Untrusted code
--system call monitoring
--inline reference monitors

External

Ref

Monitor

Untrusted code

Access

Checks

Resource

(kernel)
Monitor_check(OPEN, frame, mode);

SNEAKY:
Open(frame, mode);
Jmp %r0;
“Indirect jumps”

Q: who inserts checks?

--programmer
--compiler
--user
(but compiler and programmer are untrusted)

Advantages
--performance
--fine-grained controls

Disadvantages
--complexity
--must defend self
--self modify code
Buffer overflow, etc

Verifier
user

untrusted

trusted

programmer

code

compiler

binary

Monitor Checks
Programmer

code

compiler

binary

rewriter

trusted

user

--more complex
--slower
Monitor for memory read & writes

```
blt %r1, BASE_addr, abort
bgt %r1, TOP_addr, abort
ld %r0, [%r1]

cmp %r1, BASE_addr
blt abort
cmp %r1, TOP_addr
bgt abort
ld %r0, [%r1]

and %r1, ((1<<l)-1)
or %r1, BASE_addr
```

Untrusted data

Untrusted text

Trusted data

Trusted text

2^l size & aligned

 RO pages

NX
ld %r0,[%r1]

%r1

and

Or

%r0

Native client:

Break code into 32 byte checks, all jumps are to str to f chunks.

All monitor operators are in same chunks as checks all the instruction cannot cross chunks.

And %r0, (1< İn)-32
Or %r0, BASE_addr
Jmp %r0

%r0

and

or