WOOT'25 Artifact Evaluation BOOTKITTY: A Stealthy Bootkit-Rootkit Against Modern Operating Systems

Junho Lee
Mokpo National University

Jihoon Kwon
Korea University

HyunA Seo
Sungshin Women's University

Myeongyeol Lee
Chosun University

Hyungyu Seo
Keimyung University

Jinho Jung
Ministry of National Defense

Hyungjoon Koo
Sungkyunkwan University

1 Overview

1.1 Abstract

The following appendix describes the procedures to reproduce the BOOTKITTY attack scenario and validate its functional results. This artifact includes preconfigured QEMU virtual disk images or Virtual Machine (VM) images for Windows and Linux containing BOOTKITTY. The Android component used in the study relies on specific physical hardware and therefore cannot be reproduced in virtualization or emulator environments.

1.2 Description & Requirements

1.2.1 Security & Privacy

This artifact does not pose any security risk (breaches) to the evaluator's host system. All BOOTKITTY binary executes exclusively within the provided QEMU virtual disk images or VM images, and no malicious code runs outside of these VMs on the host machine.

1.2.2 How to access

Our samples support the following two environments

- **VM**: VM images can be downloaded from the Zenodo repository (https://zenodo.org/records/15501870)
- **QEMU**: QEMU virtual disk images can be downloaded from the Zenodo repository (https://zenodo.org/records/15582744)

These VMs include the bootkit, rootkit files, and scripts for installation and removal, and the QEMU also include execution scripts.

1.2.3 Hardware dependencies

- **Windows, Linux**: you need at least **8 GB RAM** and **50 GB free disk space** to download and run the images.
- **Android**: Due to the dependency on the specific physical hardware, BOOTKITTY for Android cannot be reproduced in a virtualization or emulator environments.

1.2.4 Software dependencies

We provide VM images for artifact evaluation, which require either VMware or QEMU to run.

If you're using QEMU, you must install the dependencies by running the following command

```
sudo apt install swtpm
sudo apt install qemu-kvm libvirt-daemon-system \
libvirt-clients bridge-utils virtinst virt-manager
```

1.2.5 Tested Environments

- **VMware VM:** We was tested on Windows 24H2 using the latest VMware Workstation.
- **QEMU virtual disk:** We was tested on Ubuntu 24.04 LTS.

2 Demonstration Setup

2.1 Importing Virtual Machines

After extracting each of the provided Windows and Linux VM archives

- **VMware VM image:** Open VMware Workstation's Scan for Virtual Machines Wizard to detect and import each VM.
- **QEMU image:** simply run the ./run.sh script in the Windows or Linux folder.

2.2 Credentials

Table 1 shows the credentials for VMs. (The Windows Encryption Password is required only when using the VMware image.)

VM Type	Credential	Value
Windows VM	Encryption password	qwer1234
Linux VM	Username	user
Linux VM	Password	1234

Table 1: VM Credentials

3 Evaluation Goals

The primary evaluation goal of this artifact is to verify that BootKitty is correctly installed and its rootkit component is successfully loaded and concealed within the kernel, using the provided scripts.

- **Windows:** After executing the `install-bootkit` script, the BootKitty logo should appear on reboot, and the file `C:\rootkit.sys` must not be visible in File Explorer or when running the `dir` command.

- **Linux:** After executing the `install-bootkit` script, the hostname should be changed to **BoB13** as shown by `uname -a`, and no rootkit-related files under `/opt/` should be listed by the `ls` command.

3.1 Experiments

3.1.1 Windows

To verify the bootkit functionality on Windows, follow the example workflow.

1. Execute the `install-bootkit.bat` script located on the desktop with administrator privileges.

2. Reboot the system.

3. If the BootKitty logo (Figure 1) appears during startup, the bootkit has been successfully loaded and the rootkit installation is in progress.

4. To verify that the rootkit is functioning correctly, inspect the `C:\` directory; if `rootkit.sys` is *not* visible, the rootkit is operating properly.

5. Execute the `remove-bootkit.bat` script located on the desktop with administrator privileges.

6. Reboot the system again; if the BootKitty logo no longer appears and `rootkit.sys` is visible in the `C:\` directory, this step confirms that the bootkit's concealment mechanism worked correctly—while loaded, it hid `rootkit.sys`, and only upon removal does the file reappear, verifying its proper operation.

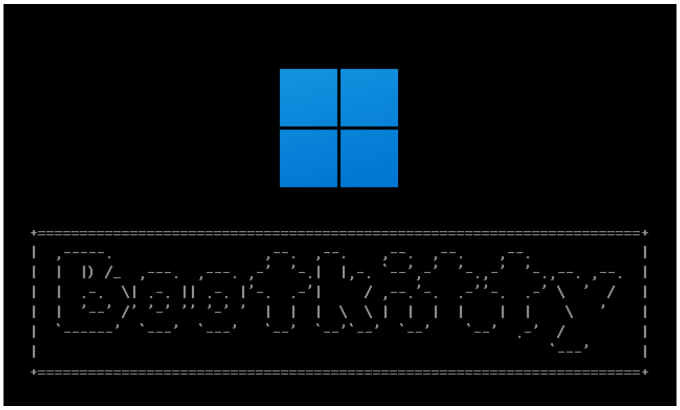


Figure 1: Windows BootKitty Logo

3.1.2 Linux

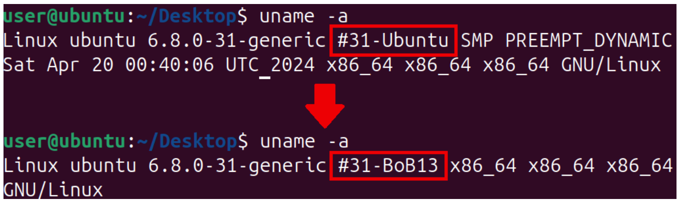
To verify the bootkit functionality on Linux, follow the example workflow.

1. Verify that the rootkit components are present under `/opt`.

2. Execute the `install-bootkit.sh` script located on the desktop with root privileges (e.g., using `sudo`).

3. Reboot the system.

4. An error window may appear during startup; this can be safely ignored.



```
user@ubuntu:~/Desktop$ uname -a
Linux ubuntu 6.8.0-31-generic #31-Ubuntu SMP PREEMPT_DYNAMIC
Sat Apr 20 00:40:06 UTC 2024 x86_64 x86_64 x86_64 GNU/Linux

user@ubuntu:~/Desktop$ uname -a
Linux ubuntu 6.8.0-31-generic #31-BoB13 x86_64 x86_64 x86_64
GNU/Linux
```

Figure 2: changed Linux hostname by BootKitty

5. Execute the `uname -a` command to verify that the hostname has been changed to **BoB13** (Figure 2) by the bootkit.

6. Check the `/opt` directory again. If the files initially present are no longer visible, the rootkit is functioning correctly. (Due to the rootkit's instability, its successful operation is probabilistic.)

7. If the rootkit did not run, reboot the system again.