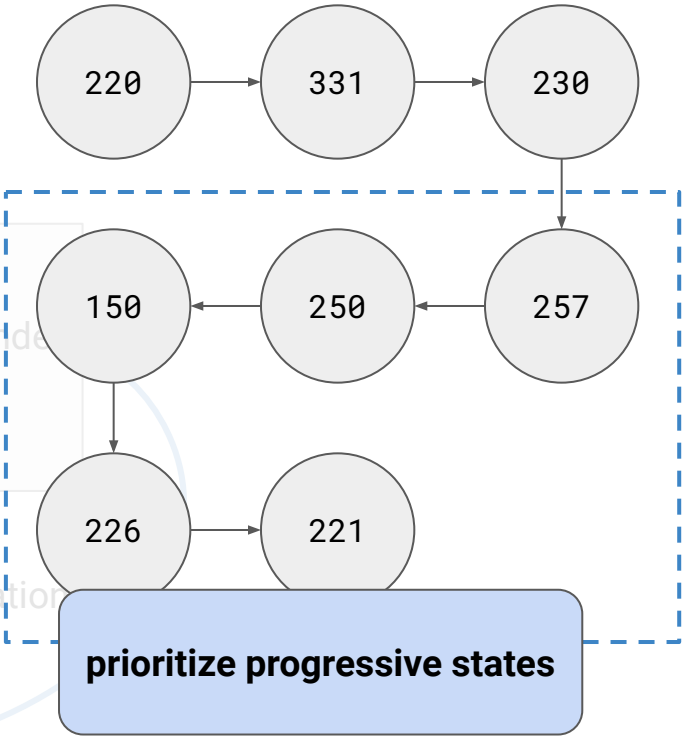
State identification

```
220 LightFTP server v2.0a ready
USER foo
331 User foo OK. Password required
PASS foo
230 User logged in, proceed.
MKD demo
257 Directory created.
CWD demo
250 Requested file action okay, completed.
STOR test.txt
150 File status okay
226 Transfer compl
QUIT
221 Goodbye!
```

AFLNet

other fuzzers may use different prioritization strategy

State coverage information



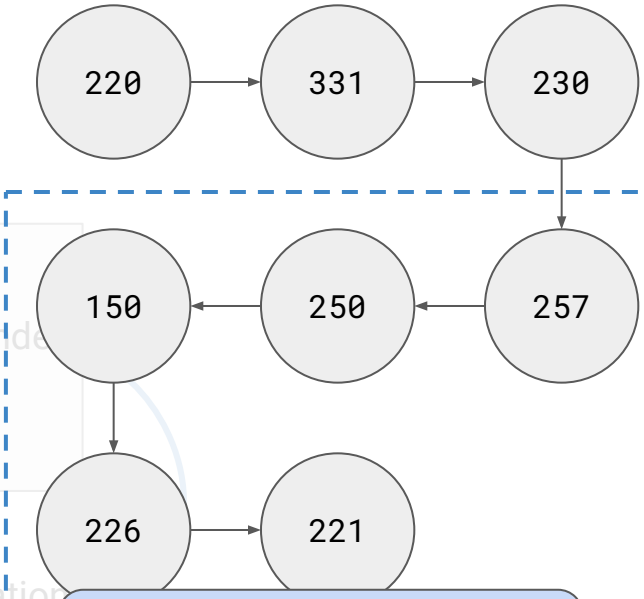
State-coverage Guided Fuzzers

State identification

```
220 LightFTP server v2.0a ready
USER foo
331 User foo OK. Password required
PASS foo
230 User logged in, proceed.
MKD demo
257 Directory created.
CWD demo
250 Requested file action okay, completed.
STOR test.txt
150 File status okay
226 Transfer compl
QUIT
221 Goodbye!
```

AFLNet

State coverage information



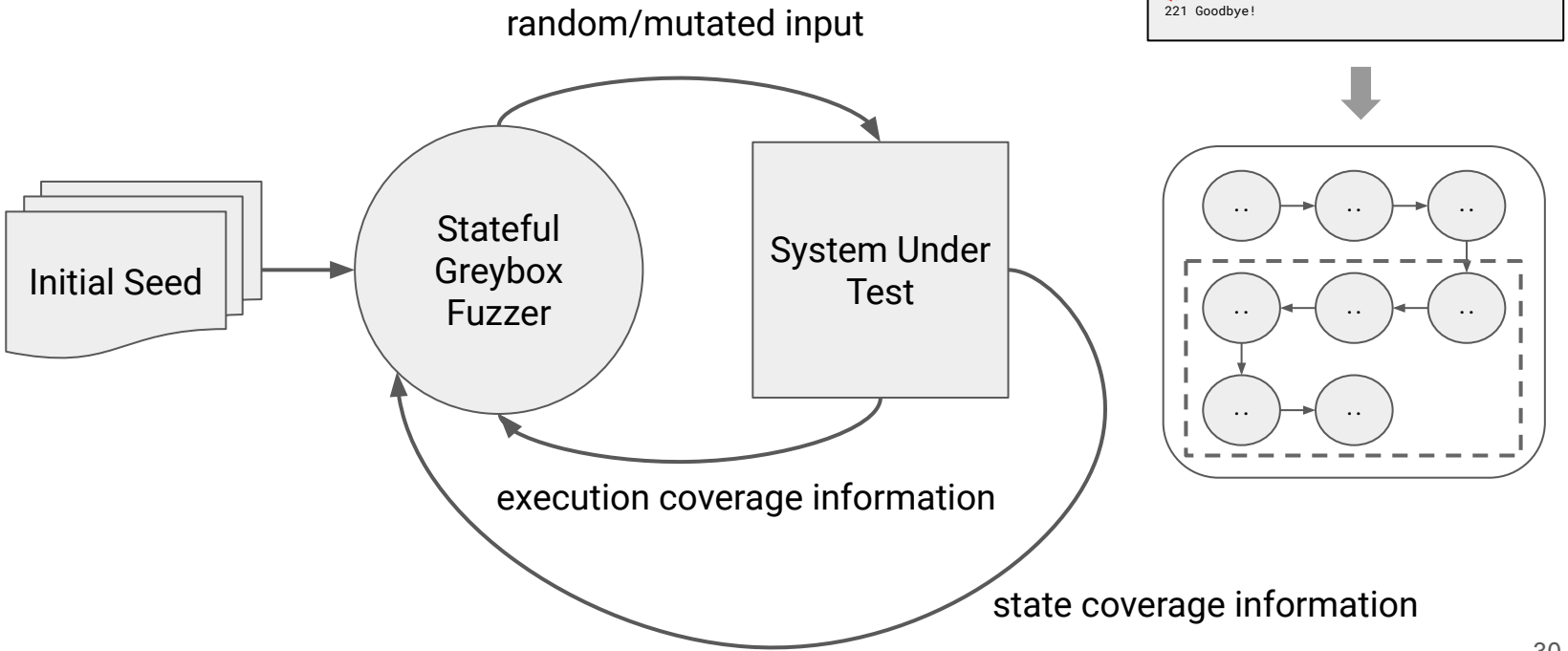
other fuzzers may use different prioritization strategy

state identification strategy also affects decision

prioritize progressive states

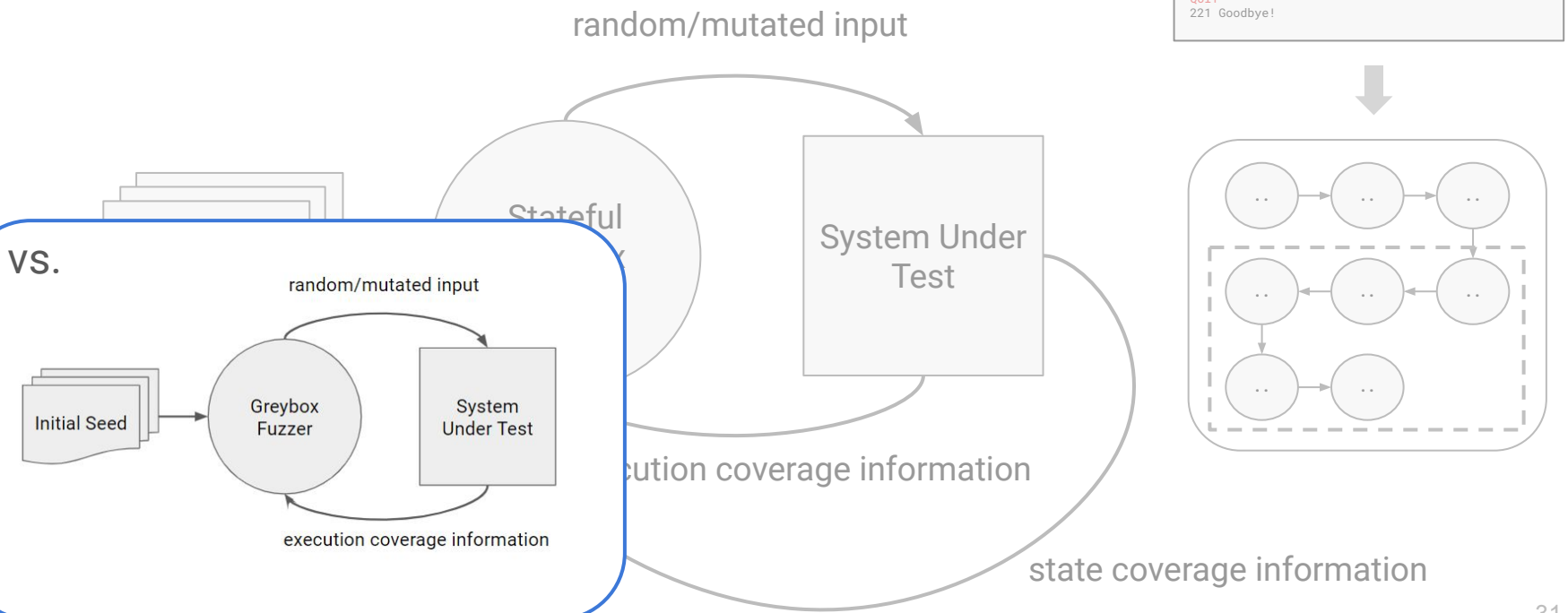
State-coverage Guided Fuzzers

```
220 LightFTP server v2.0a ready
USER foo
331 User foo OK. Password required
PASS foo
230 User logged in, proceed.
MKD demo
257 Directory created.
CWD demo
250 Requested file action okay, completed.
STOR test.txt
150 File status okay
226 Transfer complete
QUIT
221 Goodbye!
```



State-coverage Guided Fuzzers

```
220 LightFTP server v2.0a ready
USER foo
331 User foo OK. Password required
PASS foo
230 User logged in, proceed.
MKD demo
257 Directory created.
CWD demo
250 Requested file action okay, completed.
STOR test.txt
150 File status okay
226 Transfer complete
QUIT
221 Goodbye!
```

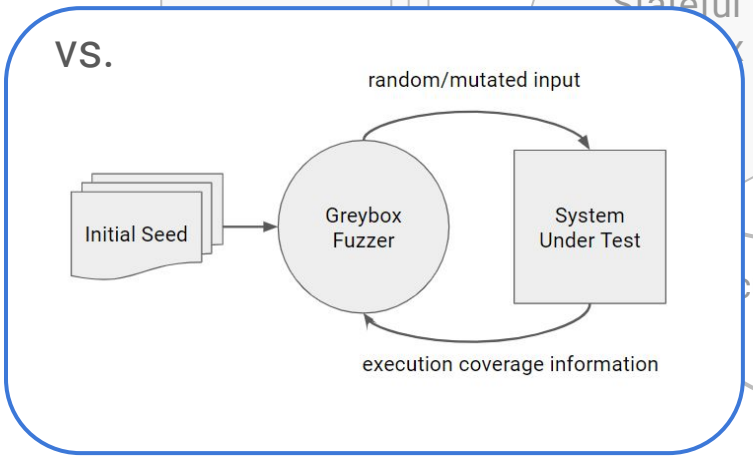


State-coverage Guided Fuzzers

```
220 LightFTP server v2.0a ready
USER foo
331 User foo OK. Password required
PASS foo
230 User logged in, proceed.
MKD demo
257 Directory created.
```

```
american fuzzy lop 1.86b (test)

process timing                    overall results
       run time : 0 days, 0 hrs, 0 min, 2 sec      cycles done : 0
  last new path : none seen yet                    total paths : 1
 last uniq crash : 0 days, 0 hrs, 0 min, 2 sec     uniq crashes : 1
  last uniq hang : none seen yet                    uniq hangs : 0
cycle progress                    map coverage
  now processing : 0 (0.00%)          map density : 2 (0.00%)
 paths timed out : 0 (0.00%)       count coverage : 1.00 bits/tuple
stage progress                    findings in depth
  now trying : havoc              favored paths : 1 (100.00%)
 stage execs : 1464/5000 (29.28%)  new edges on : 1 (100.00%)
 total execs : 1697               total crashes : 39 (1 unique)
  exec speed : 626.5/sec            total hangs : 0 (0 unique)
fuzzing strategy yields           path geometry
   bit flips : 0/16, 1/15, 0/13       levels : 1
   byte flips : 0/2, 0/1, 0/0         pending : 1
  arithmetics : 0/112, 0/25, 0/0     pend fav : 1
   known ints : 0/10, 0/28, 0/0     own finds : 0
   dictionary : 0/0, 0/0, 0/0        imported : n/a
        havoc : 0/0, 0/0             variable : 0
         trim : n/a, 0.00%
                                                  [cpu: 92%]
```

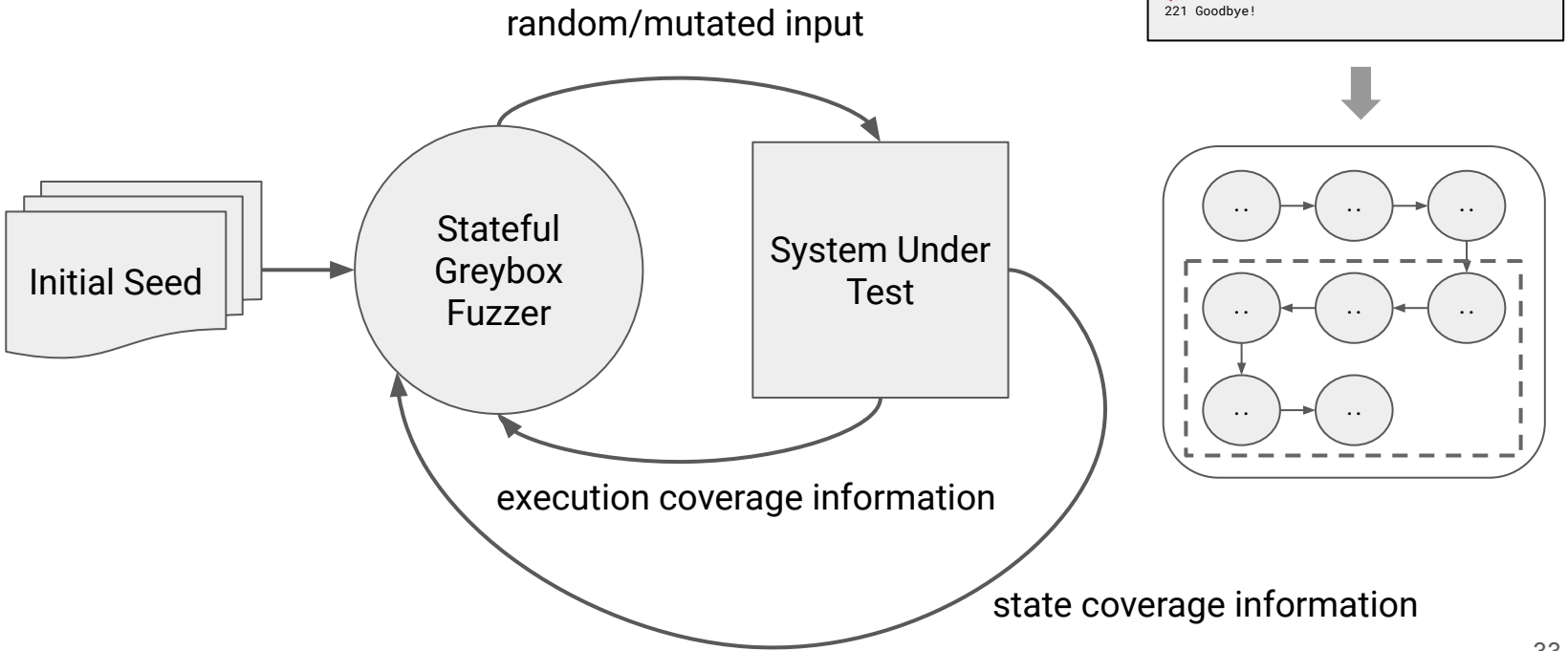


state coverage information

https://en.wikipedia.org/wiki/American_Fuzzy_Lop_%28software%29#/media/File:American_fuzzy_lop's_afl-fuzz_running_on_a_test_program.png

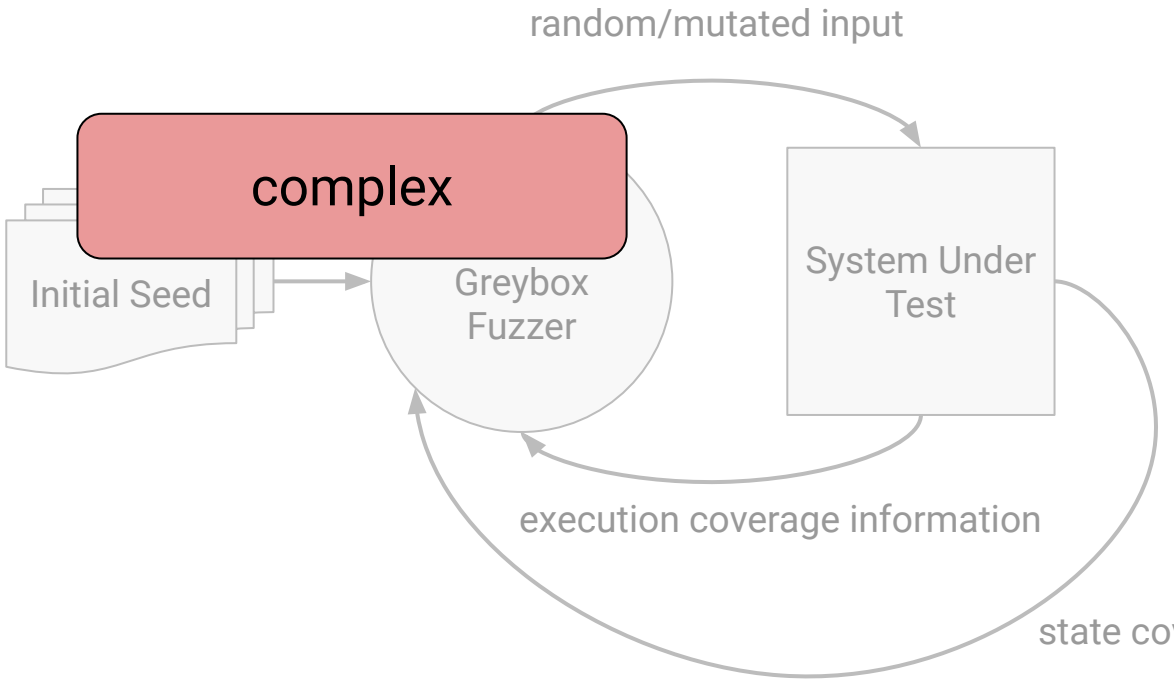
State-coverage Guided Fuzzers

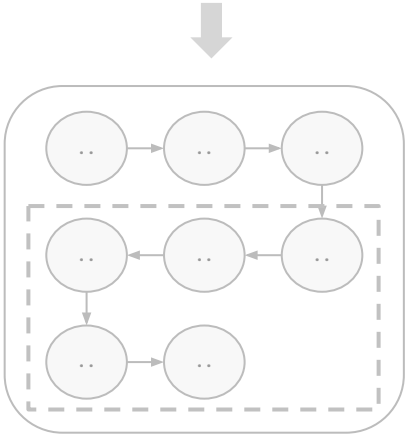
```
220 LightFTP server v2.0a ready
USER foo
331 User foo OK. Password required
PASS foo
230 User logged in, proceed.
MKD demo
257 Directory created.
CWD demo
250 Requested file action okay, completed.
STOR test.txt
150 File status okay
226 Transfer complete
QUIT
221 Goodbye!
```



State-coverage Guided Fuzzers

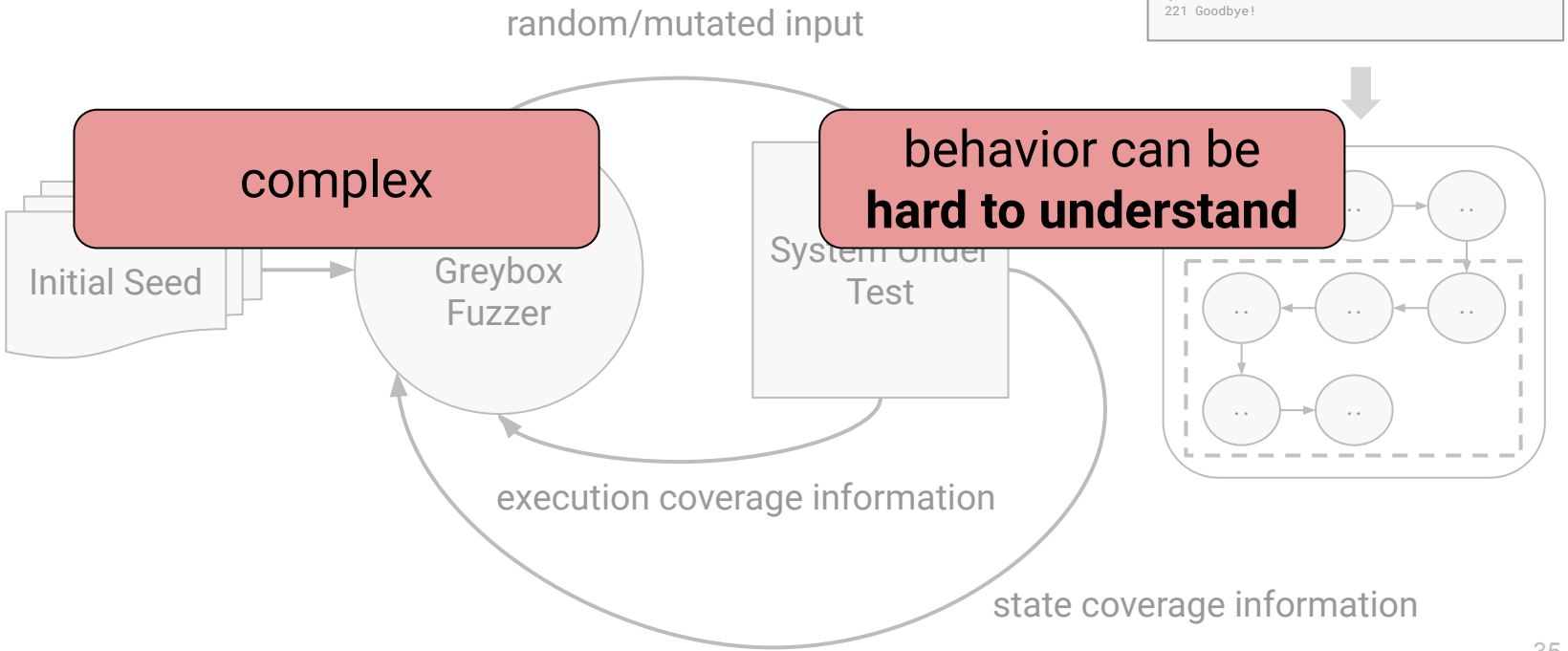
```
220 LightFTP server v2.0a ready
USER foo
331 User foo OK. Password required
PASS foo
230 User logged in, proceed.
MKD demo
257 Directory created.
CWD demo
250 Requested file action okay, completed.
STOR test.txt
150 File status okay
226 Transfer complete
QUIT
221 Goodbye!
```





State-coverage Guided Fuzzers

```
220 LightFTP server v2.0a ready
USER foo
331 User foo OK. Password required
PASS foo
230 User logged in, proceed.
MKD demo
257 Directory created.
CWD demo
250 Requested file action okay, completed.
STOR test.txt
150 File status okay
226 Transfer complete
QUIT
221 Goodbye!
```



StateFuzzVis

- **Visualizer** for **state-coverage guided fuzzers**

StateFuzzVis

- **Visualizer** for **state-coverage guided fuzzers**
  - Visualize the **states** and **state transitions exercised** during fuzzing

StateFuzzVis

- **Visualizer** for **state-coverage guided fuzzers**
  - Visualize the **states** and **state transitions exercised** during fuzzing
- Make it easier for developers/testers to **monitor** fuzzing process

StateFuzzVis

- **Visualizer** for **state-coverage guided fuzzers**
  - Visualize the **states** and **state transitions exercised** during fuzzing
- Make it easier for developers/testers to **monitor** fuzzing process
- Aim to **gain insight** on state-coverage guided fuzzers through **real-time observation**

StateFuzzVis

- **Visualizer** for **state-coverage guided fuzzers**
  - Visualize the **states** and **state transitions exercised** during fuzzing
- Make it easier for developers/testers to **monitor** fuzzing process
- Aim to **gain insight** on state-coverage guided fuzzers through **real-time observation**
  - Answer questions:
    - What states are there? How to transition between states?
    - Given a state transition (e.g. `INIT->AUTH->QUIT`) which input exercises it?

StateFuzzVis

- **Visualizer** for **state-coverage guided fuzzers**
  - Visualize the **states** and **state transitions exercised** during fuzzing
- Make it easier for developers/testers to **monitor** fuzzing process
- Aim to **gain insight** on state-coverage guided fuzzers through **real-time observation**
  - Answer questions:
    - What states are there? How to transition between states?
    - Given a state transition (e.g. `INIT->AUTH->QUIT`) which input exercises it?
    - Are the defined states “good enough”?
    - Where is the fuzzer stuck?

StateFuzzVis

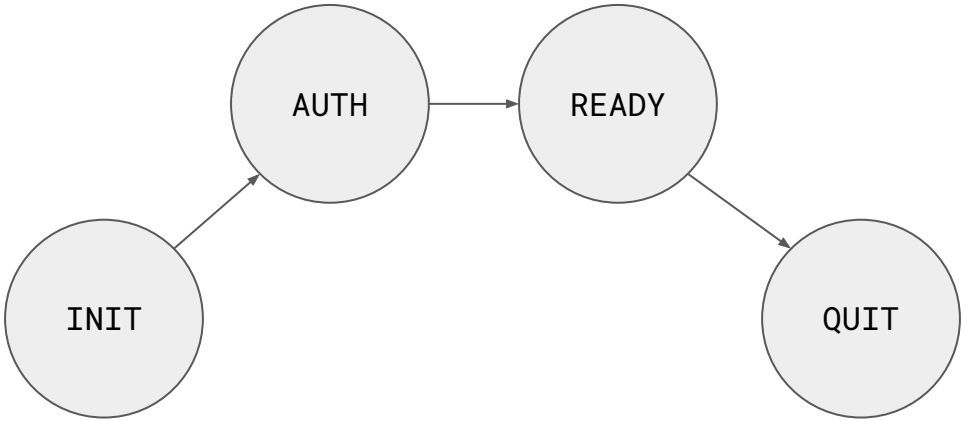
- **Visualizer** for **state-coverage guided fuzzers**
  - Visualize the **states** and **state transitions exercised** during fuzzing
- Make it easier for developers/testers to **monitor** fuzzing process
- Aim to **gain insight** on state-coverage guided fuzzers through **real-time observation**
  - Answer questions:
    - What states are there? How to transition between states?
    - Given a state transition (e.g. `INIT->AUTH->QUIT`) which input exercises it?
    - Are the defined states “good enough”?
    - Where is the fuzzer stuck?
  - Understand the fuzzers’ behavior

Key Ideas: High Level Overview

- Directed graph
- Nodes = States
- Edges = Possible/exercised state transition

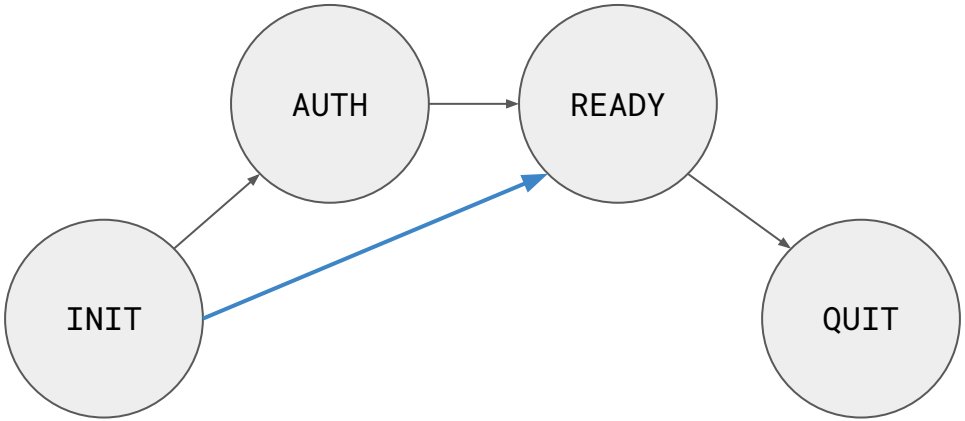
Key Ideas: High Level Overview

- Directed graph
- Nodes = States
- Edges = Possible/exercised state transition



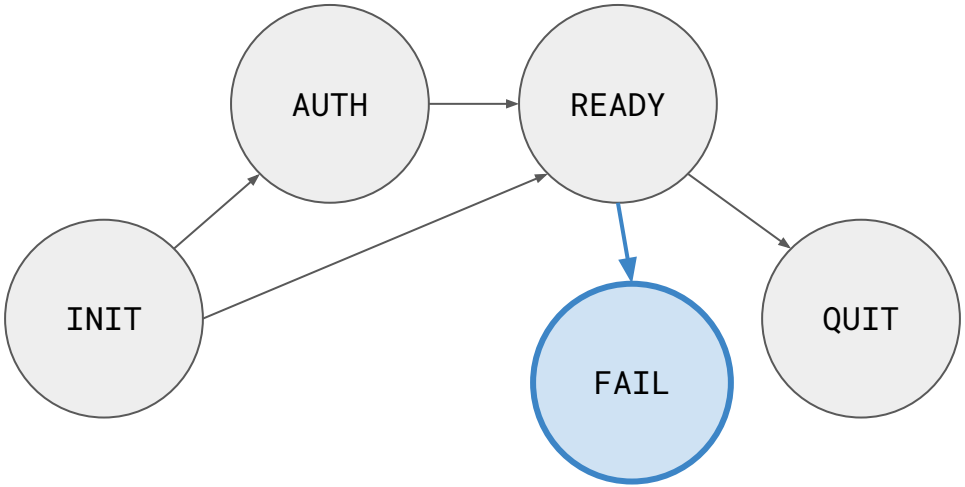
Key Ideas: High Level Overview

- Directed graph
- Nodes = States
- Edges = Possible/exercised state transition



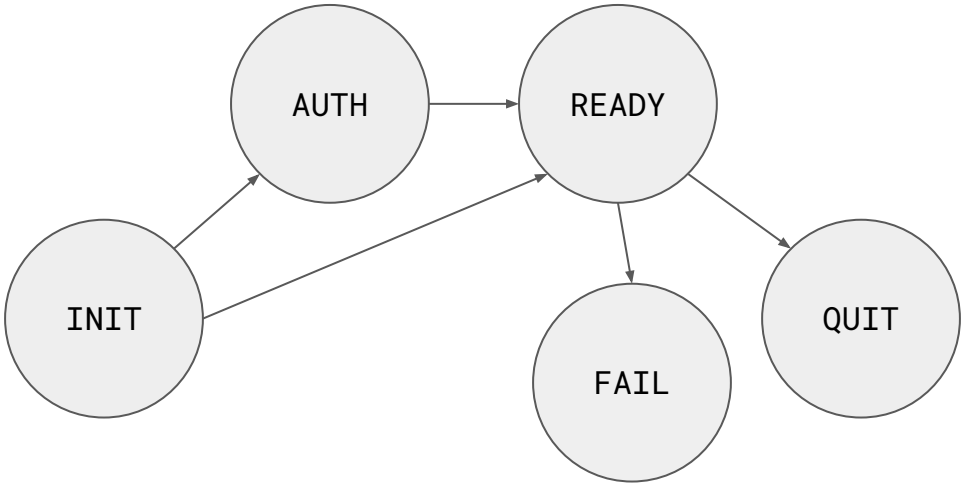
Key Ideas: High Level Overview

- Directed graph
- Nodes = States
- Edges = Possible/exercised state transition



Key Ideas: High Level Overview

- Directed graph
- Nodes = States
- Edges = Possible/exercised state transition



Key Ideas: Detailed Visualization

- Inspect interesting seeds
- Finer granularity inspection

Key Ideas: Detailed Visualization

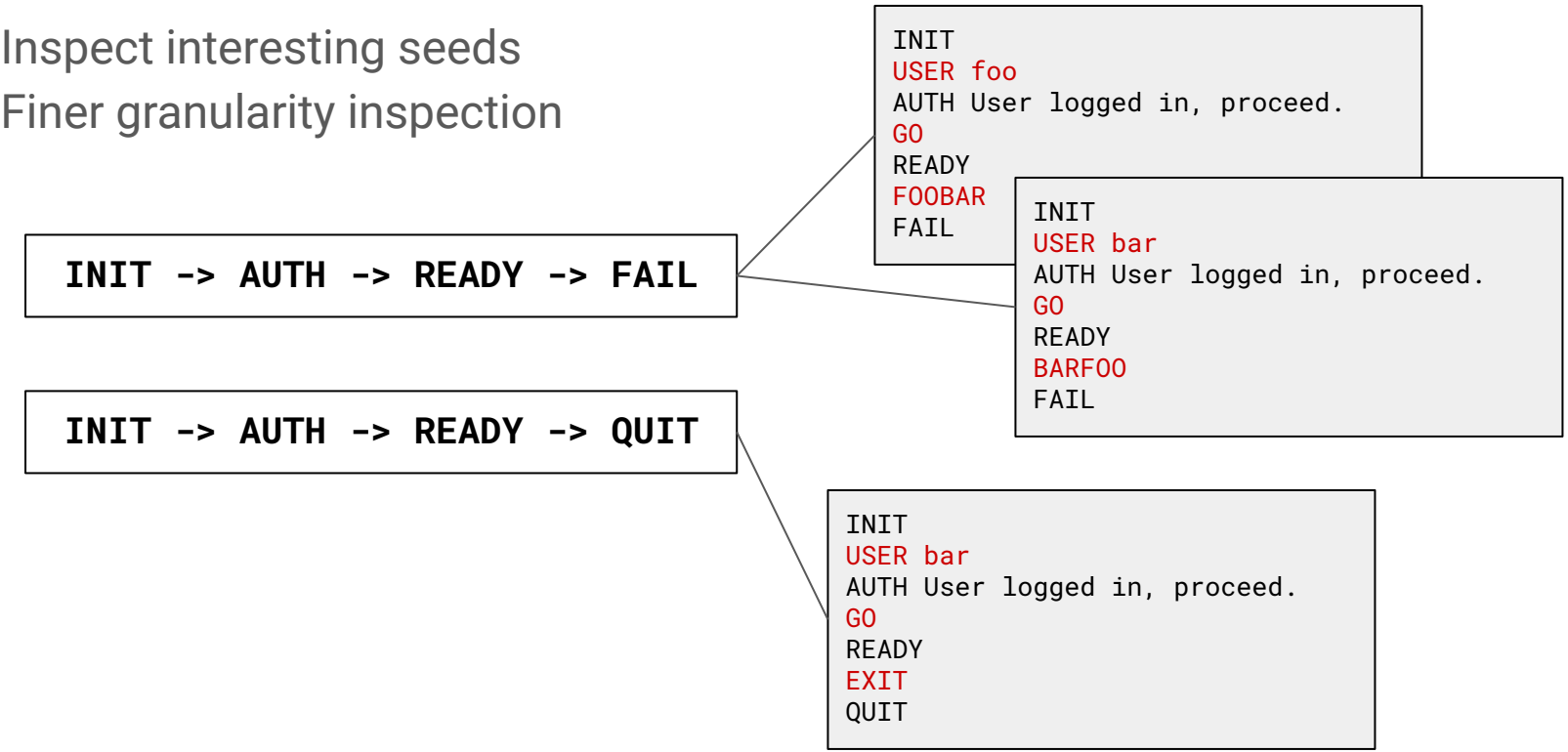
- Inspect interesting seeds
- Finer granularity inspection

```
INIT -> AUTH -> READY -> FAIL
```

```
INIT -> AUTH -> READY -> QUIT
```

Key Ideas: Detailed Visualization

- Inspect interesting seeds
- Finer granularity inspection

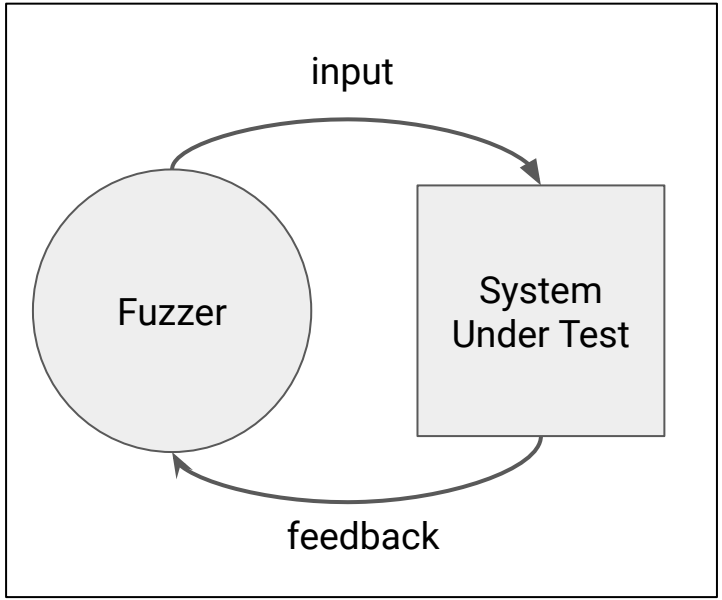


Key Ideas: Technical Details

- Visualization on Web Browser (ease of access)
- Provide API for fuzzer to interact with during fuzzing (real-time monitoring)

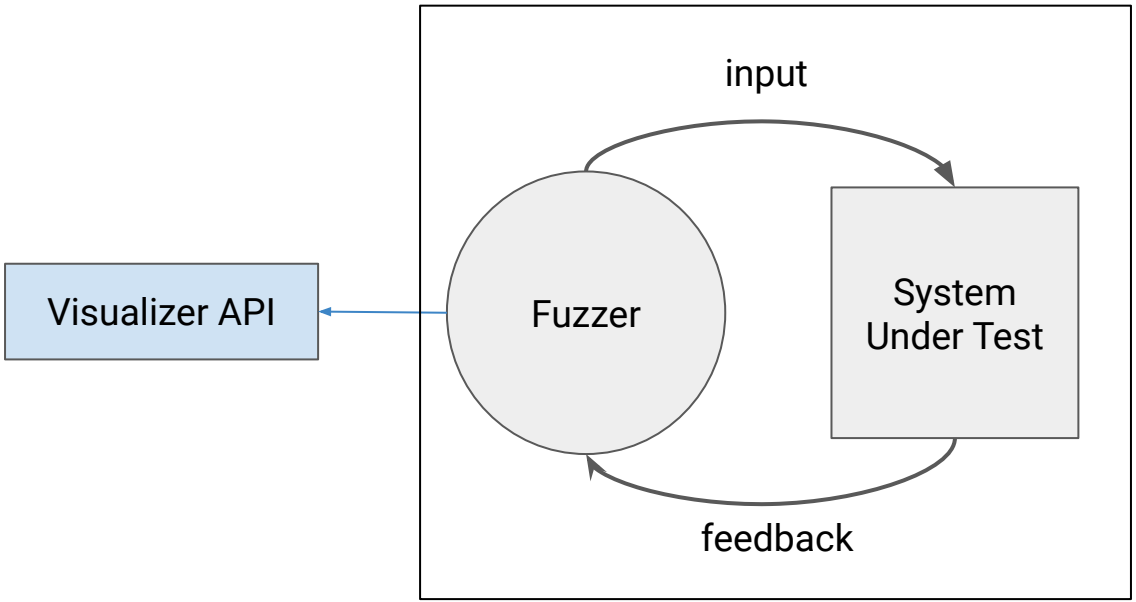
Key Ideas: Technical Details

- Visualization on Web Browser (ease of access)
- Provide API for fuzzer to interact with during fuzzing (real-time monitoring)



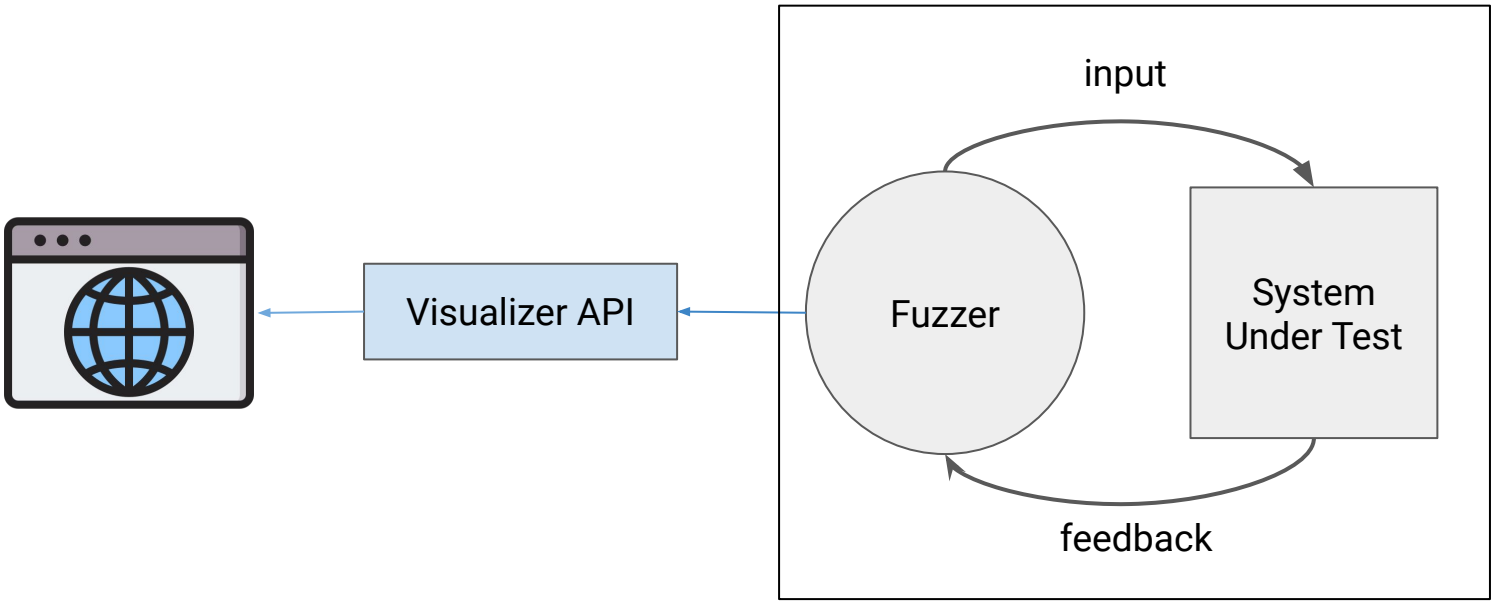
Key Ideas: Technical Details

- Visualization on Web Browser (ease of access)
- Provide API for fuzzer to interact with during fuzzing (real-time monitoring)



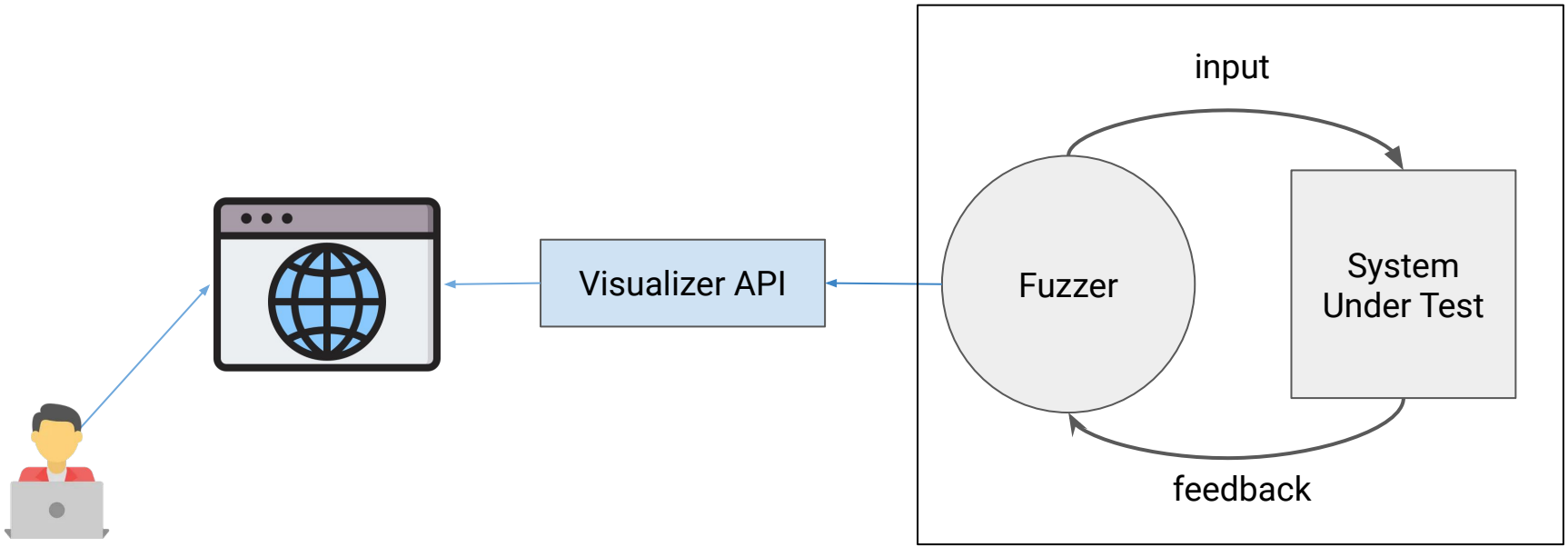
Key Ideas: Technical Details

- Visualization on Web Browser (ease of access)
- Provide API for fuzzer to interact with during fuzzing (real-time monitoring)



Key Ideas: Technical Details

- Visualization on Web Browser (ease of access)
- Provide API for fuzzer to interact with during fuzzing (real-time monitoring)



Evaluation Plan

- Can StateFuzzVis visualize exercised states and transitions?
- Can StateFuzzVis be integrated with existing state-coverage guided fuzzers?

Evaluation Plan

- Can StateFuzzVis visualize exercised states and transitions?
- Can StateFuzzVis be integrated with existing state-coverage guided fuzzers?

- ProFuzzBench for protocol benchmarks
- AFLNet, SGFuzz, StateAFL, etc. for fuzzers

What’s next?

- In-depth analysis of existing state-coverage guided fuzzers
- User-assisted fuzzing

Thank you

Steve Gustaman
stevegustaman@kaist.ac.kr

Real World Scenario

- In its repo tutorial, AFLNet patches target program manually to decompose states for Live555 RTSP Server*

- State **200** is too large

```
SETUP       -> 200 OK
PLAY        -> 200 OK
TEARDOWN    -> 200 OK
```

```
SETUP       -> 201 OK
PLAY        -> 202 OK
TEARDOWN    -> 203 OK
```

*) https://github.com/aflnet/aflnet/blob/master/tutorials/live555/ceeb4f4_states_decomposed.patch