KLogic: Runtime monitoring for Windows Kernel

Daejin Lee
KAIST
Daejeon, Republic of Korea
djlee1592@kaist.ac.kr

ABSTRACT

Kernel Vulnerabilities are crucial in OS. It can cause DoS the system or executing arbitrary code as system. This will result in a whole system compromise.

There has been researched about kernel research. Most of the approaches are related to memory corruption vulnerabilities(eg. Buffer Overflow, Heap Overflow, etc). However, this approach can not reveal the kernel's logic vulnerability which is more reliable to exploit and trigger than memory corruption vulnerabilities. This paper presents run-time monitoring tool for kernel logic bugs called KLogic especially in File Input-Output(IO), Registry.

CCS CONCEPTS

• **Security and privacy** → *Software security engineering*.

KEYWORDS

kernel vulnerability, run-time monitoring

ACM Reference Format:
Daejin Lee. 2020. KLogic: Runtime monitoring for Windows Kernel. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 1 page. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Logic vulnerabilities are high impact to kernel system. Especially in Windows OS, there have been serveral logic bugs are exploited in the wild and disclosed. KLogic focused on two types of bug, File IO and Registry. More specifically, it finds Arbitrary (File, Registry) Write, Arbitrary (Directory, File, Registry Key) Create, Arbitrary (Directory, File, Registry) Set permission, Arbitrary (Directory, File, Registry) Delete.

The KLogic consist of two parts, Hooking module and Permission check. First one, Hooking module, hooks windows system call in OS level to monitor the entire API usages. For example, if we want to find File IO vulnerabilities, we should consider the file related API(NtCreateFile, NtWriteFile, NtReadFile, etc). Second one, Permission check, checks some vulnerable cases. 1. If the caller process is SYSTEM integrity. 2. Whether the called API contains the path that guest user can access and the parent directory of it can also be accessed. If these two conditions are matched, we can assume that the argument path is vulnerable to the guest user.

2 RELATED WORK

The following items are some cases of the logic vulnerabilities.

- `Directory Create to System`: By arbitrary creating a directory in a certain path, we can leverage the bug to get the system privilege via system services(eg. WERFault Service). To achieve this purpose, the directory we made should be accessible to guest user.
- `Directory Delete to System`: By arbitrary deleting a directory in a certain path, we can leverage the bug to get the system privilege via system services(eg. WER Service).
- `File/Registry Write to System`: By arbitrary writing a file/registry in a certain path, we can leverage the bug to the the system privilege via system service(eg. USO Service)
- `SecurityInfo Overwrite to System`: By arbitrary overwriting a security descriptor in a certain file/directory, we can leverage the bug to the system privilege via system service

3 DISCUSSION

Even if we catch the detailed of a vulnerability, we can not trigger that vulnerability directly. Because the KLogic can only detect the case when API called.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM...$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn