WRITE BLOCKER IN DIGITAL FORENSICS

Jianjun "Jeffrey" Zheng

05/30/2018

WHAT IS WRITE BLOCKER

- DEVICES
  - ALLOW INFORMATION TO BE RETRIEVED FROM A STORAGE DEVICE WITHOUT CREATING POSSIBILITY OF ACCIDENTALLY DAMAGING THE DEVICE CONTENT
  - READ-ONLY ACCESS

MOTIVATION

- Duplicate a drive
  - Create a forensic copy (imaging)
  - Preserve the original copy
- Mount response drive
  - Antivirus deletion
  - Malware infection

TYPES OF WRITE BLOCKER

- Hardware Write Blocker
  - Expensive and Powerful
  - Stand-alone controller chip with writeblocking software
  - Can be used on multiple computers
- Software Write Blocker
  - Easy-to-use and Less Expensive
  - Built in the OS or Installed on the computer
  - Work on a single computer

IMPLEMENTATION OF SOFTWARE WRITE BLOCKER IN LINUX

- Linux Kernel Patch
  - Open Source Project

    https://github.com/msuhanov/Linux-write-blocker
- Udev Rules
  - Use Linux Udev rules to mount external devices as read-only

UDEV BASICS

- Linux Dynamic Device Management
- Dynamically identify devices based on their properties
- Composed of some kernel services and the *udevd* daemon
- Kernel informs the udevd daemon when certain events happen
- The daemon responds to events with corresponding actions
- The response of the udevd daemon is specified by Udev rules

UDEV BASICS – CONT'D

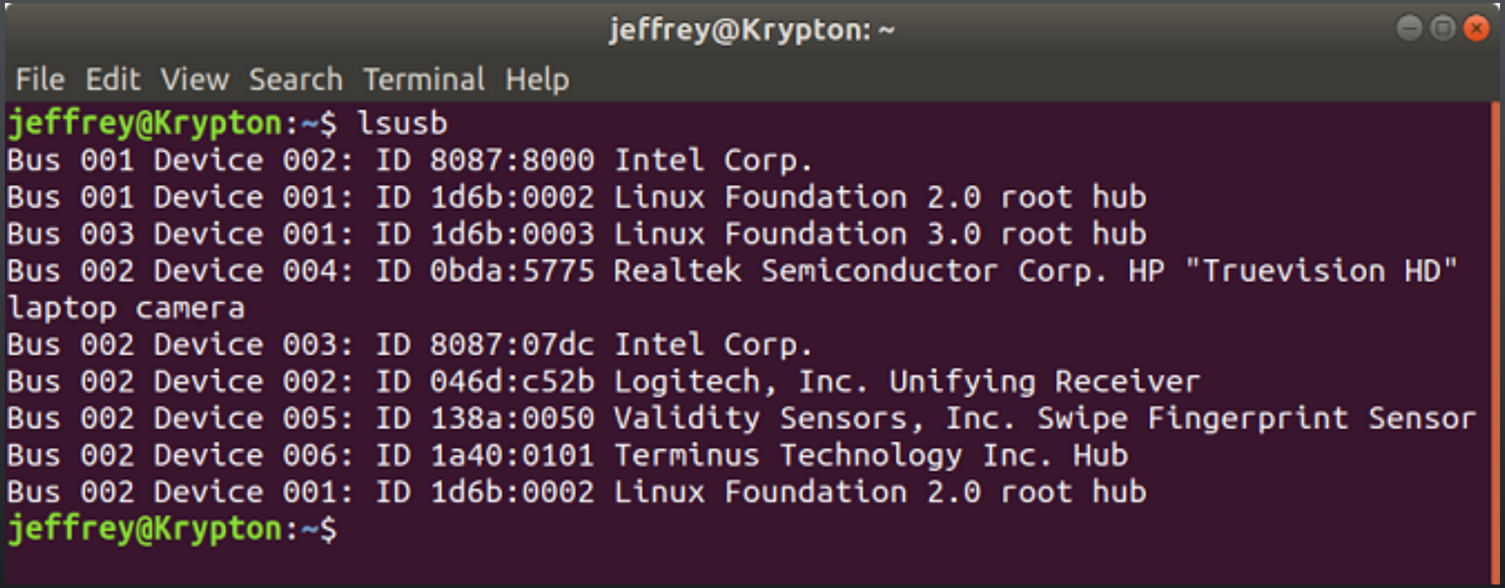
- Udev Rules
  - Matching criteria and actions
  - Four commonly used matching categories:
    - KERNEL, SUBSYSTEM, DRIVER, ATTR
  - Start with matching by using "==" or "!="
  - Use "=" to create a new item and use "+=" to add an item to an existing item

UDEV BASICS – CONT'D

- Udev Rules
  - Set variables in "ENV" space in earlier rules and refer to them with later rules
  - Rule file has extension *.rules*
  - Rule files are saved in /etc/udev/rule.d/ folder
  - Earlier rules have precedence over later rules
    - 10-this-rule-runs-earlier.rules, 20-this-rule-runs-later.rules

UDEV BASICS – CONT'D

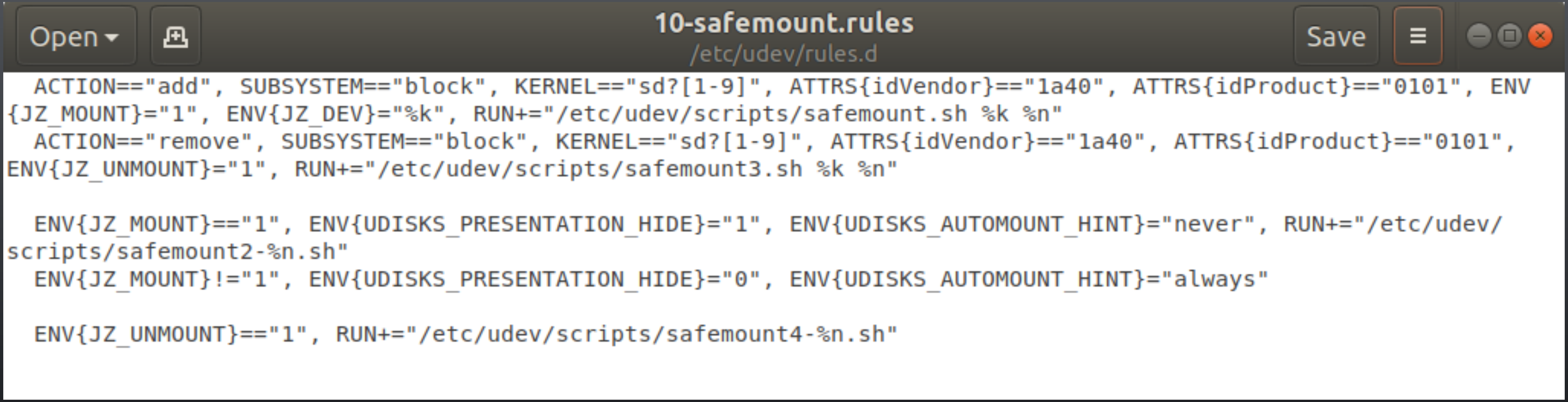
- Udev Rules Example:
  - Executing a shell script when a USB hub is inserted
    - Identify the vendor ID and product ID of the hub with "lsusb"



```
jeffrey@Krypton:~$ lsusb
Bus 001 Device 002: ID 8087:8000 Intel Corp.
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 003 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 002 Device 004: ID 0bda:5775 Realtek Semiconductor Corp. HP "Truevision HD"
laptop camera
Bus 002 Device 003: ID 8087:07dc Intel Corp.
Bus 002 Device 002: ID 046d:c52b Logitech, Inc. Unifying Receiver
Bus 002 Device 005: ID 138a:0050 Validity Sensors, Inc. Swipe Fingerprint Sensor
Bus 002 Device 006: ID 1a40:0101 Terminus Technology Inc. Hub
Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
jeffrey@Krypton:~$
```

UDEV BASICS – CONT'D

- Udev Rules Example:
  - <u>Executing a shell script when a USB hub is inserted</u>



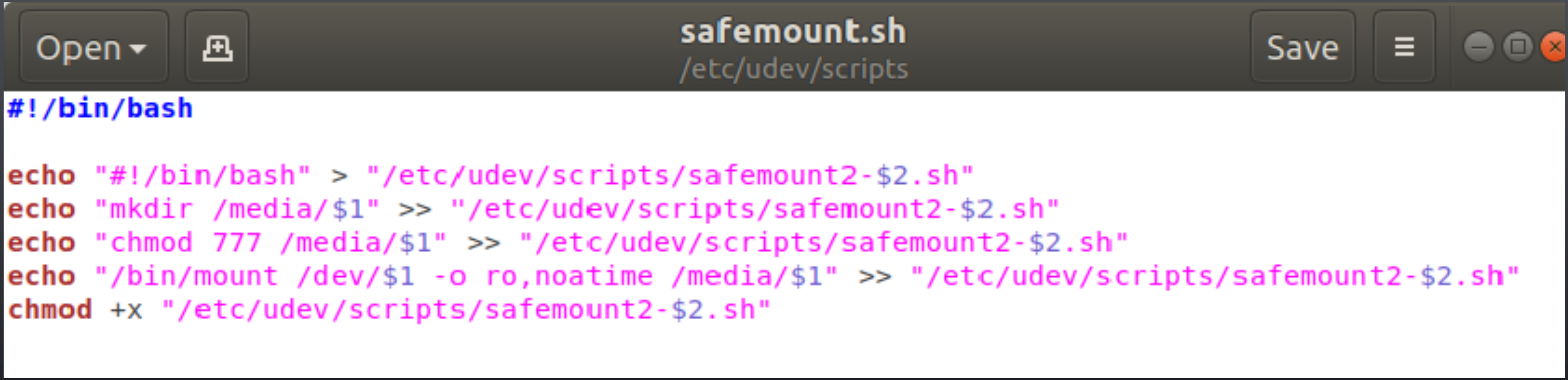
```
  ACTION=="add", SUBSYSTEM=="block", KERNEL=="sd?[1-9]", ATTRS{idVendor}=="1a40", ATTRS{idProduct}=="0101", ENV
{JZ_MOUNT}="1", ENV{JZ_DEV}="%k", RUN+="/etc/udev/scripts/safemount.sh %k %n"
  ACTION=="remove", SUBSYSTEM=="block", KERNEL=="sd?[1-9]", ATTRS{idVendor}=="1a40", ATTRS{idProduct}=="0101",
ENV{JZ_UNMOUNT}="1", RUN+="/etc/udev/scripts/safemount3.sh %k %n"

  ENV{JZ_MOUNT}=="1", ENV{UDISKS_PRESENTATION_HIDE}="1", ENV{UDISKS_AUTOMOUNT_HINT}="never", RUN+="/etc/udev/
scripts/safemount2-%n.sh"
  ENV{JZ_MOUNT}!="1", ENV{UDISKS_PRESENTATION_HIDE}="0", ENV{UDISKS_AUTOMOUNT_HINT}="always"

  ENV{JZ_UNMOUNT}=="1", RUN+="/etc/udev/scripts/safemount4-%n.sh"
```

UDEV BASICS – CONT'D

- Udev Rules Example:
  - Executing a shell script when a USB hub is inserted
    - Safemount.sh

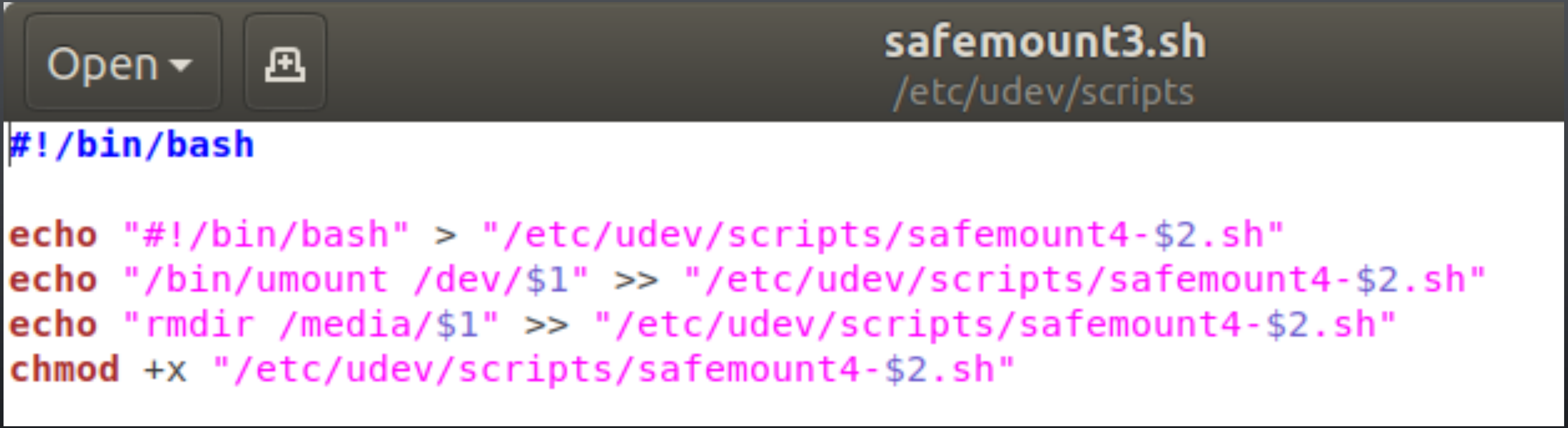


```
#!/bin/bash

echo "#!/bin/bash" > "/etc/udev/scripts/safemount2-$2.sh"
echo "mkdir /media/$1" >> "/etc/udev/scripts/safemount2-$2.sh"
echo "chmod 777 /media/$1" >> "/etc/udev/scripts/safemount2-$2.sh"
echo "/bin/mount /dev/$1 -o ro,noatime /media/$1" >> "/etc/udev/scripts/safemount2-$2.sh"
chmod +x "/etc/udev/scripts/safemount2-$2.sh"
```

UDEV BASICS – CONT'D

- Udev Rules Example:
  - Executing a shell script when a USB hub is removed
    - Safemount3.sh



```
#!/bin/bash

echo "#!/bin/bash" > "/etc/udev/scripts/safemount4-$2.sh"
echo "/bin/umount /dev/$1" >> "/etc/udev/scripts/safemount4-$2.sh"
echo "rmdir /media/$1" >> "/etc/udev/scripts/safemount4-$2.sh"
chmod +x "/etc/udev/scripts/safemount4-$2.sh"
```

SOFTWARE WRITE BLOCK DEMO

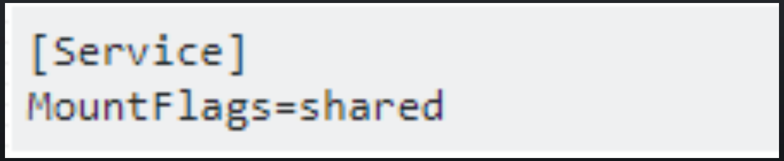
- TASK
  - Use udev ruls to turn a USB hub into a read-only hub so that any USB device plugged into is read-only

SOFTWARE WRITE BLOCK DEMO

- WHAT DO YOU NEED
  - A PC RUNNING UBUNTU* (17.10 FOR THIS DEMO)
  - A USB HUB
  - AT LEASE ONE WORKING USB DRIVE
  - ONE UDEV RULE FILE (.RULES)
  - TWO SHELL SCRIPTS
  - * THIS DEMO DOES NOT WORK WITH VIRTUAL MACHINE

SOFTWARE WRITE BLOCK DEMO

- STEPS
  1. Copy "/lib/systemd/system/systemd-udevd.service" to "/etc/system/system/systemd-udevd.service"
  2. Open "system-udevd.service" file and change "MountFlags=slave" to "MountFlags=shared"
  3. Create a new directory and file /etc/systemd/system/systemd-udevd.service.d/myoverride.conf, with the following two lines:



```
[Service]
MountFlags=shared
```

SOFTWARE WRITE BLOCK DEMO

- STEPS
  4. Create the rule file and name it "10-safemount.rules". The proceeding number will ensure the rule is executed early
  5. Put the rule file in "/etc/udev/rule.d" folder
  6. Create two shell scripts as discussed in class and name them as "safemount.sh" and "safemount3.sh", respectively.

SOFTWARE WRITE BLOCK DEMO

- STEPS
  5. Open a terminal and run the following two commands:

     ```
     sudo service udev restart
     sudo udevadm control --reload
     ```

  6. Plug in the USB hub, then insert a USB drive into the hub
  7. A drive “sdc1” will be mounted and appear in the file manager

SOFTWARE WRITE BLOCK DEMO

- STEPS
  8. OPEN THE NEW DRIVE AND TRY TO CREATE A NEW FOLDER OR NEW FILE
  9. WHILE IN THE TERMINAL, RUN "CD /MEDIA/SDC1"
  10. RUN "TOUCH TEST.TXT"
  11. RUN "SUDO CP /ETC/UDEV/SCRIPTS/SAFEMOUNT.SH /MEDIA/SDC1"

ANY "WRITE" OPERATION SHOULD RESULT IN "READ-ONLY FILE SYSTEM" ERROR