
Secure Systems Engineering

Chester Rebeiro

Indian Institute of Technology Madras

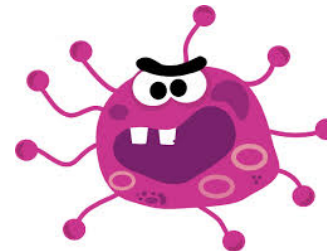
Secure Systems

- Computer systems can be considered a closed box.
- Information in the box is safe as long as nothing enters or leaves the box.



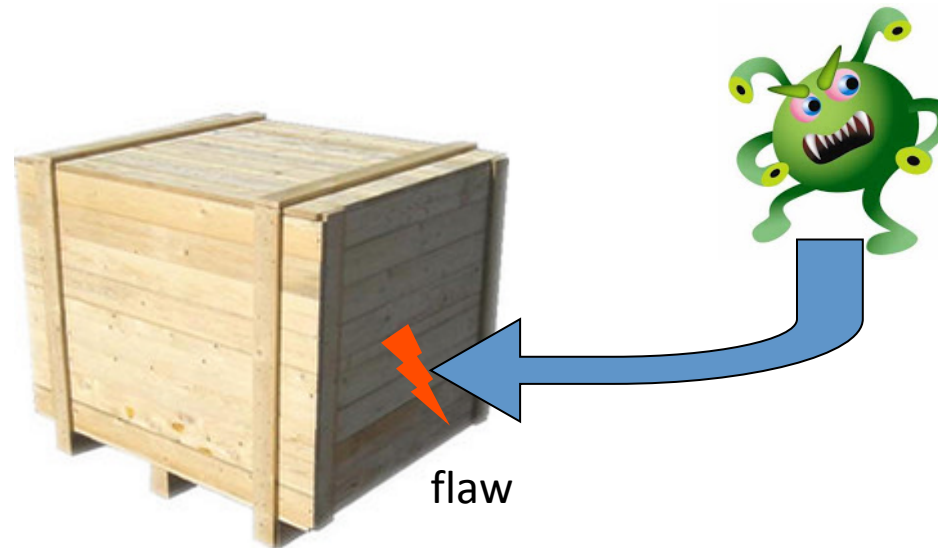
Systems Still Secure

- Even with viruses, worms, and spyware around information is still safe as long as they do not enter the system



Vulnerability

- A flaw that an attacker can use to gain access into the system





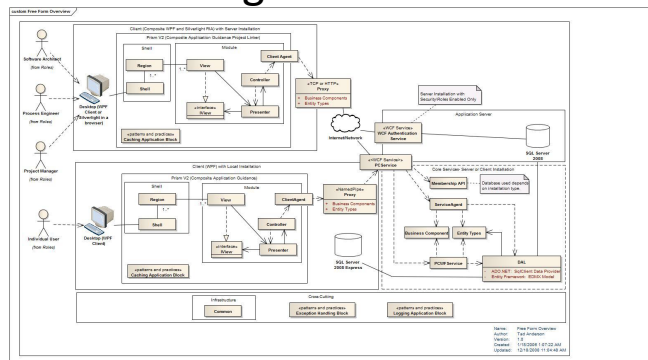
Flaws that would allow an attacker access a system

The attacker just needs oneflaw!!!

Hardware Flaws



Design Flaws



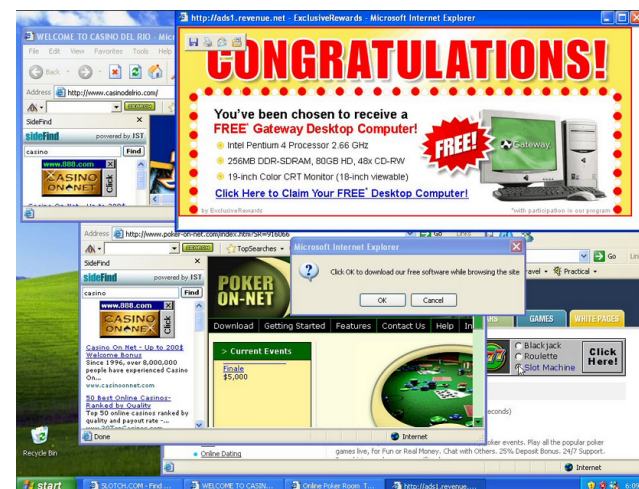
flaw

The Human factor



Bugs in the Program

```
qemu-option.c (~/.work/deca1.9) - VIM
const char *tag, const char **pstr)
{
    const char *p;
    char option[128];
    p = *pstr;
    for(;;) {
        p = get_opt_name(option, sizeof(option), p, '-');
        if (*p != '-')
            break;
        p++;
        if (strcmp(tag, option)) {
            *pstr = get_opt_value(buf, buf_size, p);
            if (**pstr == ',') {
                (*pstr)++;
            }
            return strlen(buf);
        } else {
            p = get_opt_value(NULL, 0, p);
        }
        if (*p != '-')
            break;
        p++;
    }
}
```



Program Flaws

- In application software
 - SQL Injection
- In system software
 - Buffers overflows and overreads
 - Heap: double free, use after free
 - Integer overflows
 - Format string
- Side Channels Attacks
 - Cache timing attacks
 - Power Analysis Attacks
 - Fault Injection Attacks

Secure Systems Engineering

Approach 1: Design flawless systems

eg. SeL4

(Not easy to develop these systems in a large scale)



Static analysis /
Formal Proof Assistant
eg. COQ

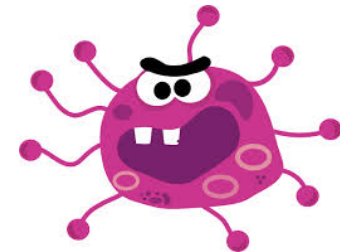


Secure Systems Engineering

Approach 2: Isolate systems : sandbox environments, virtual machines, trusted execution environments (trusted computing)



Takes care of the human factor as well

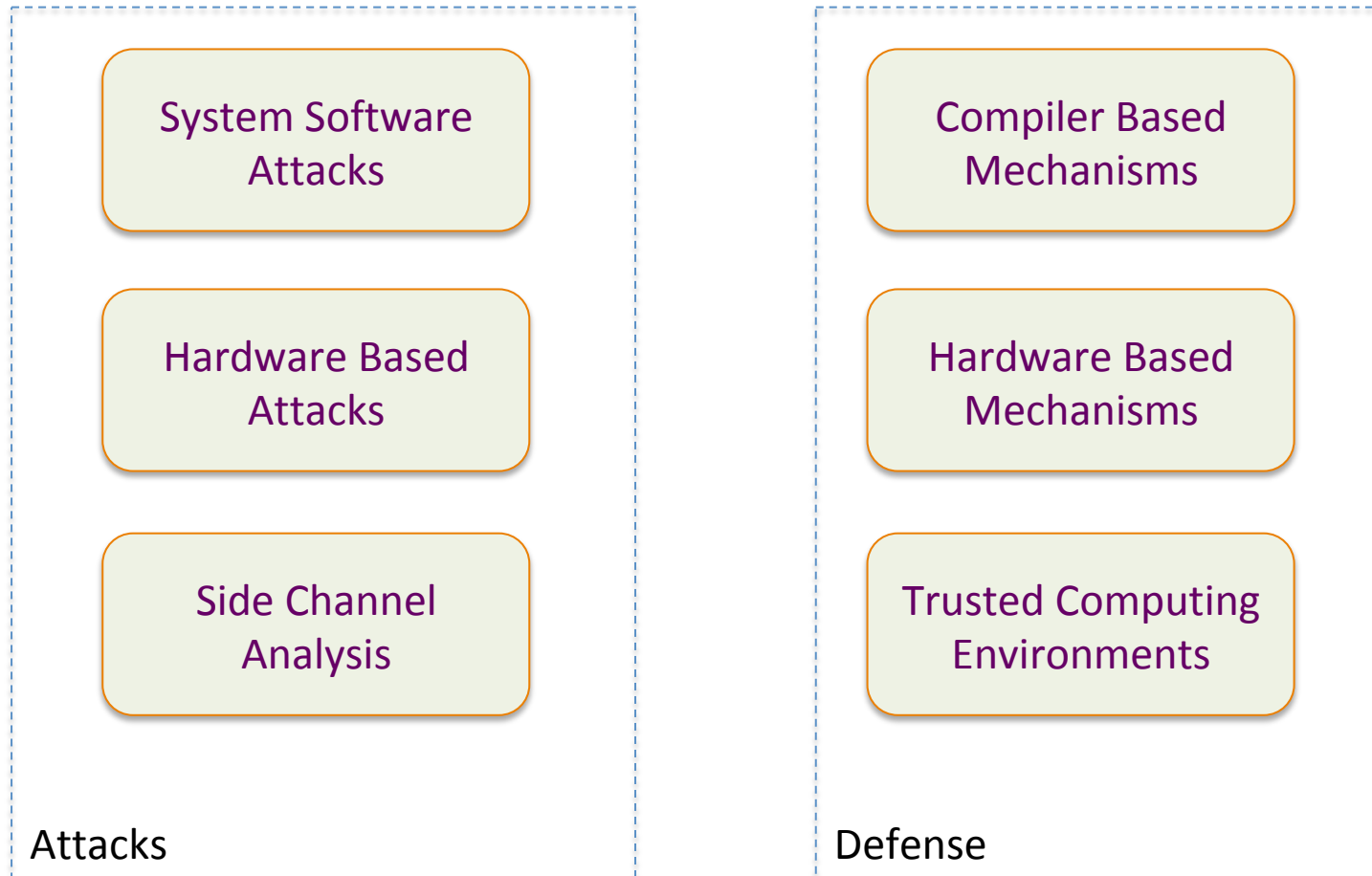


Secure Systems Engineering

Approach 3: Detect and Mitigate Attacks



Course Structure



What to expect during this course

- Deep study of systems:
 - Software
 - Assembly level
 - Compiler and OS level
 - Hardware
 - Some computer organization features
- Expected Outcomes
 - Understand the internals of malware and other security threats
 - Evaluate security measure applied at the hardware, OS, and compiler
 - Understand trade offs between performance and security

Websites and Communication

- Reference Textbooks

mostly research papers; will be provided as per topic

- For slides and programming assignments

<https://chetrebeiro@bitbucket.org/casl/sse.git>