Guide Me To Exploit: Assisted ROP Exploit Generation for ActionScript Virtual Machine

Fadi Yilmaz¹, Meera Sridhar², Wontae Choi³
¹fadiyilmaz@ybu.edu.tr, ²msridhar@uncc.edu, ³wtchoi.kr@gmail.com

Motivation
Monitoring the execution of exploit scripts is crucial
Writing exploit for each vulnerability requires intense human-effort
Discovering unorthodox methods to exploit vulnerabilities
Needling a tool that produces exploit scripts in an autonomous fashion

Methodology
Automated Exploit Generation
The challenge of determining the exploitability of given vulnerability [1]
Explore all execution paths that follow the vulnerability state [2]
Hope one of the execution paths reaches one of exploit states

Intuition Behind Target Exploit Generation
The exploit pattern
Malicious activity must be performed under the radar
Deconstructing the exploit scripts into smaller subgoals
Subgoals are defined by the security experts

Exploit Subgoals
Consist of a search space and an invariant
Set of instructions that can contribute to achieve the corresponding exploit subgoal
The test that checks whether the corresponding exploit subgoal is achieved

Methodology – Cont.
Testing synthesized executables with the given invariant
Focusing on synthesizing executables for the next exploit subgoal after the current exploit subgoal is achieved
Stitching up executables that achieve an exploit subgoal to obtain the final exploit script

Optimization Techniques
Exploit Deconstruction
Allows us to define exploit subgoals
Enables us to solve same problem with smaller instances (~10^4)
Stack Simulation
Disqualifies candidate slices that perform illegal stack operation (98.78%)
Instruction Tiling
Creates bigger and more meaningful instruction chains to be used (~10^14)
Feedback from the AVM
Allows us to detect error-raising instruction sequences and disqualifies candidate slices that contain such instruction sequence (58%)

Experimental Results
Our tool generates the exploit script in less than 15 minutes
Our tool exploits a use-after-free vulnerability CVE-2015-5119, which is one of the most commonly exploited vulnerabilities in 2016
Our tool synthesized more than 650 million executables to reach the target exploit state

Future Work
Conducting a user study to measure
• How much human interaction is required
• What level of human expertise is required
• How much effort does our tool save for a seasoned developer

Key References