Enemy of the State: A State-Aware Black-Box Web Vulnerability Scanner

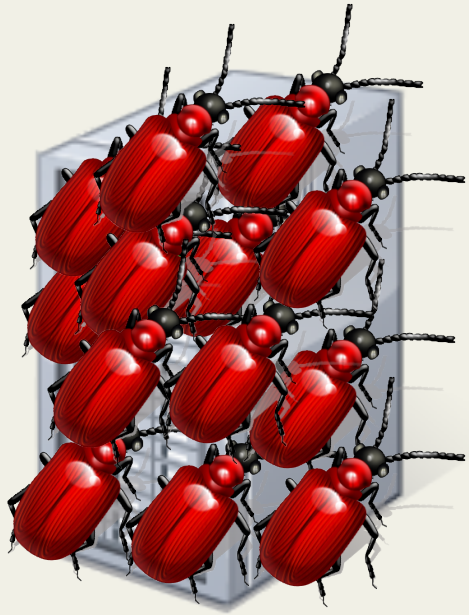
Adam Doupé, Ludovico Cavedon,
Christopher Kruegel, and Giovanni Vigna

University of California, Santa Barbara

USENIX 2012 – 8/10/12

THE COMPUTER SECURITY GROUP AT UCSB

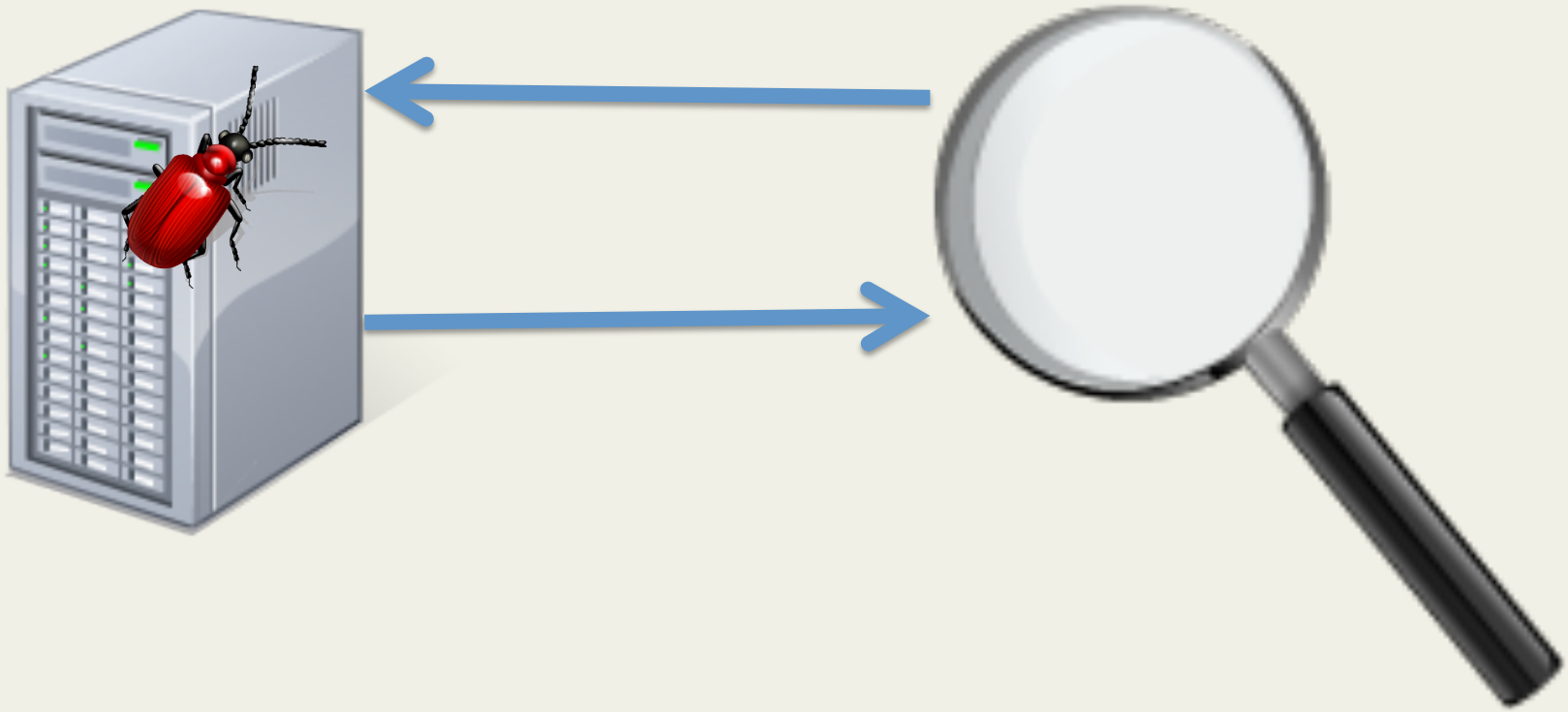
Web Applications Have Bugs



White-Box



Black-Box



seclab
THE COMPUTER SECURITY GROUP AT UCSB

Commercial Tools







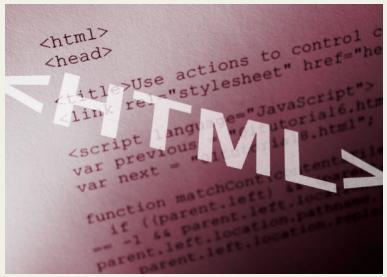






Black-Box Vulnerability Scanners Crawling

GET /index.php

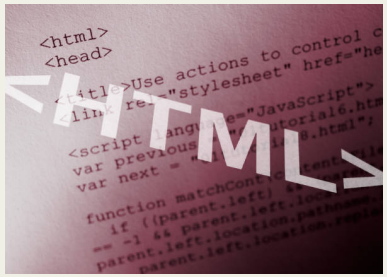


seclab
THE COMPUTER SECURITY GROUP AT UCSB

Black-Box Vulnerability Scanners Crawling

GET /view.php?id=1



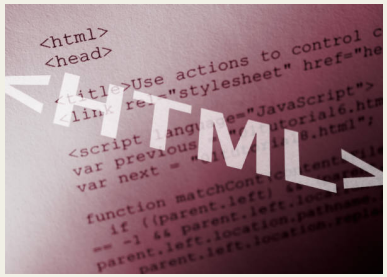


seclab
THE COMPUTER SECURITY GROUP AT UCSB

Black-Box Vulnerability Scanners Fuzzing

GET /view.php?id=
<script>alert(1)</
script>

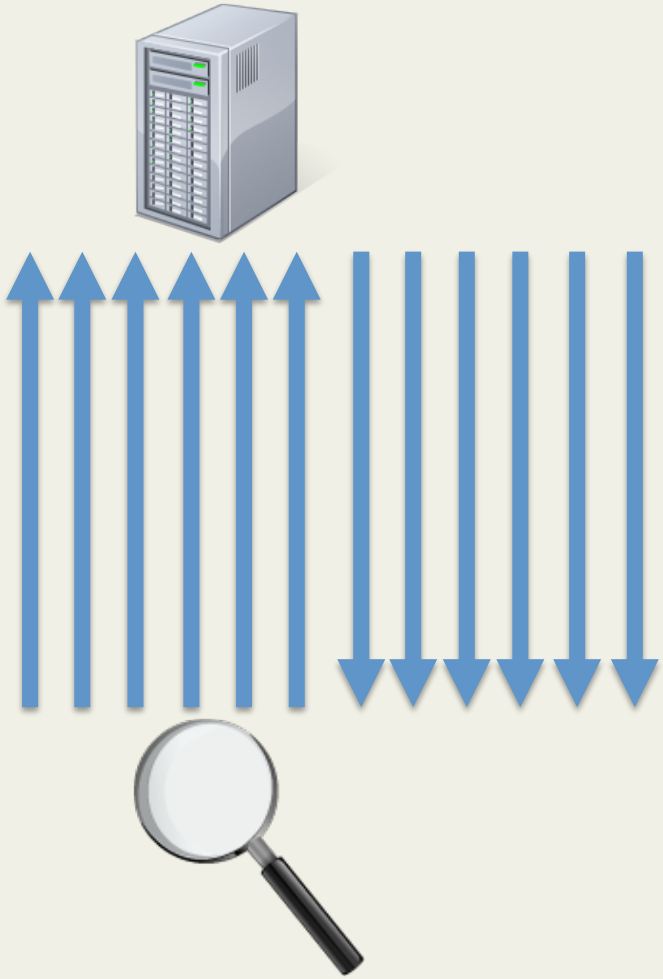


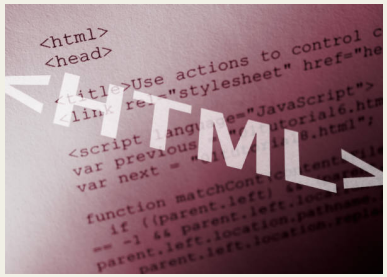


seclab
THE COMPUTER SECURITY GROUP AT UCSB

The Shotgun Approach

GET /view.php?id=
<script>alert(1)</
script>



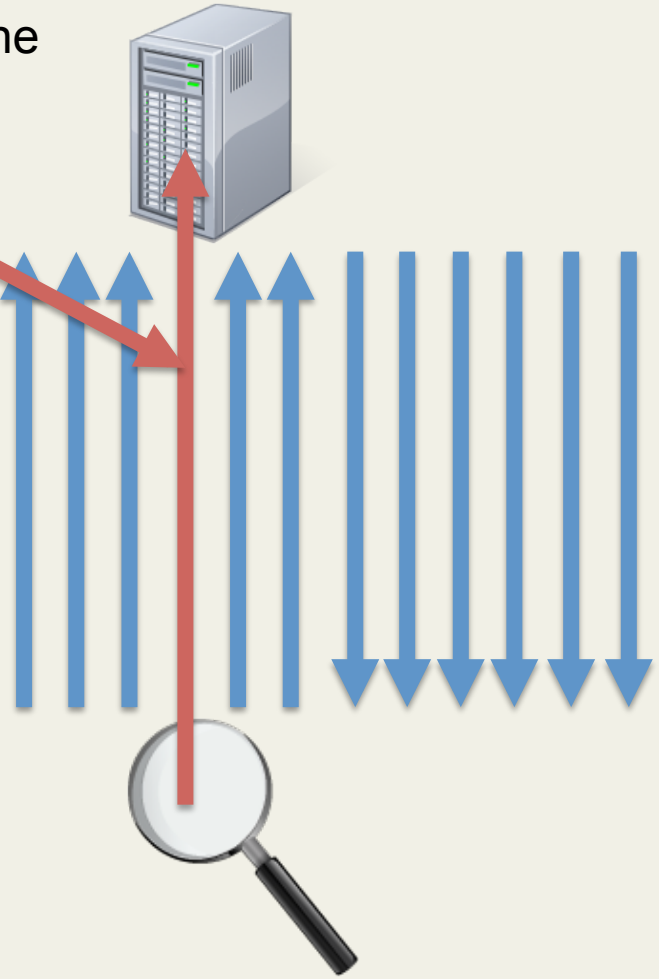


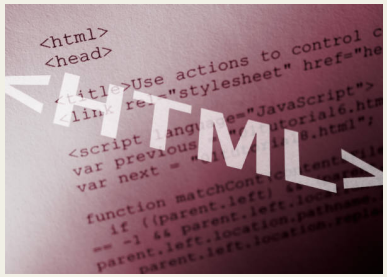
The Shotgun Approach

What if this request changed the state of the application?

Logged the user out?

GET /view.php?id=
<script>alert(1)</
script>





seclab
THE COMPUTER SECURITY GROUP AT UCSB

Simple Web Application



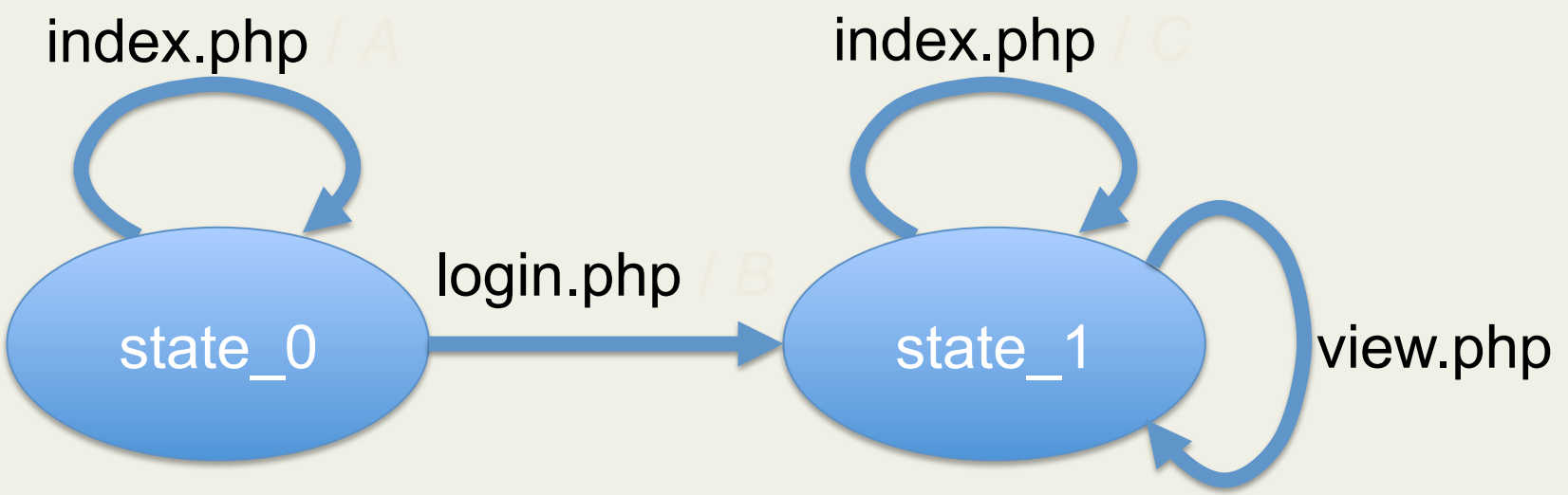
Must access login.php before view.php





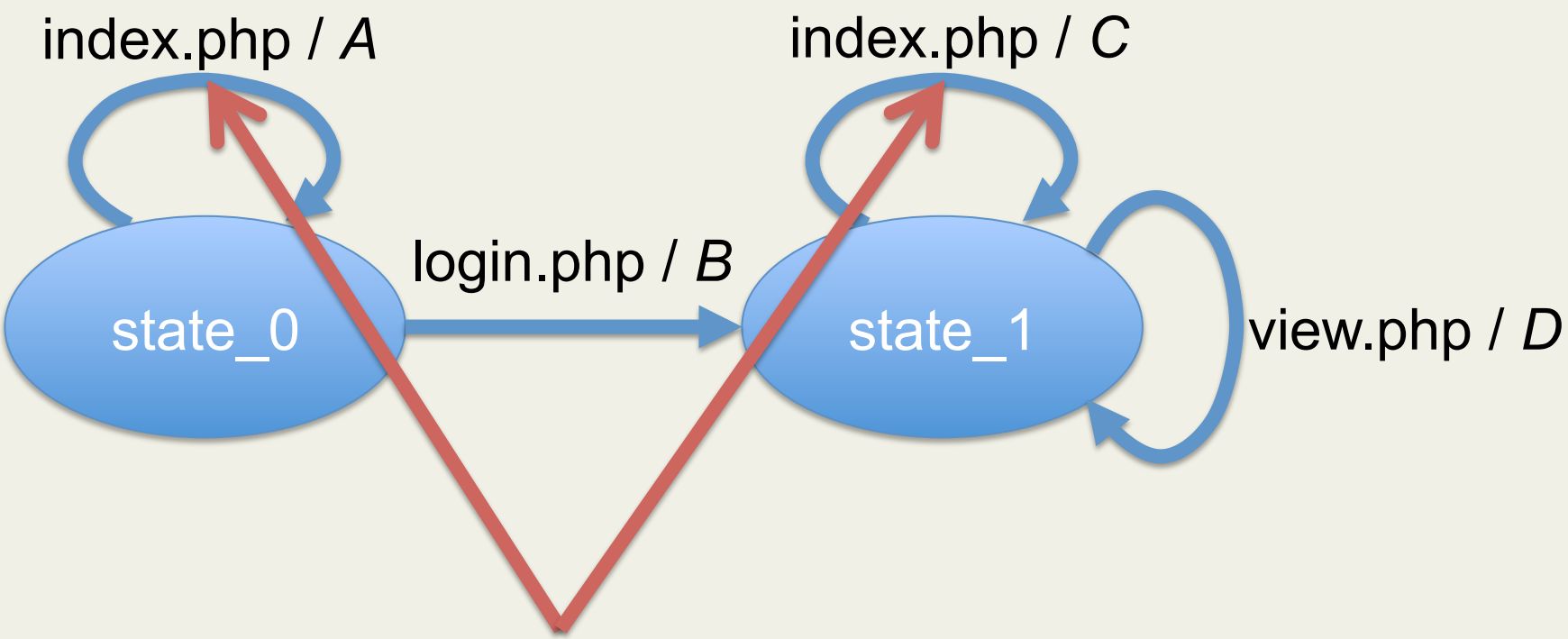
seclab
THE COMPUTER SECURITY GROUP AT UCSB

Internal State Graph



seclab
THE COMPUTER SECURITY GROUP AT UCSB

Mealy Machine



Must fuzz in different states

seclab
THE COMPUTER SECURITY GROUP AT UCSB

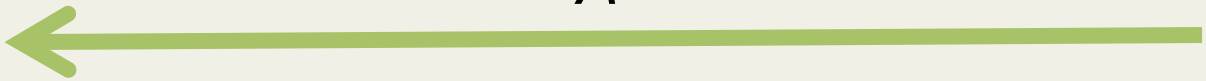
Inferring the State



index.php



A



login.php



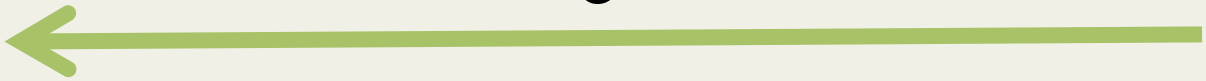
B



index.php



C



view.php



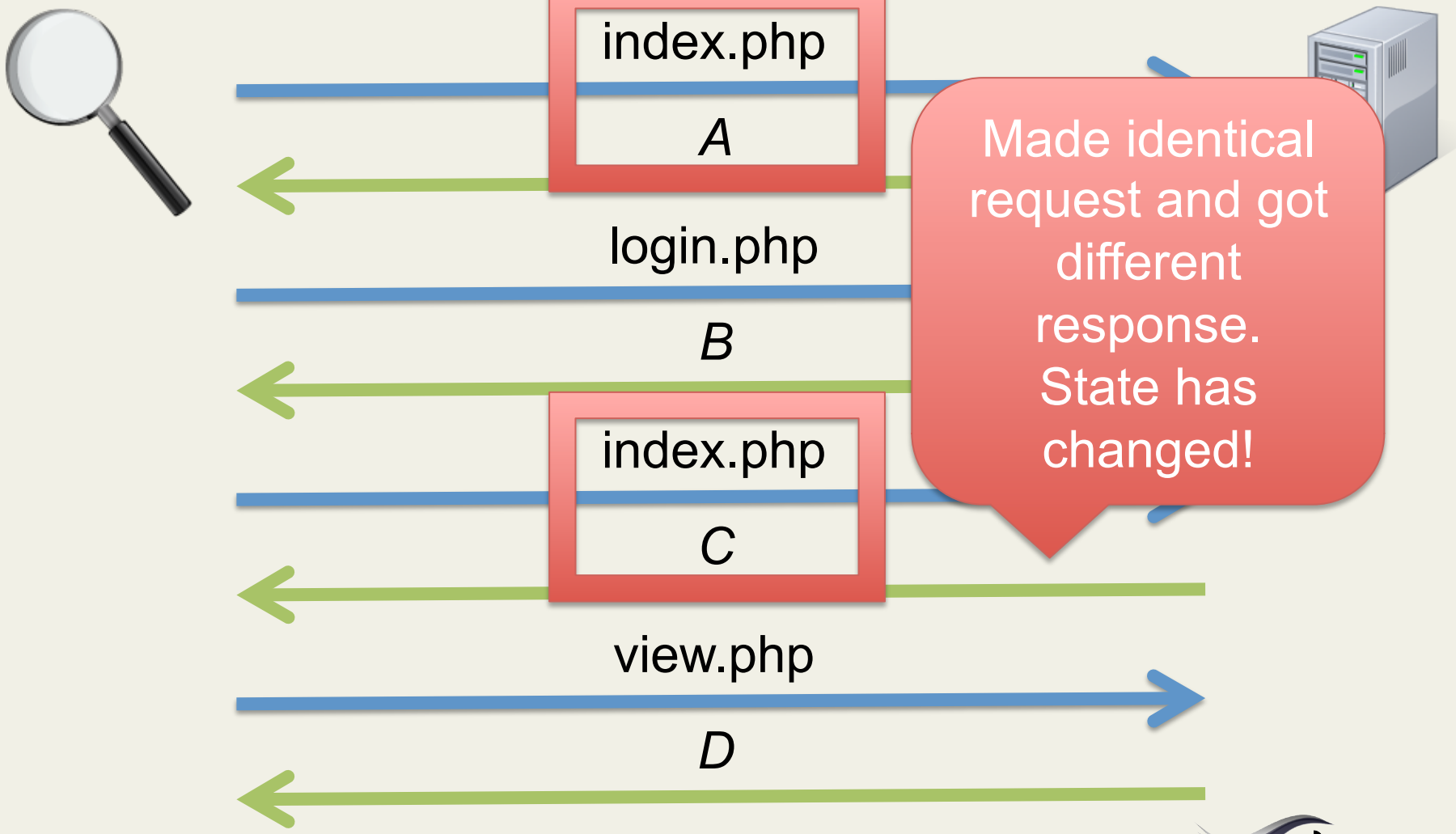
D





seclab
THE COMPUTER SECURITY GROUP AT UCSB

Inferring the State



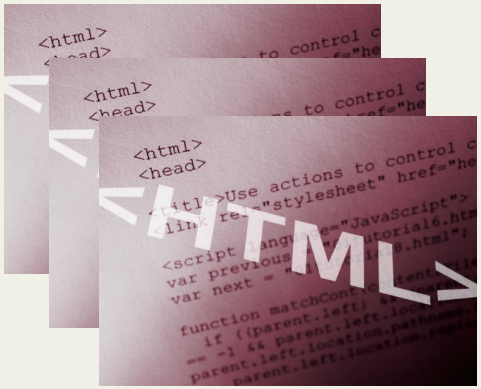
seclab
THE COMPUTER SECURITY GROUP AT UCSB

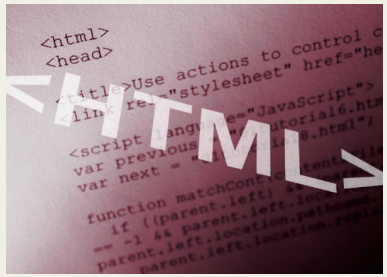
Necessary Steps to Inferring the State

- Cluster similar pages (using links)
  - Links changing means what a user can do to the application has changed
- Determine state-changing request
  - Which request in the list changed the state?
- Collapse similar states
  - How to know if, when we detect a state change, we return to a previous state?

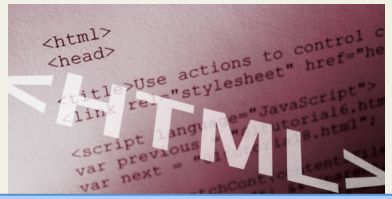
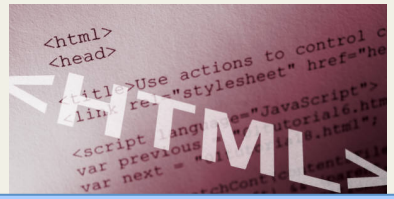
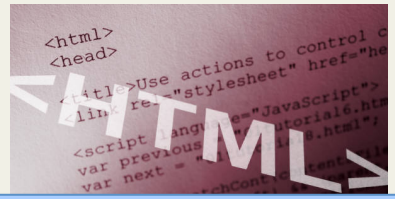
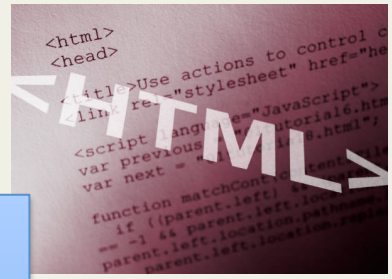
seclab
THE COMPUTER SECURITY GROUP AT UCSB

Cluster Similar Pages



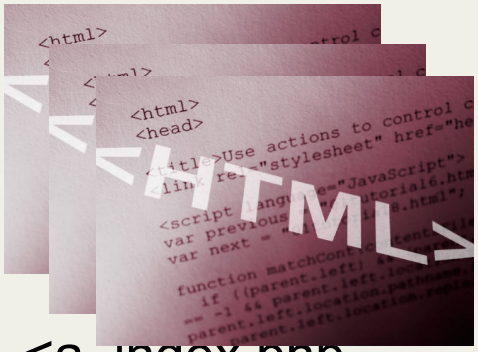


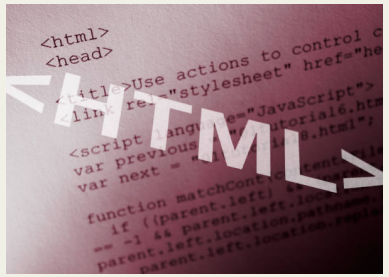
Cluster Similar Pages

<a, index.php, home>	<a, index.php, home>	<a, index.php, home>	<a, index.php, home>
<a, profile.php, id=1>	<a, profile.php, id=2>	<a, profile.php, id=3>	<form, POST, add.php> <a, review.php, check>
<form, POST, logout.php>	<form, POST, logout.php>	<form, POST, logout.php>	

Cluster Similar Pages





<a, index.php, home>

<a, index.php, home>

<a, index.php, home>

<a, profile.php, id=1>

<a, profile.php, id=2>

<a, profile.php, id=3>

<form, POST, logout.php>

<form, POST, logout.php>

<form, POST, logout.php>

<a, index.php, home>

<form, POST, add.php>

<a, review.php, check>

seclab
THE COMPUTER SECURITY GROUP AT UCSB

Determine State-Changing Request

Request	Resp
GET index.php	*A*
GET blah.php	*B*
POST login.php	*C*
GET account.php	*D*
GET index.php	*E*

Use a heuristic that favors new requests over old requests, POST requests over GET requests, and requests that always change the state over those that never change the state.

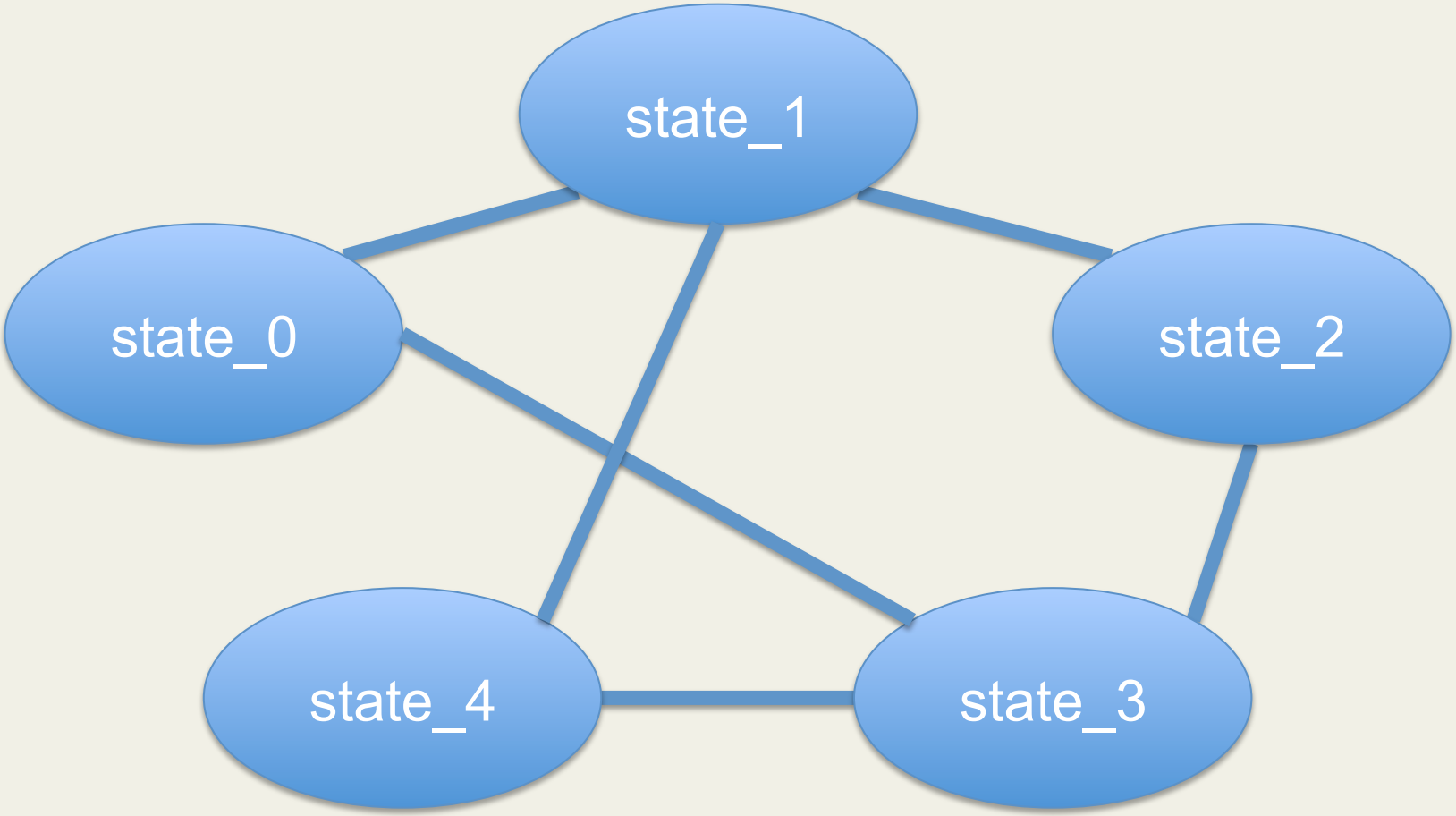
seclab
THE COMPUTER SECURITY GROUP AT UCSB

Collapse Similar States

- Graph coloring
  - States as nodes
  - Edge between two states when they cannot be the same
  - Greedy coloring algorithm

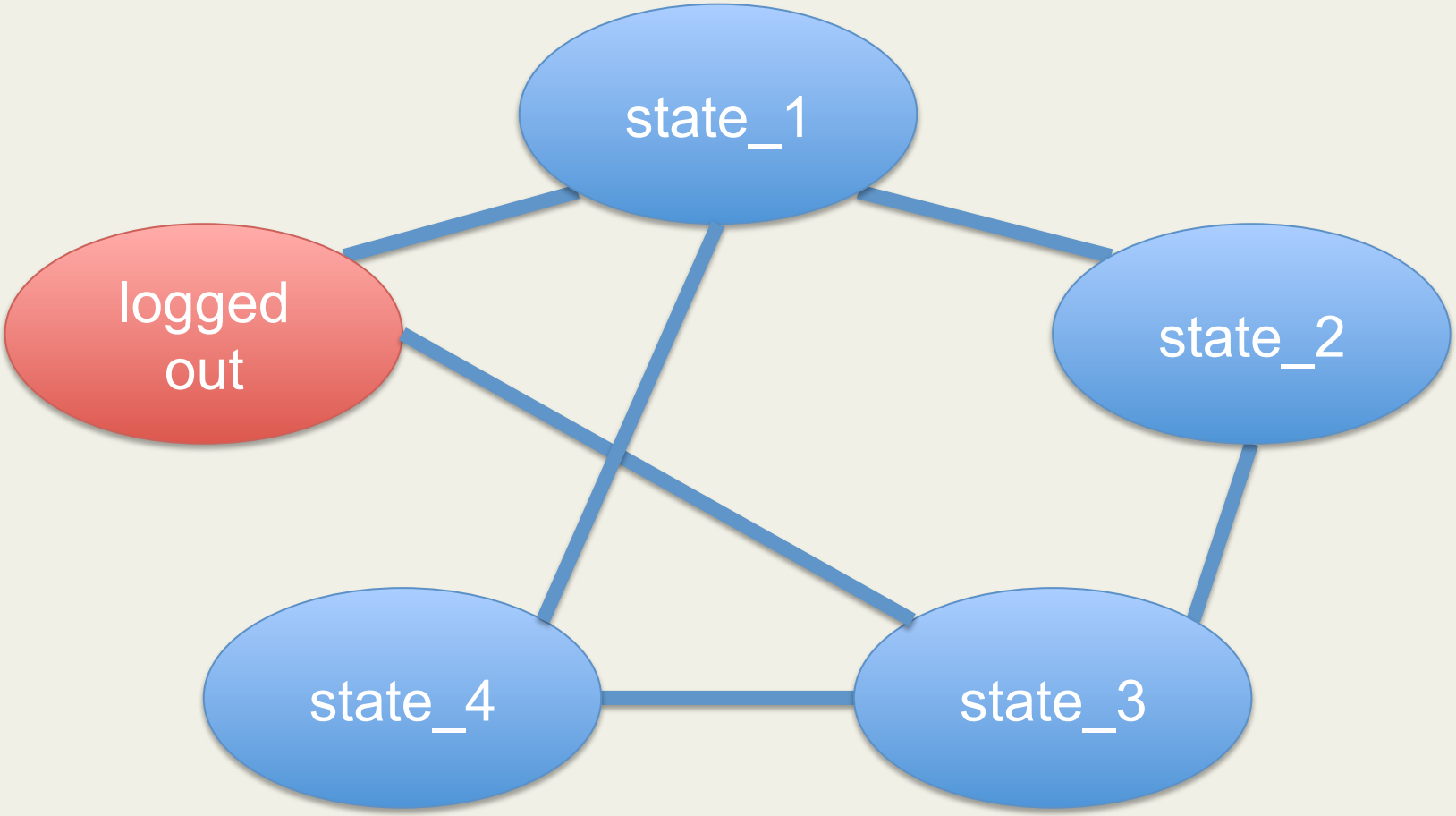
seclab
THE COMPUTER SECURITY GROUP AT UCSB

Collapse Similar States



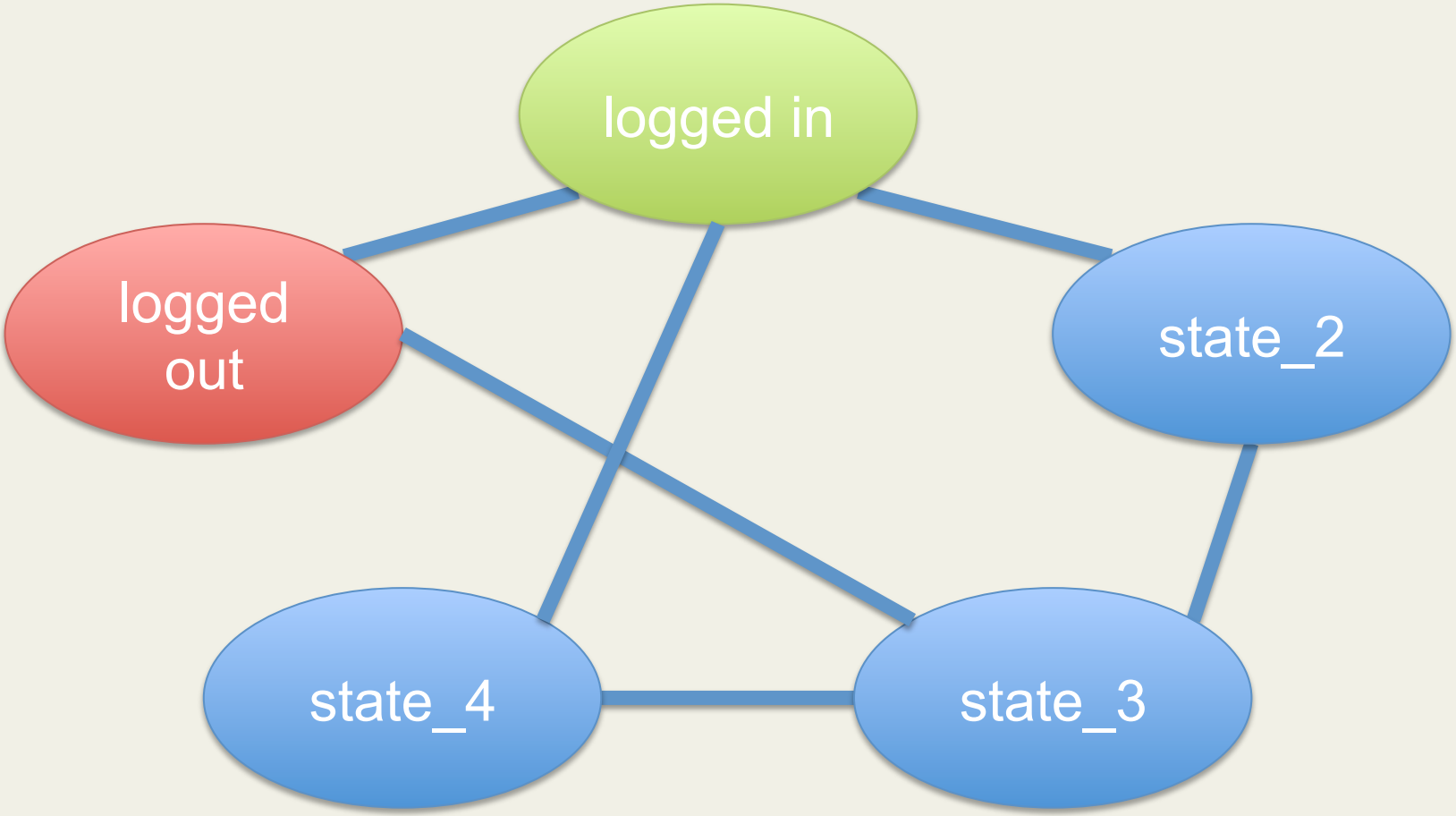
seclab
THE COMPUTER SECURITY GROUP AT UCSB

Collapse Similar States



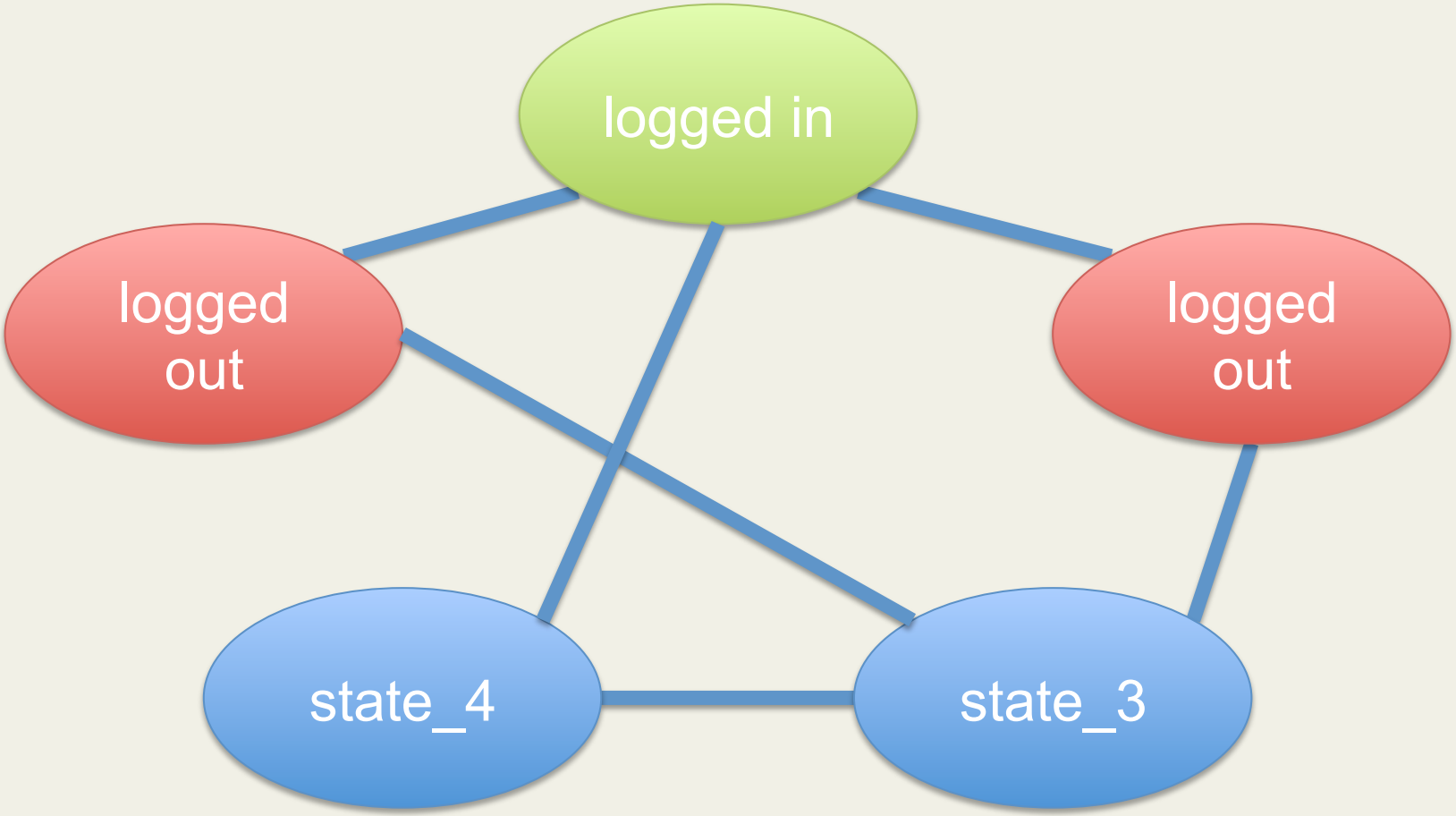
seclab
THE COMPUTER SECURITY GROUP AT UCSB

Collapse Similar States



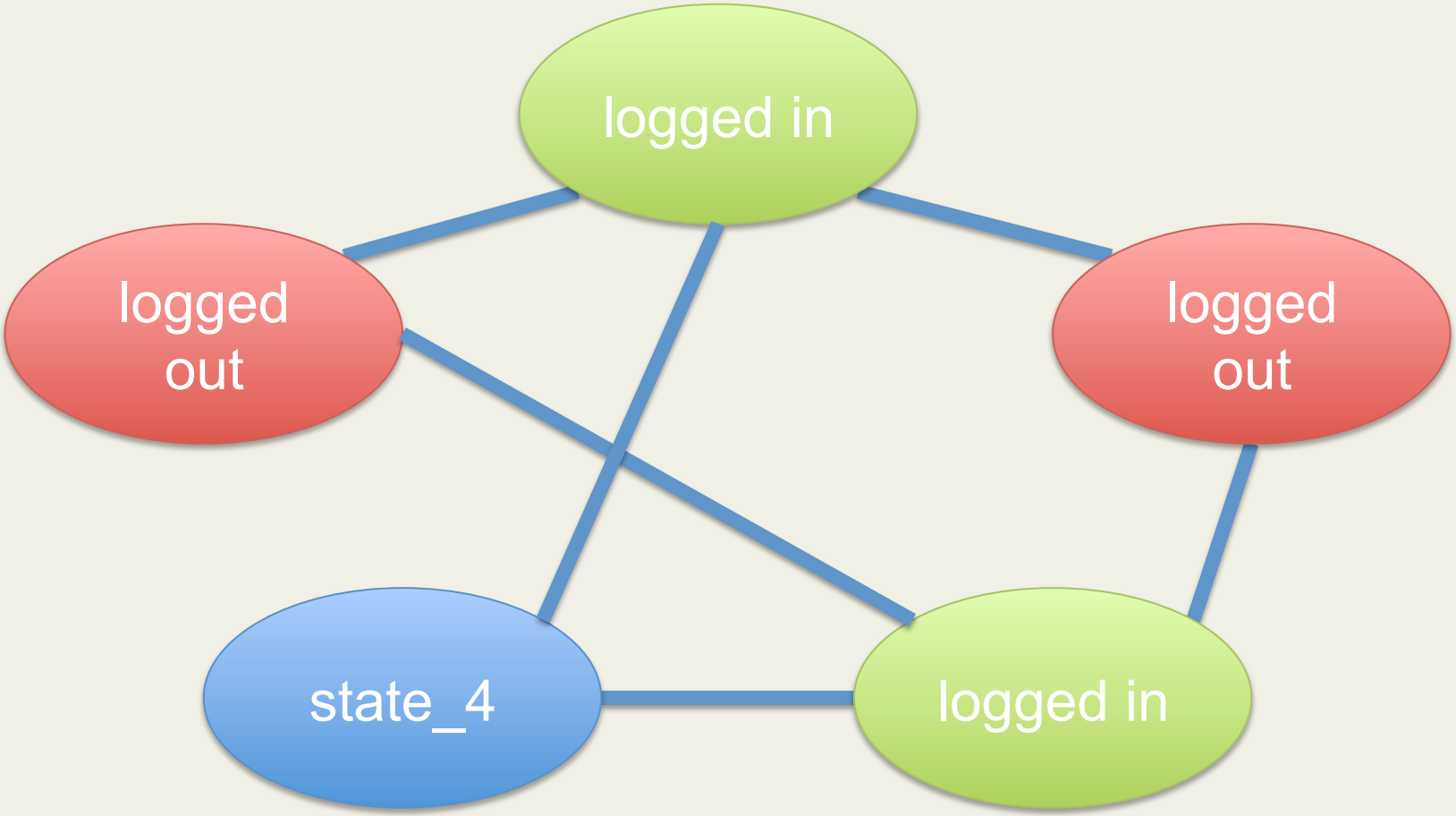
seclab
THE COMPUTER SECURITY GROUP AT UCSB

Collapse Similar States



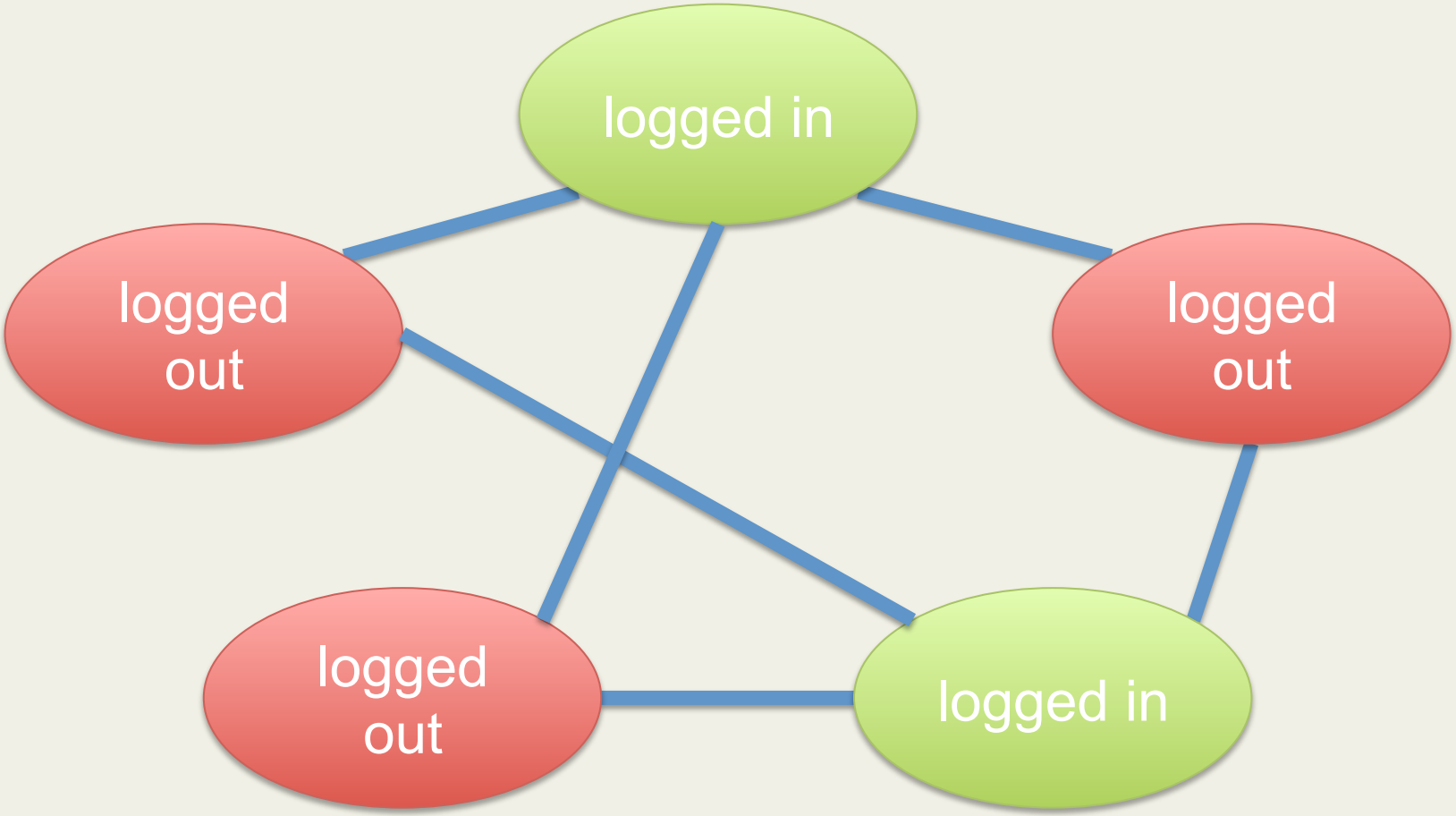
seclab
THE COMPUTER SECURITY GROUP AT UCSB

Collapse Similar States



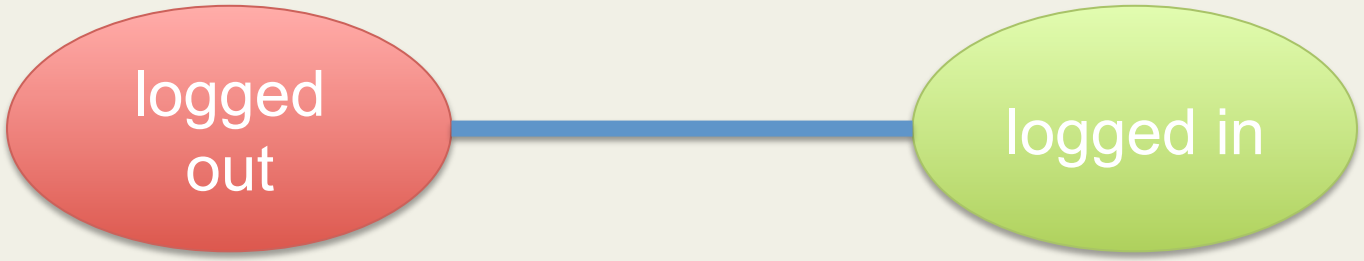
seclab
THE COMPUTER SECURITY GROUP AT UCSB

Collapse Similar States



seclab
THE COMPUTER SECURITY GROUP AT UCSB

Collapse Similar States



seclab
THE COMPUTER SECURITY GROUP AT UCSB

State-Aware Fuzzing

```
def fuzz_state_changing( fuzz_request ):
  make_request( fuzz_request )
  if state_has_changed():
    if state_is_reversible():
      make_requests_to_revert_state()
      if not back_in_previous_state():
        reset_and_put_in_previous_state()
    else:
      reset_and_put_in_previous_state()
```


seclab
THE COMPUTER SECURITY GROUP AT UCSB

Evaluation—Scanners

- skipfish
- w3af
- state-aware-crawler
- wget

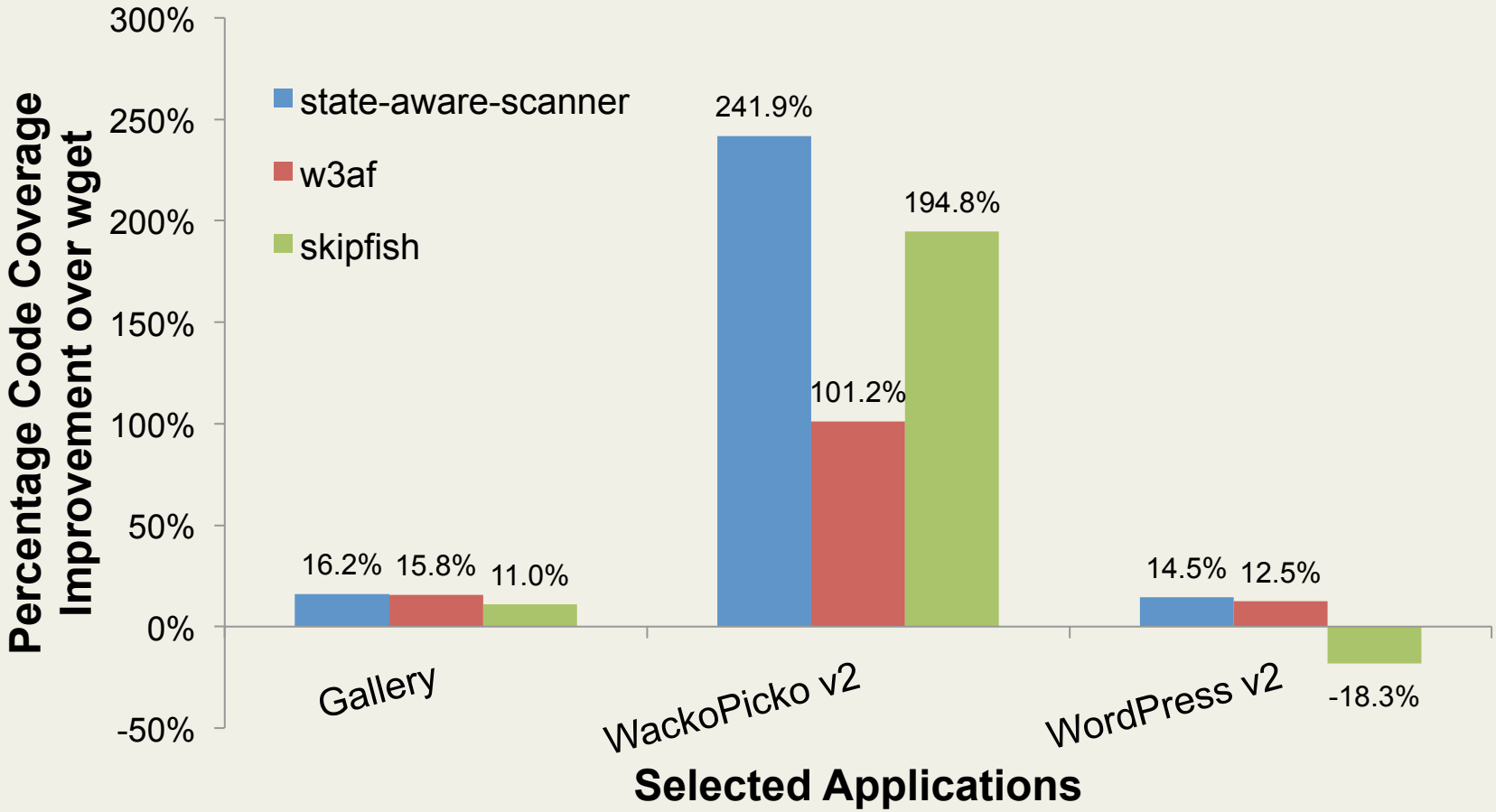
seclab
THE COMPUTER SECURITY GROUP AT UCSB

Evaluation—Applications

Web Application	Lines of Code
Gallery	26,622
PhpBB v2	16,034
PhpBB v3	110,186
SCARF	798
Vanilla Forums	43,880
WackoPicko v2	900
WordPress v2	17,995
WordPress v3	71,698

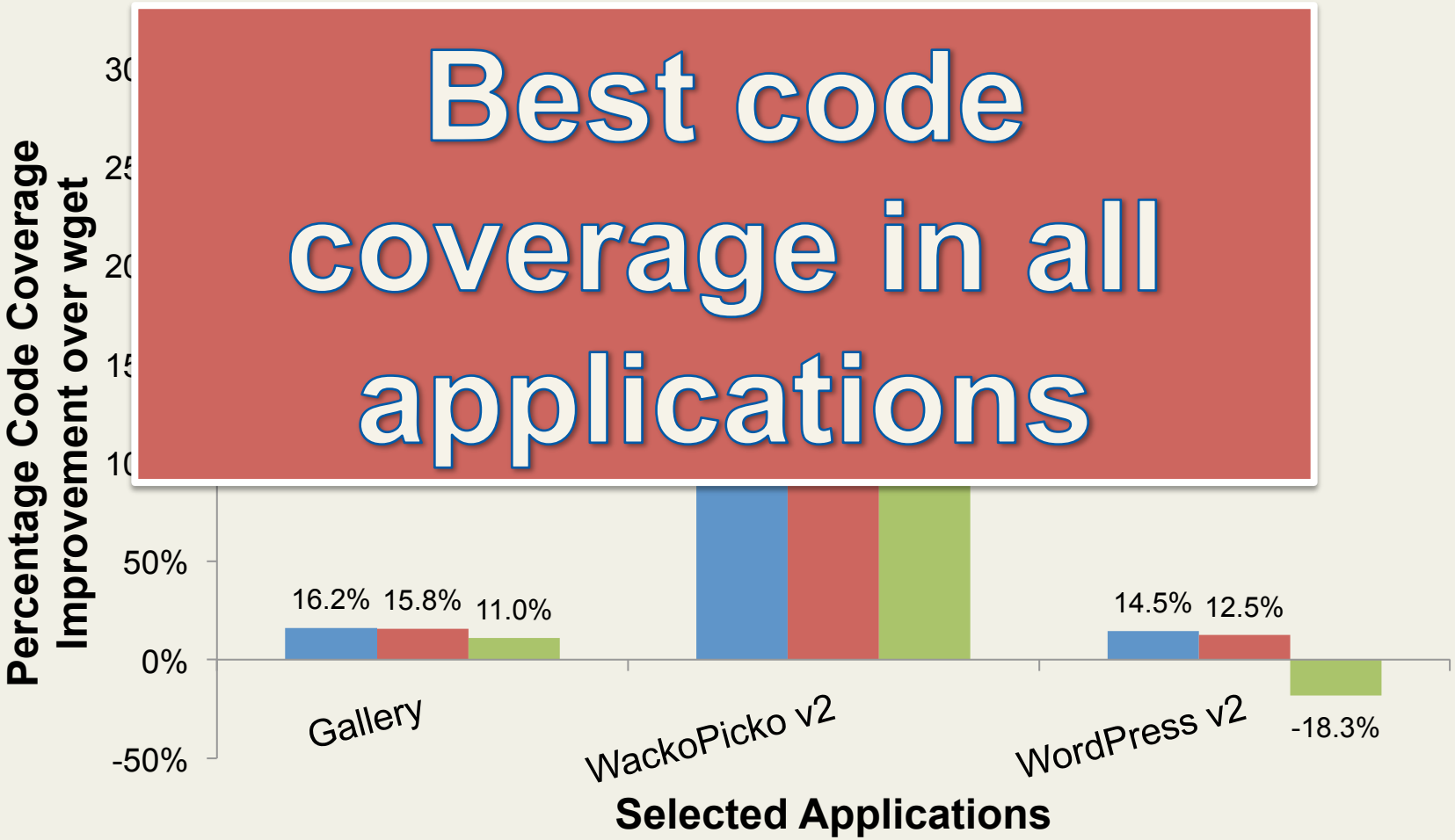
seclab
THE COMPUTER SECURITY GROUP AT UCSB

Code Coverage Results



seclab
THE COMPUTER SECURITY GROUP AT UCSB

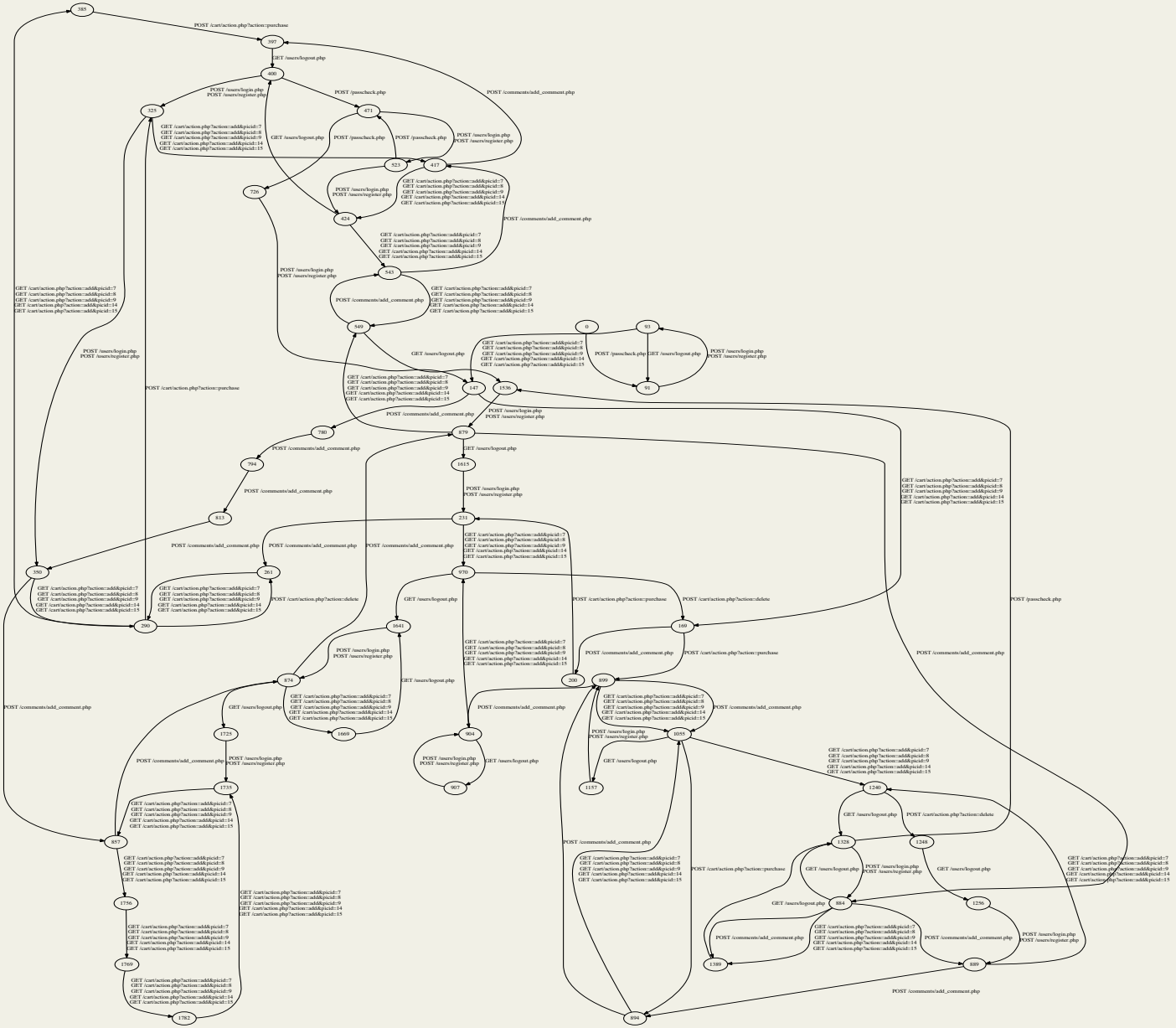
Code Coverage Results



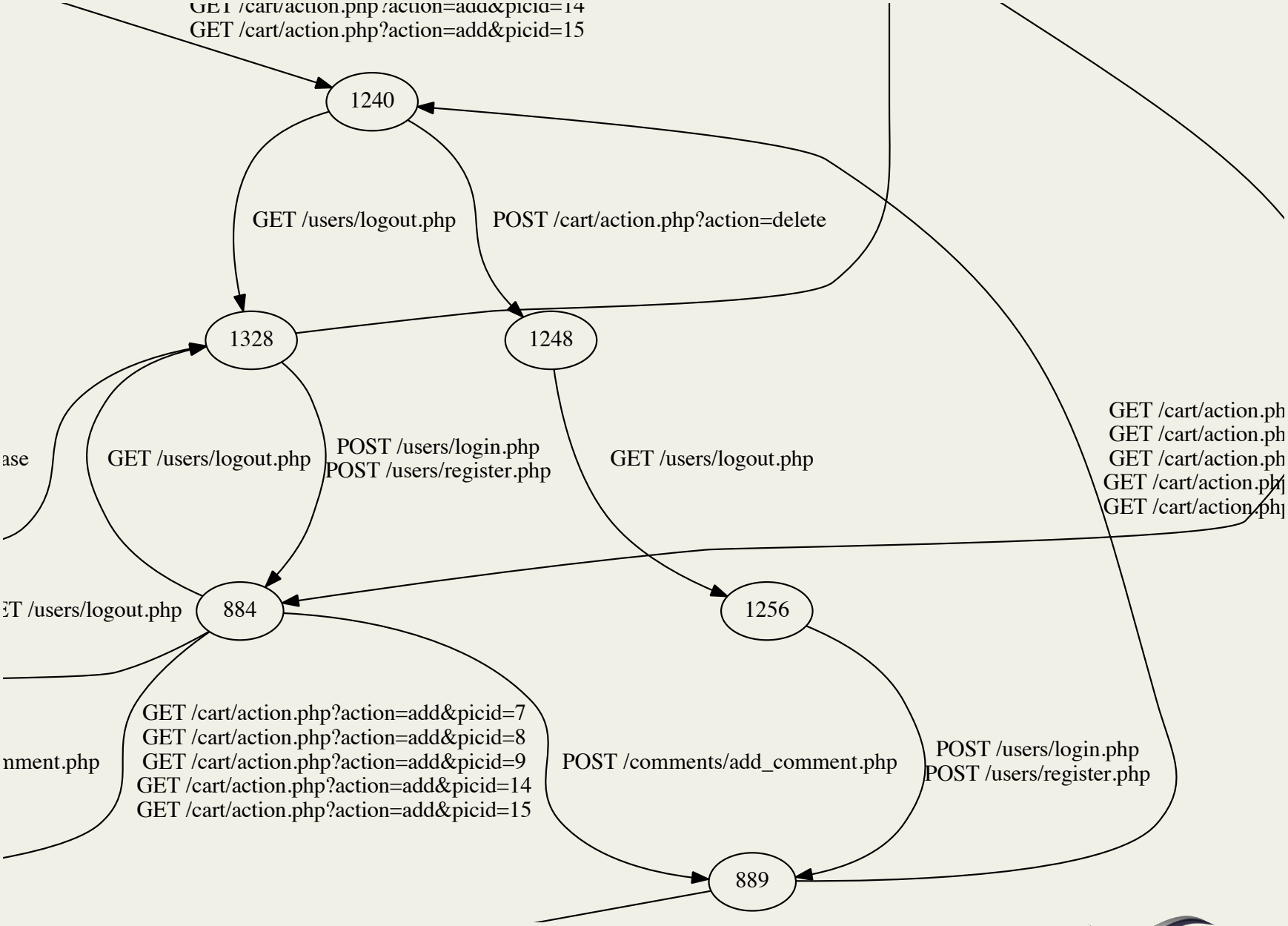
seclab
THE COMPUTER SECURITY GROUP AT UCSB

Web App	Scanner	Code %	True Vuln	Unique Vuln
PhpBB v2	state	38.34	3	1
PhpBB v2	w3af	1.04	1	0
PhpBB v2	skipfish	5.10	2	0
SCARF	state	67.03	1	1
SCARF	w3af	55.66	0	0
SCARF	skipfish	21.55	0	0
Vanilla	state	30.89	0	0
Vanilla	w3af	1.06	0	0
Vanilla	skipfish	-2.32	15	2
WackoPicko	state	241.86	5	1
WackoPicko	w3af	101.15	5	1
WackoPicko	skipfish	194.77	3	1

seclab
THE COMPUTER SECURITY GROUP AT UCSB



seclab
THE COMPUTER SECURITY GROUP AT UCSB



seclab
THE COMPUTER SECURITY GROUP AT UCSB

Adam Doupé

Email: adoupe@cs.ucsb.edu

Twitter: @adamdoupe

ENEMY OF THE STATE: A STATE-AWARE BLACK-BOX WEB VULNERABILITY SCANNER

seclab
THE COMPUTER SECURITY GROUP AT UCSB