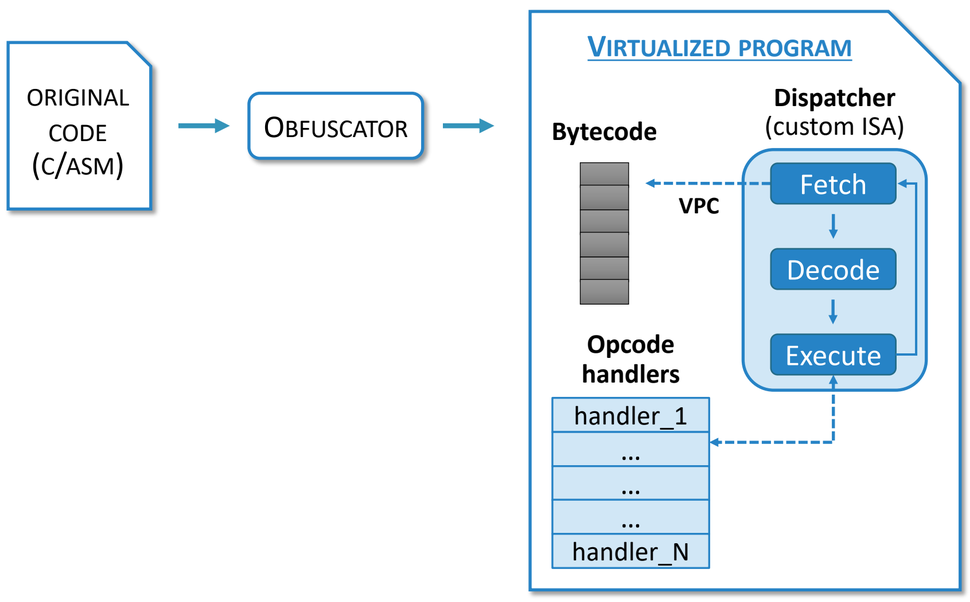
Boosting Virtualization Obfuscation with Return Oriented Programming

Pietro Borrello, Emilio Coppa, Daniele Cono D'Elia, Camil Demetrescu
Sapienza University of Rome

Virtualization Obfuscation

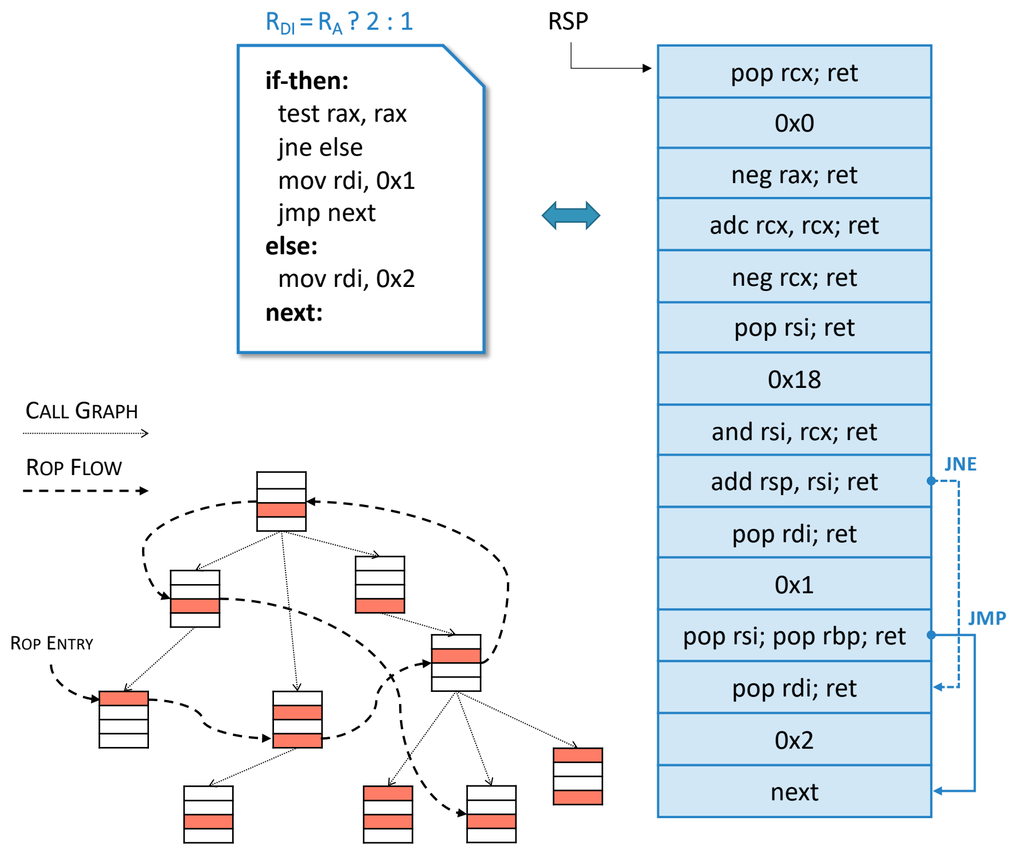


- Lengthy, obfuscated opcode handlers
- Static diversity in ISA generation
- Hide **V**irtual **PC**

Known attacks

- VPC identification [SP09]
- General-purpose deobfuscation [SP15]
- Symbolic execution [ACSAC16, DIMVA18]
- Program synthesis [USENIX17]

Return Oriented Programming



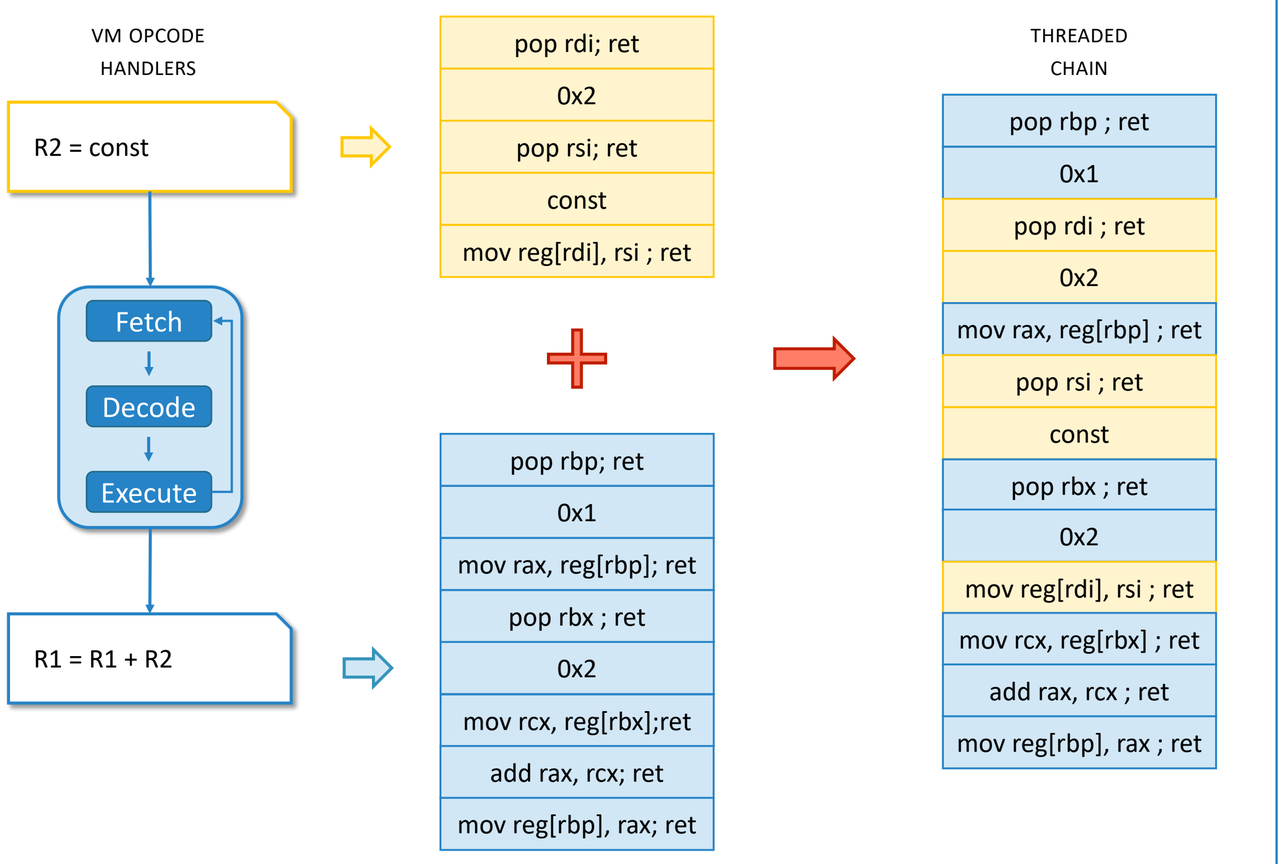
Good properties for obfuscation setting

- Control flow is destructured
- Gadget diversity & sparsity
- Same gadgets reused in different tasks
- Static analysis can be difficult

Boosting VM Obfuscation

- Opcode handlers → ROP chains
- Dispatcher → ROP program
- Combine handlers into *threaded* chains, then transform them (swap gadgets, add junk, etc.)

Prototype: Tigress+ROP



Open Questions

- How to assess the effectiveness? Automatic techniques fail already for current obfuscators!
- Measuring structural properties of chains/execution is interesting, but still not enough... Thoughts? 🙃