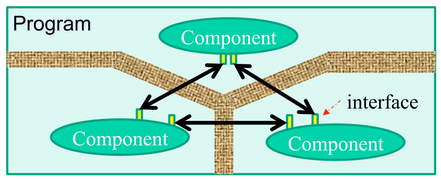
Dereference Under the Influence (DUI)
You Can't Afford It

School of Computing, NUS National University of Singapore

Hong Hu, Zheng Leong Chua, Prateek Saxena, Zhenkai Liang

Problem Introduction

- Security-critical components are often protected using isolation mechanisms
  - Interactions via API interfaces



- Attackers can affect the protected component by input to interfaces
  - Data values
  - Memory addresses
- We call memory dereference affected by attackers *Dereference Under the Influence (DUI)*.

DUI Detector: An automatic tool to detect DUI

binary

input

Execution State Collection

- Executed instructions log
  - Raw instruction
  - Instruction operands
  - Memory states
- Module loading/unloading log
  - Tracking memory page permission
- Dynamic taint tracking
  - Fine-grained taint source tracking
  - 1-level table lookup

Instruction Shortlisting

- Write DUI detection
  - Memory writing instruction
  - Tainted source operand register
  - Tainted writing address

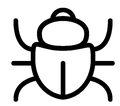
```
mov   %eax, (%esi)
```

- Read DUI detection
  - Memory read instruction
  - Tainted read address
  - Result is used at sinks

```
mov    (%esi), %eax
……
sink(%eax)
```

Access Behavior Analysis

- Trace formula generation
  - Data-flow constraints
  - Control-flow constraints
  - Memory permission
  - Data life-cycle
- Attacker's capability estimation
  - Build queries on memory
    - Bit-pattern
    - Range
  - Solve the query using solver
- DUI filtering



vulnerability

severity

Types of DUI

- **Write DUI**: memory writing operation

```
v1 = API_recv();
v2 = API_recv();
array[v1] = v2;
```

memory corruption

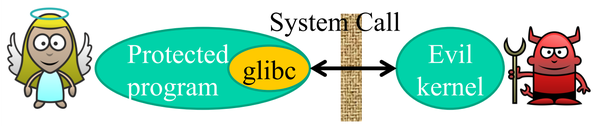
- **Read DUI**: memory read operation

```
v1 = API_recv();
data = *(base+v1);
API_send(data);
```

information leakage

* API_recv() receives data from outside
* API_send() sends data to outside

DUI in glibc



brk system call

Setup the heap region:

```
addr1 = brk(arg1)
addr2 = brk(arg2)
*(addr1 + 4) = addr2 - addr1
```

Corresponding inst.:

```
mov %eax, 0x4(%edx)
…
mov %eax, 0x4(%edi)
```

Detected DUIs

condition (brk1 %8 == 0 && brk2 > brk1)

```
address = brk1 + 0x2718 ;
data = (brk2 - brk1 - 0 x2718) | 0x1;
```

condition (brk1 %8 != 0 && brk1 < brk2 && brk2< brk3)

`address` : dependent on brk1 ;
`data` : dependent on brk1 and brk2 ;

condition (brk1 %8 != 0 && brk1 < brk2 && brk2 > brk3)

`address` : dependent on brk1 ;
`data` : dependent on brk1 and brk3 ;

mmap2 system call

Map files or devices into memory

Related inst. :

```
mov   %eax, 0x1ac(%edi)
```

Conclusion

- Attackers can influence memory operations of isolated components through inputs to their public interfaces.
- We present DUI Detector, an automatic tool to detect dereference under the influence (DUI) through memory access patterns in execution traces.

This work is supported by the Ministry of Education, Singapore under Grant No. R-252-000-495-133.

Some pictures courtesy of http://icons8.com/