POSTER: Quantifying the Impact of Fine-grained Code Randomization on Attack Surface Reduction

Md Salman Ahmed[1], Ya Xiao[1], Gang Tan[2], Kevin Snow[3], Fabian Monrose[4], Danfeng (Daphne) Yao[1]
[1]Computer Science, Virginia Tech, [2]Computer Science & Eng., Penn State, [3]Zeropoint Dynamics, LLC, [4]Computer Science, UNC at Chapel Hill
{ahmedms, yax99, danfeng}@vt.edu, gtan@cse.psu.edu, kevin@zeropointdynamics.com, fabian@cs.unc.edu

1. Motivation

Quantitative evaluation of how fine-grained ASLR or code randomization impacts code reuse attacks is necessary.

Methodologies for security metrics such as gadget reduction, upper bound for re-randomization intervals, etc. are necessary.

2. Challenges

How to quantify the impact of fine-grained ASLR

How to quantify the difficulty of accessing system functions

How to quantify the quality of a gadget chain

3. Approach

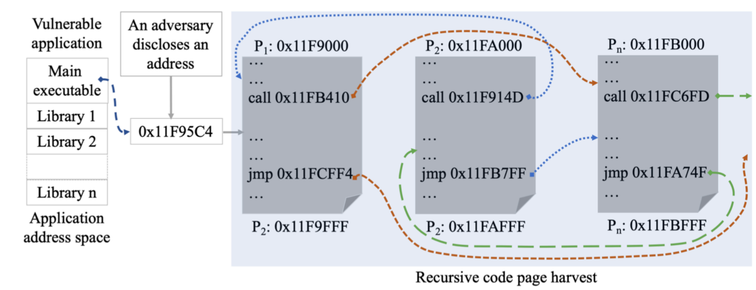
- ❑ We measure **JIT-ROP gadgets** using **5** tools, **13** applications, and **19** libraries.
- ❑ We **quantify the importance of a code pointer leak** using JIT-ROP gadgets.
- ❑ We measure the gadget quality using **register corruption rate**.
- ❑ We provide a **methodology** for determining the **re-randomization interval**.

4. Experimental Design

(Re-)randomization tools

Zipr[1] | Selfrando[2] | CCR[3] | Multicompiler[4] | Shuffler

Identification of Turing-complete gadgets

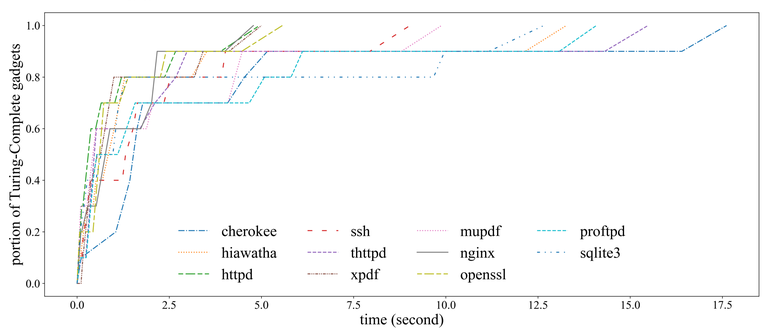


5. Evaluation

Gadget reduction by FG tools

Tools	Reduction
Zipr	**80%~90%**
Selfrando	23%~40%
CCR	10%~27%
Multicompiler	6%~37%
Shuffler	~24%

upper bound of re-randomization intervals



6. Conclusion

We presented multiple general methodologies for quantitatively measuring the ASLR security under the JIT-ROP threat model and conducted a comprehensive measurement study. Major findings include i) instruction-level randomization has the most impact on gadget reduction, ii) equal viability of pointer leaks, and iii) insignificant impact on gadget quality. Besides, upper bound helps guide re-randomization adopters to make more informed configuration decisions.

[1]https://git.zephyr-software.com/opensrc/irdb-cookbook-examples
[2]https://github.com/immunant/selfrando
[3]https://github.com/kevinkoo001/CCR
[4]https://github.com/securesystemslab/multicompiler

This work is supported by ONR Grant N00014-17-1-2498 and DARPA/ONR N66001-17-C-4052.

VIRGINIA TECH

Yao Group on Cyber Security
http://yaogroup.cs.vt.edu/

IEEE SecDev 2019