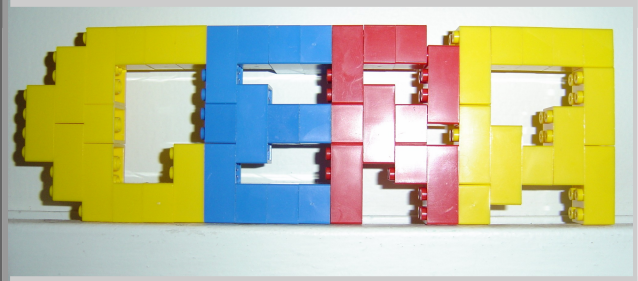
CORE SECURITY TECHNOLOGIES

CORE LABS RESEARCH & DEVELOPMENT

Two small RE tools

- Iterative Decompilation
- Python disassembling engine



Iterative decompilation

- why?
- how? – very simple
  - compile new C to DLL
  - add new import entry
  - VirtualProtect() + patch on DLL startup
- demo 1 – normal use case
- demo 2 – for instrumentation

CORE SECURITY TECHNOLOGIES

CORE LABS RESEARCH & DEVELOPMENT

Python disassembling “engine”

- why?
- how?
  - pgraph as graph backend (PAIMEI compatibility)
  - pydasm (libdasm) as disassembly backend
  - pefile for parsing PEs
  - functions and basic blocks queues
- demo 1 – breaking on every function or block
- demo 2 – graphing

Python disassembling “engine”

- debugging

```
m = module()
m.load('notepad.exe')
m.analyze()

for function in m.nodes:
    dbg.bp_set(function.ea_start)

    for basic_block in function.nodes:
        dbg.bp_set(basic_block.ea_start)
```

Python disassembling “engine”

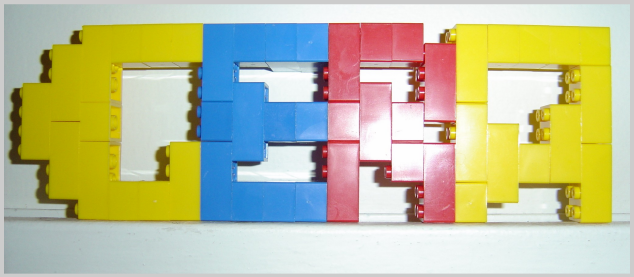
- graph
  - compared to IDA's
  - one byte – many basic blocks
- PAIMEI / PIDA

CORE SECURITY TECHNOLOGIES

CORE LABS RESEARCH & DEVELOPMENT

Questions?

- Thank you
- Thanks for ReCon
  - keep doing it
  - we love it!



@corest.com

http://oss.corest.com