APPLICATIONS OF SYMBOLIC EXECUTION

Chong-Kuan Chen 陳仲寬 @ DSNSLab NCTU

https://www.facebook.com/Bletchley13

Twitter: @bletchley13

Bletchley13@gmail.com

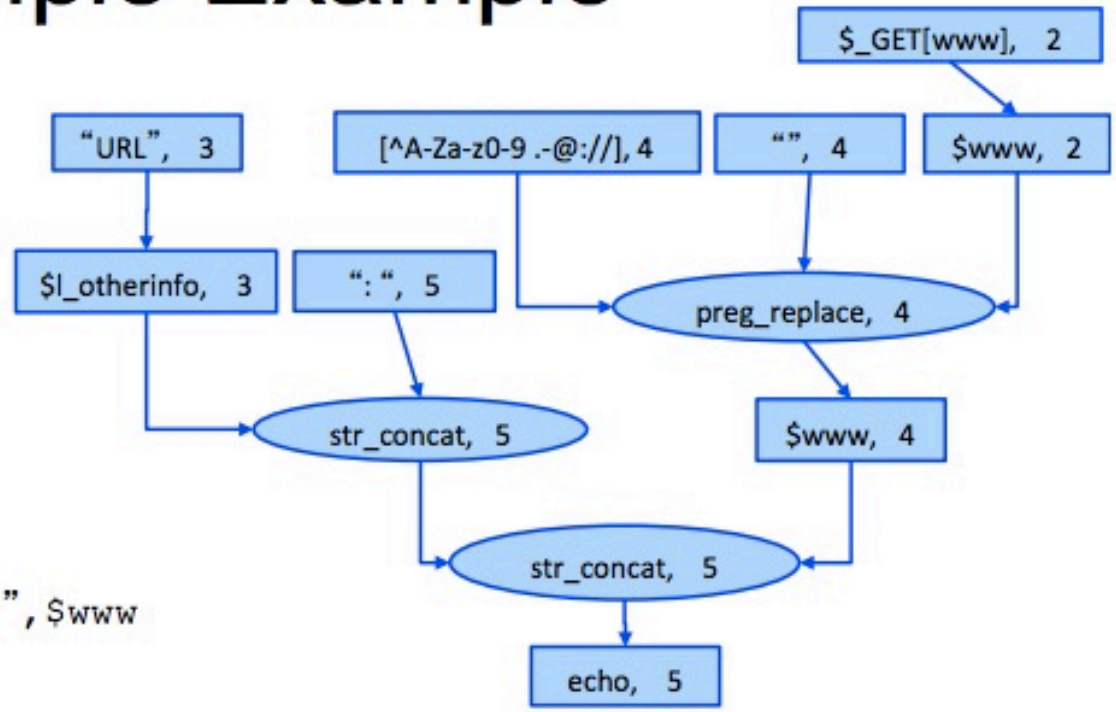
AUTOMATA-BASED STRING ANALYSIS

- Tevfik Bultan. 2015. String Analysis for Vulnerability Detection and Repair. In Proceedings of the 22nd International Symposium on Model Checking Software

- In their work, they try to detect the vulnerabilities in the web, and then repair these vulnerabilities.

- Web application contains many various length strings, the conventional symbolic execution is difficult to deal with.

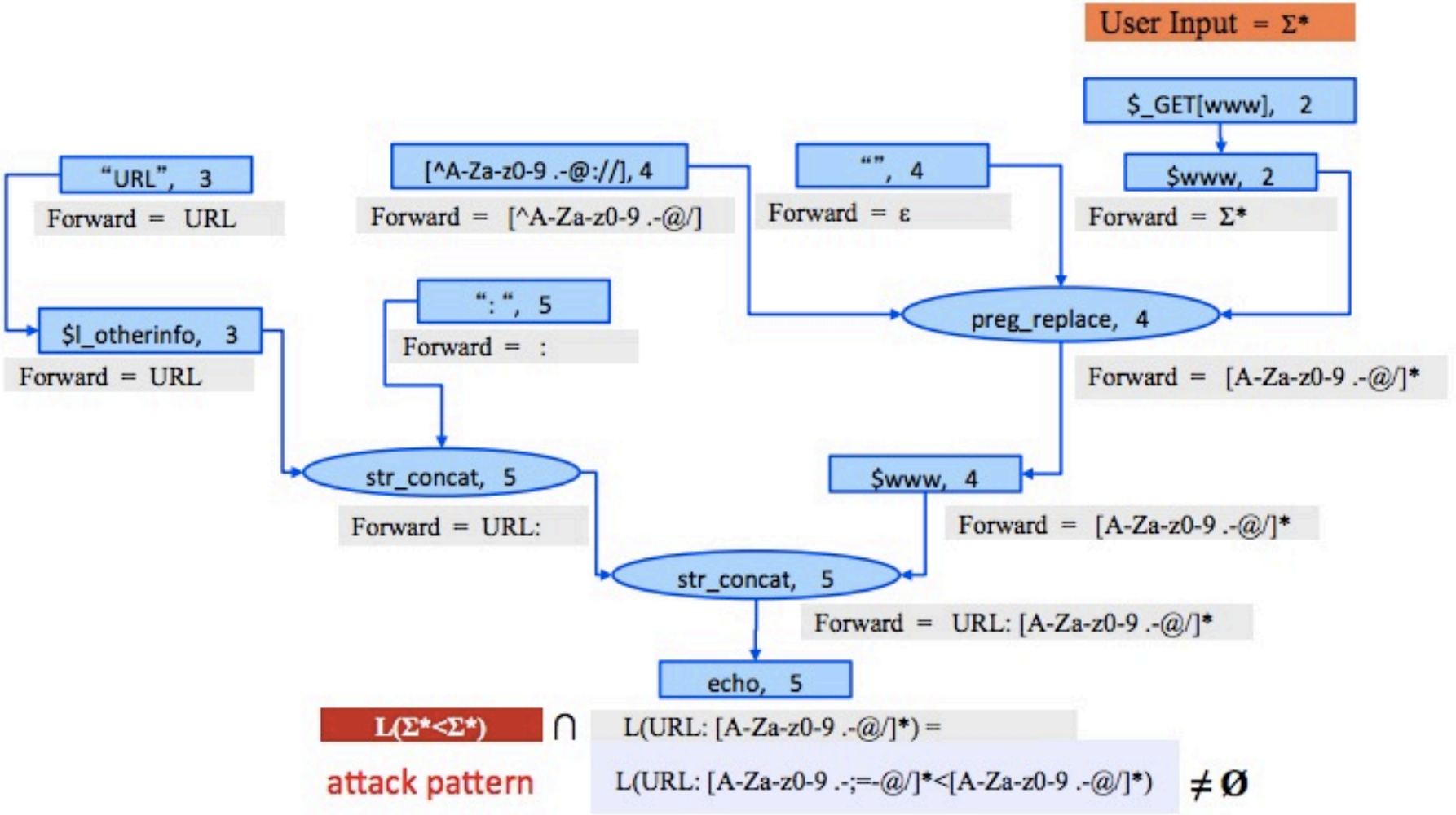
Simple Example




```
1:<?php
2: $www = $ GET["www"];
3: $l_otherinfo = "URL";
4: $www = ereg_replace(
   "[^A-Za-z0-9 .-@://]","",$www
   );
5: echo $l_otherinfo .
   ": " .$www;
6:?>
```

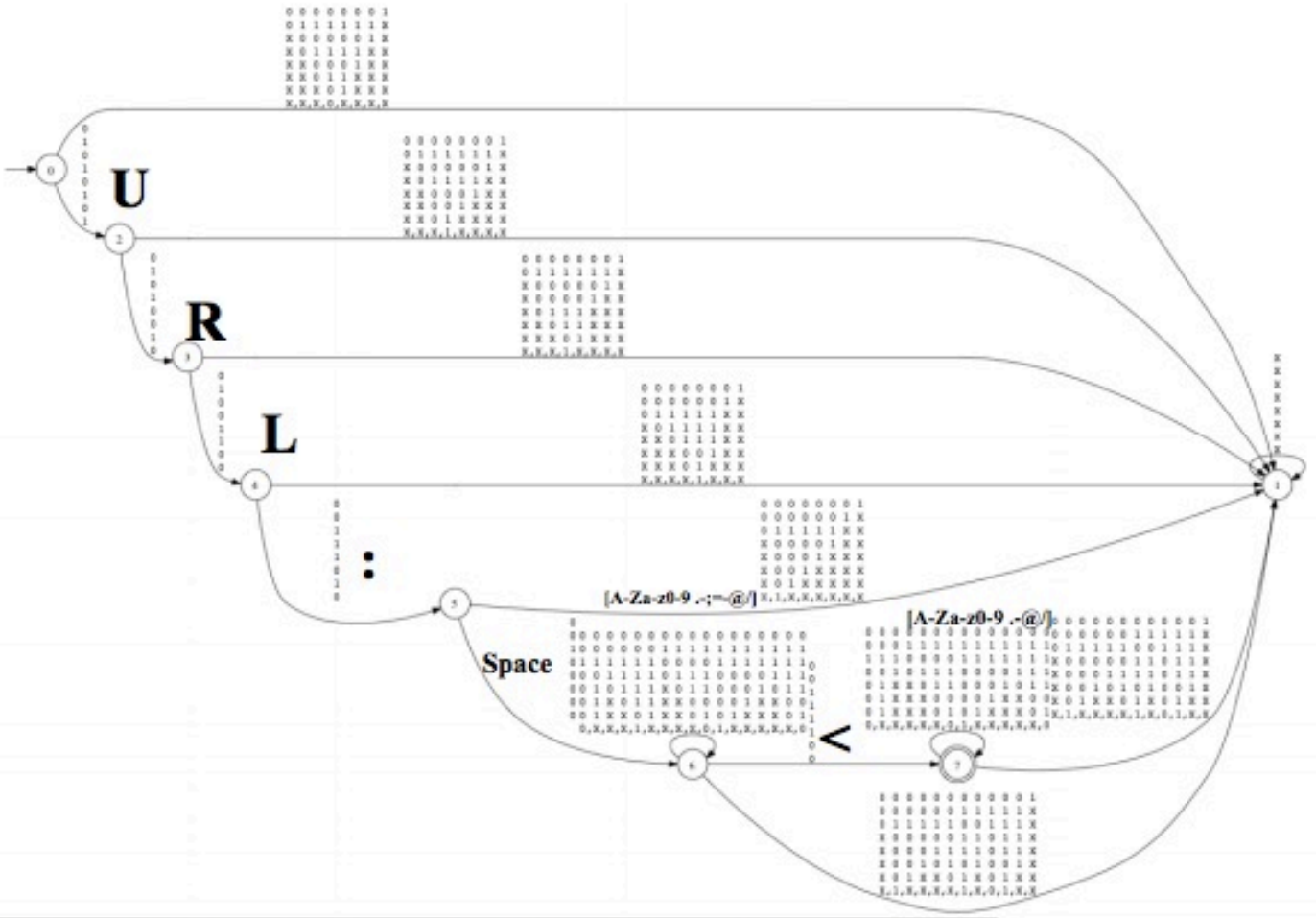
String Analysis for Vulnerability Detec5on and Repair, Tevfik Bultan, SPIN 2015
https://pdfs.semanticscholar.org/a175/7bee1ac1794c73d212cfa1a30d1f63be14b0.pdf

Forward Analysis



String Analysis for Vulnerability Detec5on and Repair, Tevfik Bultan, SPIN 2015
https://pdfs.semanticscholar.org/a175/7bee1ac1794c73d212cfa1a30d1f63be14b0.pdf

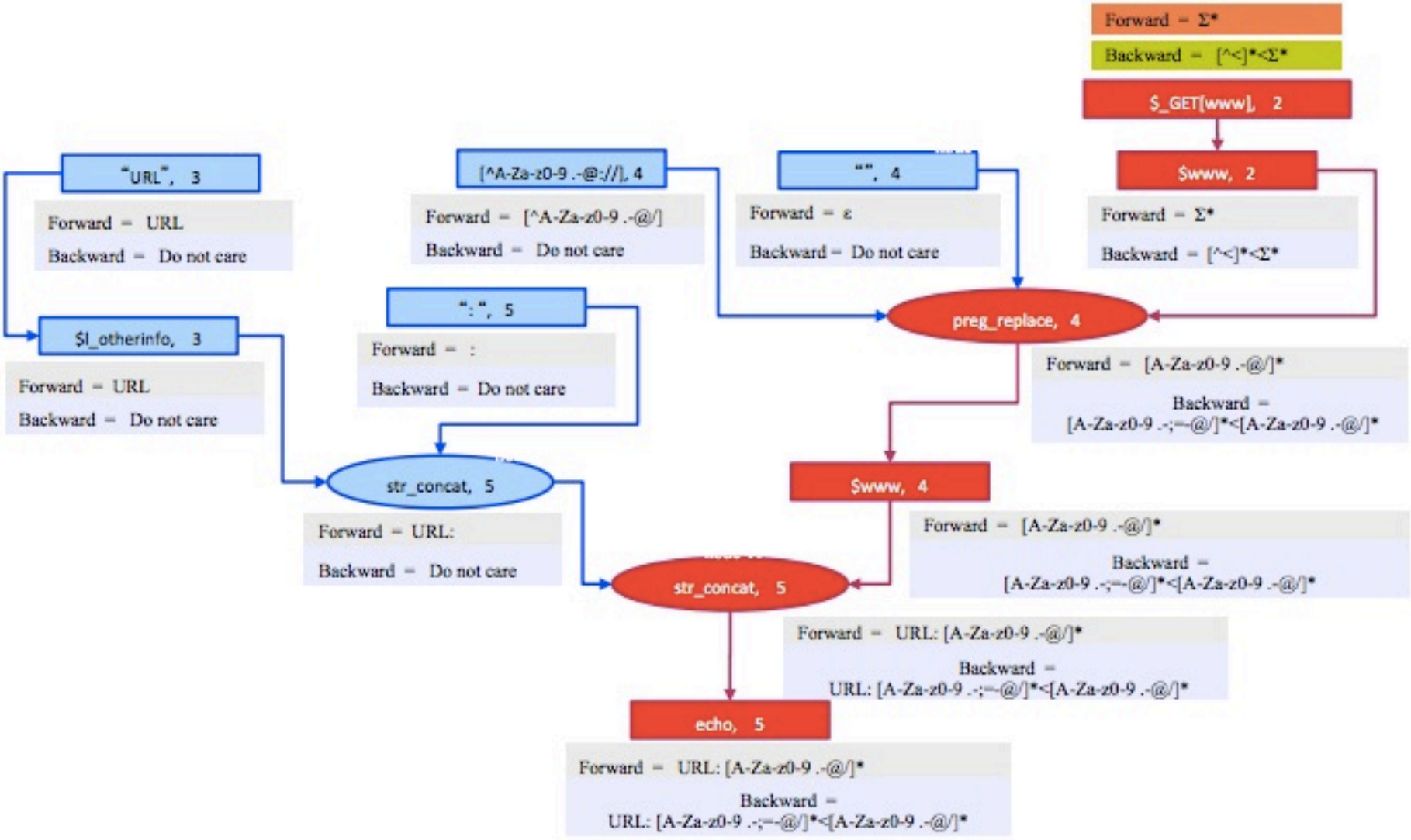
Result Automaton



URL: [A-Za-z0-9 .-;=-@/]*<[A-Za-z0-9 .-@/]*

String Analysis for Vulnerability Detec5on and Repair, Tevfik Bultan, SPIN 2015
https://pdfs.semanticscholar.org/a175/7bee1ac1794c73d212cfa1a30d1f63be14b0.pdf

Backward Analysis



String Analysis for Vulnerability Detec5on and Repair, Tevfik Bultan, SPIN 2015
https://pdfs.semanticscholar.org/a175/7bee1ac1794c73d212cfa1a30d1f63be14b0.pdf

MBA

- Malware behavior analysis developed by DSNS@NCTU
- Open source
  - https://github.com/GlacierW/MBA
- Windows 10 64 bits Analysis Environment Support
- Binary Analysis Modules
  - System call tracer, instruction tracer, out-of-box hooking, whole-system taint engine
- Virtual Machine Forensics Modules
  - Disk, registry, network traffic, and memory forensics
- Extensibility
  - Module APIs exported
- Ready to Use
  - Interactive command interface provided

MBA

- Currently, we are trying to integrate MBA with symbolic execution

```
hao@MBA2:~/thesis/MBA$ ./start.sh
QEMU 2.3.50 monitor - type 'help' for more information
(qemu) load_global_variable g
(qemu) load_structures t
(qemu) mba_winit
Agent thread starting
(qemu) cclc /home/hao/MBA/fm
find ntkrnlmp.pdb in offset fffff801a8e10000
guid A0BB79D9FEF34DCF9B60121A8D715FA41
[cclc] result: 0
(qemu) ----reset triton----
0x401000
0x408433
KPCR found fffff801a9173000
System Receive : impo concolic.exe
add new inst tracer
```

MBA

```
symbolized branch instruction
====solving constraint====
[O]7657c6b6 -> 7657c6bd
        SymVar_0 = 0xF5
[ ]7657c6b6 -> 7657c6b8
        SymVar_0 = 0xA
============================
symbolized branch instruction
====solving constraint====
[ ]7657c6e0 -> 7657c7ef
        SymVar_0 = 0x1A
[O]7657c6e0 -> 7657c6e6
        SymVar_0 = 0xE5
============================
symbolized branch instruction
====solving constraint====
[ ]7657c6e8 -> 7657c6f3
        SymVar_0 = 0xD
[O]7657c6e8 -> 7657c6ea
        SymVar_0 = 0xF2
============================
```

```
symbolized branch ins
====solving constrain
[ ]4083dc -> 4083ef
        SymVar_0 = 0x
[O]4083dc -> 4083de
        SymVar_0 = 0x
============================
```