Inspecting Virtual Machine Diversification Inside Virtualization Obfuscation

Naiqian Zhang*; Dongpeng Xu*; Jiang Ming†; Jun Xu‡; Qiaoyan Yu*

*University of New Hampshire, USA; †Tulane University, USA; ‡University of Utah, USA

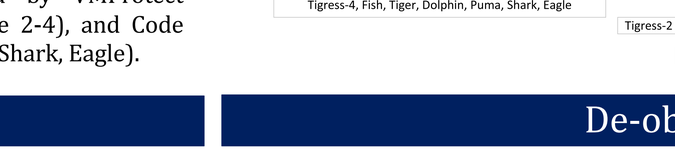
IEEE S&P

Introduction

- Virtualization obfuscators are commonly employed to safeguard proprietary code or to impede malware analysis.
- Core of modern virtualization obfuscators are the virtual machines (VMs), which employ a variety of diversification techniques to complicate their structures.
- No systematic study of VMs diversification techniques to reveal the technical details behind each of them, creating a knowledge gap that needs to be addressed to enhance the VM de-obfuscation side.
- Our work aims to bridge the gap with unveiling the techniques under the hood of VMs, patching current de-obfuscation tools, and paving the way for next-generation VM de-obfuscation.

Virtualization Obfuscation

- Semantic-preserving transformation on selected code part.
- Transform code to bytecode format in a customized virtual instruction set architecture (ISA).
- Emulate the bytecode by an embedded VM at runtime.



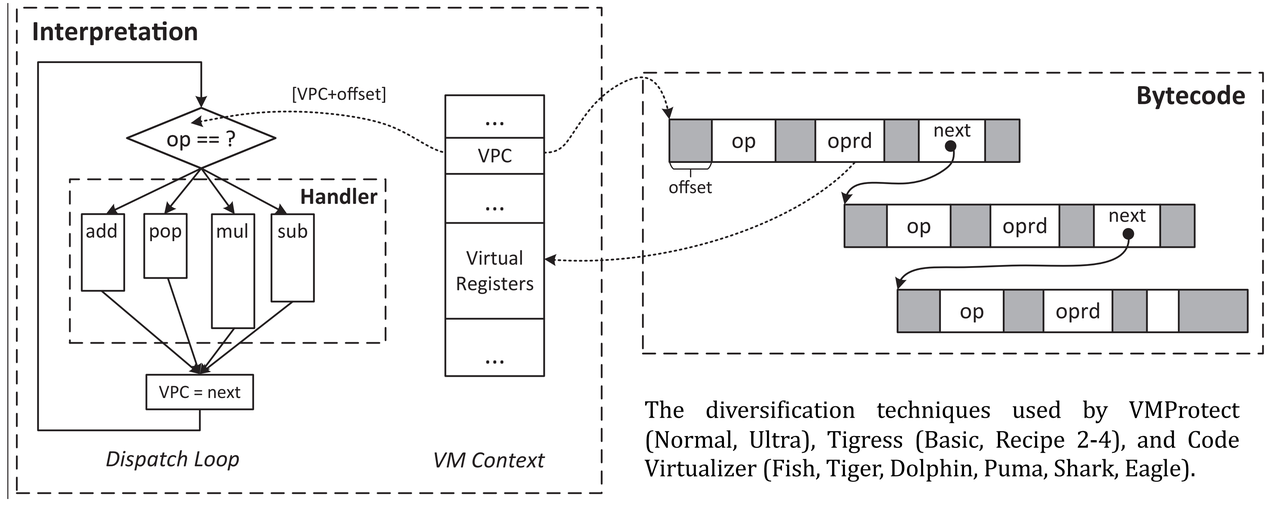
https://www.oreans.com/CodeVirtualizer.php

Contributions

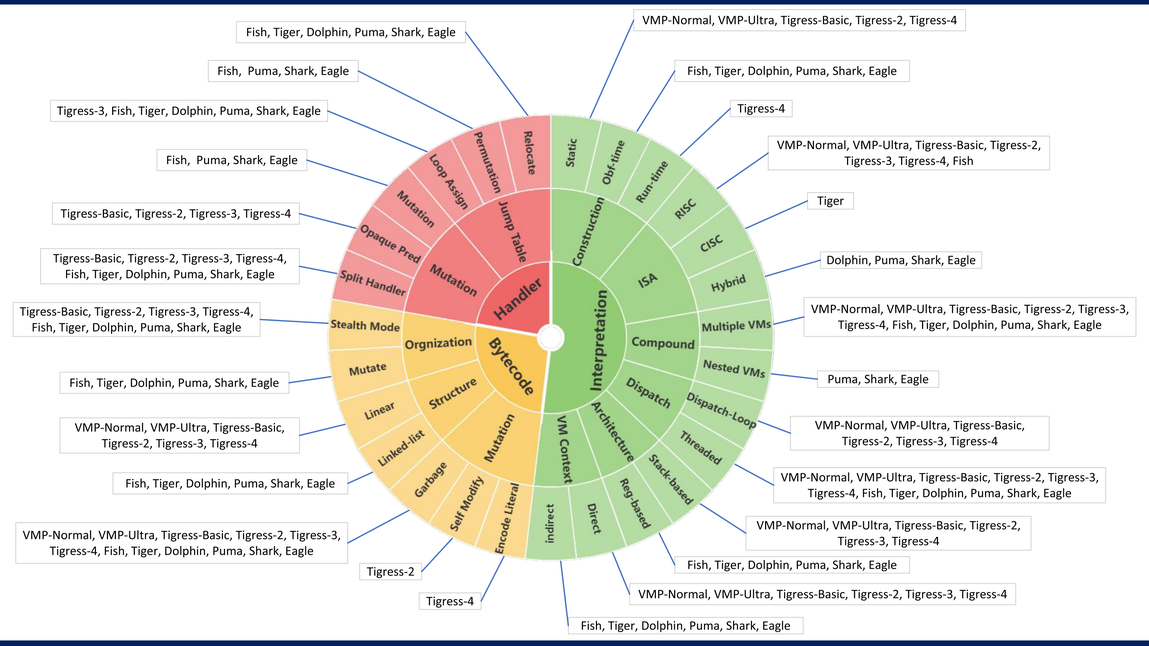
- An in-depth taxonomy of the existing VM diversification techniques, bridging a knowledge gap in current understanding of virtualization.
- A new research tool, VM-Doctor, to detect and extract the VM diversification features from 12 VM variants of 3 state-of-the-art virtualization obfuscators.
- Enhancement and improvement of 3 existing de-obfuscation methods in their precision and performance by applying the VM diversification knowledge we extracted and summarized.

VM Diversification Features

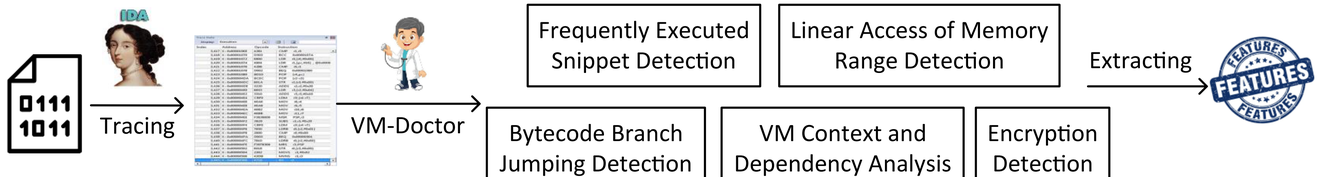
The architecture of a normal VM showing the interpretation, bytecode, and handler levels. The interpretation has a dispatch-loop The bytecode is a linked-list, where each node is one bytecode instruction. The virtual Program Counter (VPC) always points to the start of the current bytecode instruction. The dashed arrows represent possible memory access.



The diversification techniques used by VMProtect (Normal, Ultra), Tigress (Basic, Recipe 2-4), and Code Virtualizer (Fish, Tiger, Dolphin, Puma, Shark, Eagle).



VM-Doctor



A set of analysis tools to extract the VM components based on dynamic trace recording and static disassembly output, contains of Java, Python, and C++ code, build on Intel PinTool, IDAPro.

De-obfuscation Enhancement

Virtual-Dobfuscator		Fish	Tiger	Dolphin
	FNR (Before/After)	72.70% / 0.00%	75.86% / 0.00%	63.04% / 0.00%
	FPR (Before/After)	85.02% / 40.04%	91.67% / 39.02%	93.54% / 76.31%
VMHunt		Puma	Shark	Eagle
	# of VM Snippet (Before/After)	51 / 34	49 / 43	46 / 22
	# of Instruction (Before/After)	3,764,280 / 2,705,520	1,464,806 / 1,285,442	4,240,004 / 2,027,828
Syntia		Tiger	Dolphin	Puma
	# of Trace Window (Before/After)	340 / 290	1,385 / 1,321	2,427 / 2,271
	Synthesis Time(s) (Before/After)	3,808 / 3,190	15,927 / 14,795	27,182 / 24,981