Symbolic Execution in radare2

Using Modality (built on angr) to perform symbolic execution, switch between concrete and symbolic execution, detect vulnerabilities and generate exploits

I'm 0xchase, a hobbyist

Introduction

radare2

Reverse engineering framework

https://github.com/radareorg/radare2

modality

Integration

https://github.com/0xchase/Modality

angr

Symbolic execution tool

https://github.com/angr/angr

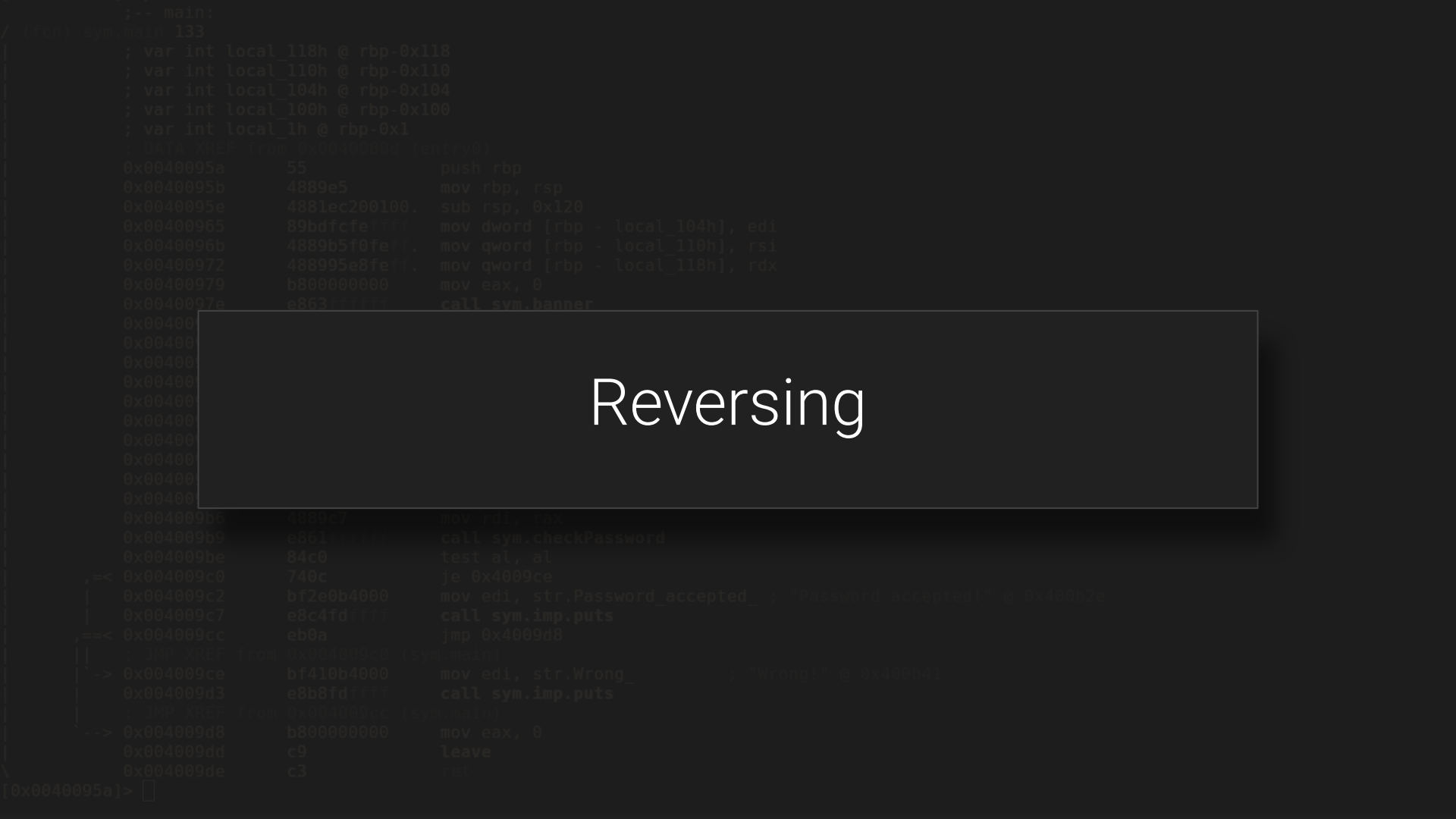
Symbolic Execution Refresher

Determine what input values are required to reach certain program states

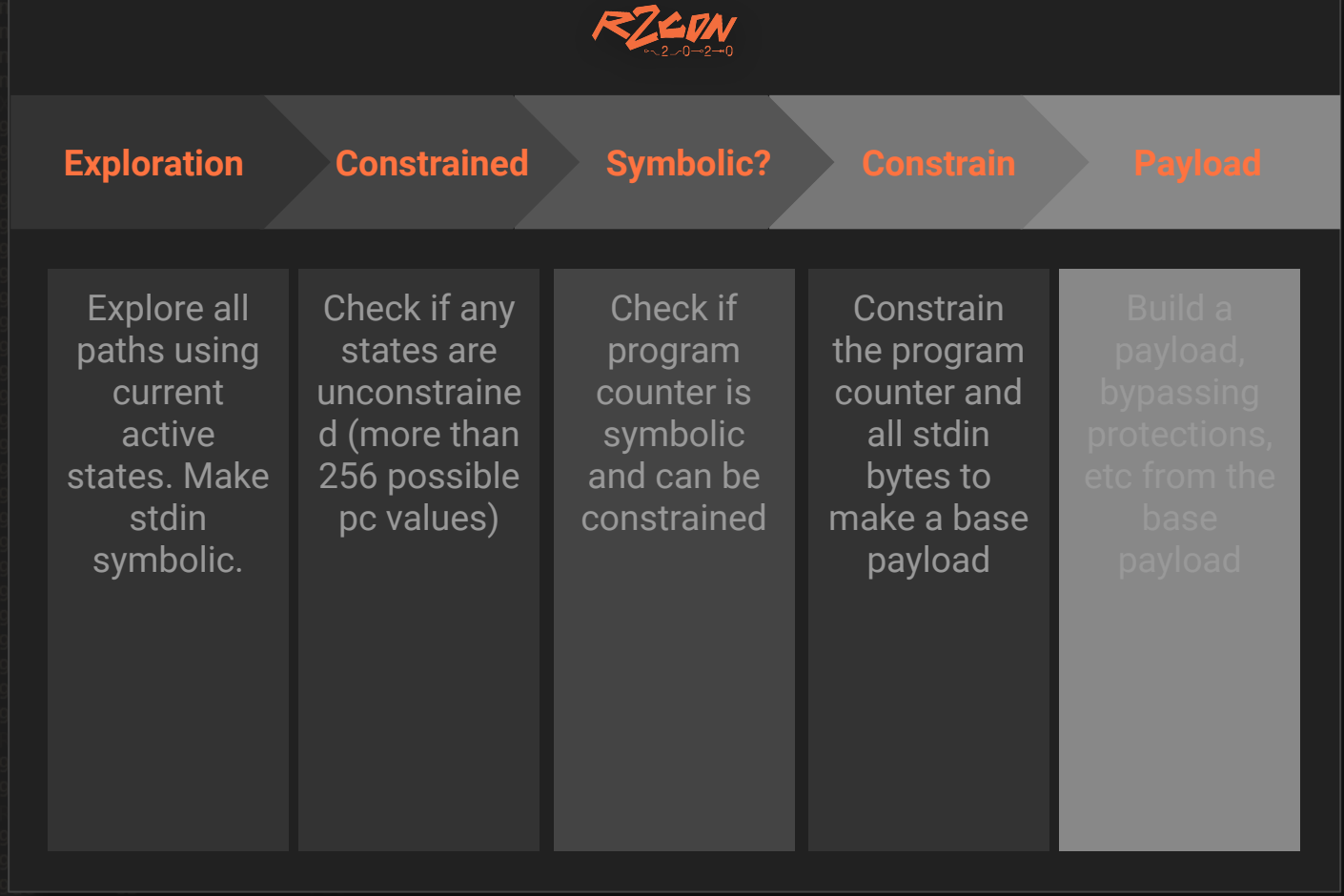
Interpreter follows the program, providing symbolic values for program input, and computing constraints for these symbolic parameters

In angr, the program is lifted to VEX IR, and the program is interpreted by the simulation manager, using what are called bitvectors as symbolic values, adding constraints which can eventually be passed to z3

Each path in the execution has an associated state, and at any point in the exploration each state is in a stash, normally the active, deadended, found, unconstrained stashes.


Reversing

Exploitation


RZCON
2020
Exploration
Constrained
Symbolic?
Constrain
Payload
Explore all paths using current active states. Make stdin symbolic.
Check if any states are unconstrained (more than 256 possible pc values)
Check if program counter is symbolic and can be constrained
Constrain the program counter and all stdin bytes to make a base payload
Build a payload, bypassing protections, etc from the base payload

Review

Solved some reversing challenges with symbolic execution

Showed various state initialization techniques

PC overwrite vulnerability discovery

Basic exploit generation

A few ASLR/DEP bypasses

RZCON 2020

Looking Forward

Integrate analyses

Integrate symbion

Track heap usage

More ASLR/DEP bypass techniques

Bug fixes

And more

The End