2022

High-Fidelity Virtual Machine Artifact Mitigation

Singh, Gurminder

Monterey, California: Naval Postgraduate School

NAVAL RESEARCH PROGRAM
NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

HIGH-FIDELITY VIRTUAL MACHINE ARTIFACT MITIGATION

EXECUTIVE SUMMARY

Principal Investigator (PI): Dr. Gurminder Singh, Computer Science

Additional Researcher(s): Dr. Alan Shaffer, Information Sciences

Student Participation: Mr. Charles Prince, CIV, Computer Science and LT Eric Chamberlin, USN, Computer Science

Prepared for:
Topic Sponsor Lead Organization: HQMC Information (DCI)
Topic Sponsor Organization(s): MARFORCYBER
Topic Sponsor Name(s): MAJ Audrey Callanan, USMC
Topic Sponsor Contact Information: email: afcalla@nsa.gov, phone: 667.812.8521

This research is supported by funding from the
Naval Postgraduate School, Naval Research Program (PE 0605853N/2098).

Project Summary

The use of virtualized systems has grown across the application domains that include cyber operator training and offensive and defensive cyber operations. Using virtualized systems is, however, not without its risks, especially if an adversary can determine whether or not a host is virtualized. To prevent such detection, the fidelity of the hypervisor needs to be extended so that adversaries cannot distinguish between a virtualized or a real system. This project is a continuation of our previous work in high-fidelity virtualization (HFV) and HFV artifact mitigation (HFVAM). Previously, we used the Xen hypervisor and DRAKVUF to obfuscate an executable such that the adversary would not know that the system was virtualized. This new work concentrated on the use of a new DRAKVUF capability—process injection by execution—for mitigating HFV artifacts. Our hypothesis was that process injection would lead to even better mitigation of HFV artifacts. We found that process injection was not suitable at this time for mitigating virtualization artifacts, and that DRAKVUF process injection did not work, although the code reported that it had. Thesis research is on-going, including coordination with DRAKVUF developers on the contradictory results that we observed, and updated results should be available by the end of 2022 (Prince, 2022). This project also explored methods for obfuscating the operating system (OS) and libraries to mitigate attack vectors of intelligent malware.

Keywords: *high-fidelity virtualization, HFV, HFV artifact mitigation, HFVAM, Xen, DRAKVUF, process injection, virtual machine, Linux, cloud resilience, web resilience, insider threats*

Background

In recent years, the use of virtualized systems has grown tremendously to include application domains that were originally not envisioned to use virtualized systems. These domains include cyber operator training and offensive and defensive cyber operations. But these domains require certain capabilities in hosts that are beyond what current state-of-the-art hypervisors can support. Additionally, virtualized systems are more vulnerable to attacks when an adversary can determine that the host is virtualized. To support the requirements of these cyber applications, the hypervisor needs to be extended with advanced capabilities, leading to HFV. To prevent the detection of virtualized systems, we need to mitigate artifacts of virtualization. This work and area of study will be beneficial to cyber operations, as well as cloud resiliency and web resiliency, and may expose and perhaps prevent external and insider attacks.

The objective of this project was to make an attacker believe that a system is not a virtual machine, when in reality that system is running as a virtual machine. The reasons to do this are various. An intelligent malware munition (or in general, an attacker) may detect what kind of Linux system is running and choose an attack vector based on what the malware detects. If the malware misdiagnoses the system, then the attack may fail and lead to exposing that malware to system defenders.

This work is a continuation of previous work (Norine, 2020) that used DRAKVUF to relabel the path of an executable process to that of another executable in order to deceive the attacker. This relabeling technique provided results to lead the attacker into believing that they had gained access onto a non-

virtualized "bare metal" system. This work tried to extend the previous work by injecting the executable with spoofed information in such a way that the attacker could not determine what had happened.

Accomplishing this mitigation is difficult in that it requires direct manipulation of the Linux kernel and the corresponding data structures. The Linux kernel has become progressively harder to decompose, or even to determine what happens in a system call. Another layer of difficulty occurs when considering shellcode.

Findings and Conclusions

This research found that the DRAKVUF process-injection using execution (DPIE) was not currently suitable for HFVAM. DPIE could not affect standard output, nor could it overwrite an output file, even though the module believed that the injection had taken place. However, the DRAKVUF process-injection using shellcode (DPIS) may be able to support HFVAM. The DPIS is based on *vdso*, which is a kernel function and not a system call, making it more difficult for an attacker to recognize that HFVAM is taking place. In addition, this research revealed that HFVAM of OS and library versions may well lead smart malware munitions and attackers to be fooled in exposing themselves, thus preventing attacks.

The short-term implications of this research indicate that process injection is not suitable at this time for mitigating virtualization artifacts. However, our findings showed that it may be used for deceiving malware's perception of the OS and library versions, which suggest that in the long-term, continued research is warranted despite the null hypothesis observed in DRAKVUF process injection.

Recommendations for Further Research

The DPIS method appears to be a good candidate for use in high-fidelity virtual machine artifact mitigation (HFVAM) because that module's method is based on *vdso*, a hard-to-track kernel function. DPIS could be designed to jump to an area of code where it can run another executable, outputting the results to standard output (i.e., the display) or to an external file. Another area of further research is using HFVAM for operating system and library versions as a way to extend resilience for smart malware munitions as well as attackers. Another area of future research should be micro-kernel extensions to Xen in order to provide greater resilience to the hypervisor itself.

References

Norine, C. R. (2020). *Artifact mitigation in high-fidelity hypervisors* [Unpublished master's thesis, Naval Postgraduate School].

Prince, C. D. (2022). *High-fidelity virtual machine artifact mitigation using DRAKVUF* [Unpublished master's thesis, Naval Postgraduate School].

Acronyms

HFV	high-fidelity virtualization
HFVAM	high-fidelity virtualization artifact mitigation
DPIE	DRAKVUF process-injection using execution
DPIS	DRAKVUF process-injection using shellcode
OS	operating system