# Pardus 21 Linux发行版-远程代码执行0天2021 CVE-2021-3806

几天前，我提出了帕杜斯将组织一次报道错误比赛的消息。我喜欢为开源项目做出贡献。因此，这是一个重访我的老朋友帕杜斯并揭露安全和/或隐私问题的好机会。

**什么是帕杜斯？**

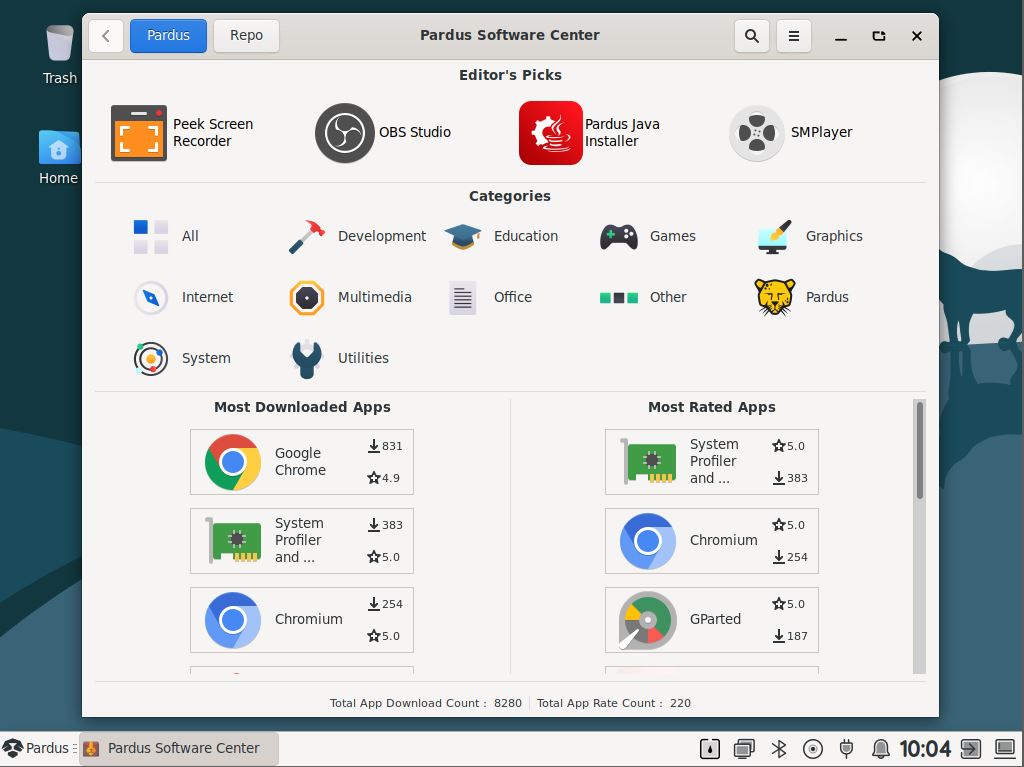
**Pardus**是在[土耳其](https://en.wikipedia.org/wiki/Turkey)政府支持下开发的[Linux发行版](https://en.wikipedia.org/wiki/Linux_distribution)。Pardus的主要重点是与办公室相关的工作，包括在土耳其[政府机构](https://en.wikipedia.org/wiki/Government_agency)中的使用。尽管如此，帕杜斯还是用几种语言发货。它易于使用和免费提供，在世界各地催生了许多社区。[[1]](https://en.wikipedia.org/wiki/Pardus_(operating_system))

在发展自己独特的身份之前，它开始作为一个基于Gentoo的项目生活。自2012年底以来，该发行版基于Debian GNU/Linux。

**这是基于Debian的发行版。那么，目标在哪里？**

早在2009年，我就是帕尔杜斯的志愿者之一。在那些日子里，这是一个优秀而有前途的项目。但Pardus选择将其Linux基础转移到Debian。据我所知，自2012年以来，他们主要围绕新核心开发软件包。

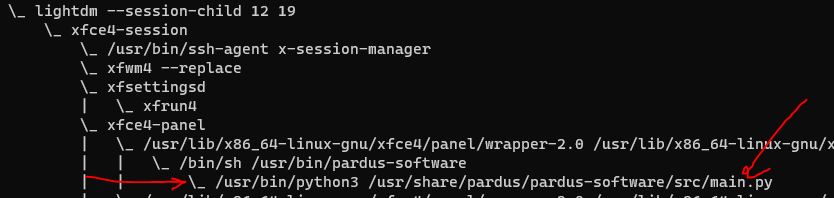
出于这个原因，我决定毫不犹豫地专注于软件包管理器的GUI。



**脆弱性研究**

Pardus 21的安装非常简单。但在互联网上找到帕杜斯团队开发的软件包的源代码有点复杂。我认为他们在努力发布时仍在处理此类问题。

让我们找到Pardus Software Center的位置。单击左下角的Pardus图标，然后单击Pardus Software Center快捷方式。这将启动这个过程。这个过程必须是xfce4会话的子进程。我们可以使用ps auxf轻松定位路径和程序。



作为下一步，将目标目录下的所有文件复制到主主机。这样我们就可以开始进攻性代码审查了。

?──（根💀DESKTOP-I30B071）-[~/pardus-software

?─# ls -al

总计196

drwxr-xr-x 2根4096 9月13日09:18。

drwx------ 4根4096 9月13日09:18...

-rwxr-xr-x 1 root 2707 9月13日 09:18 Actions.py

-rw-r--r-- 1根根 1719 9月13日 09:18 AppDetail.py

-rw-r--r-- 1根根 1674 9月13日 09:18 AppImage.py

-rw-r--r-- 1根根 1723 9月13日 09:18 AppRequest.py

-rw-r--r-- 1根根 1999年9月13日09:18 CellRendererButton.py

-rw-r--r-- 1根 2091 9月13日 09:18 GnomeComment.py

-rw-r--r-- 1根根 1094 9月13日 09:18 GnomeRatingServer.py

-rw-r--r-- 1根132924 9月13日 09:18 MainWindow.py

-rw-r--r-- 1根4974 9月13日 09:18 Package.py

-rw-r--r-- 1根根7744 9月13日 09:18 Server.py

-rw-r--r-- 1根根 1869年9月13日 09:18 UserSettings.py

-rw-r--r-- 1根根 9月11日 13日 09:18 \_\_版本\_\_

-rwxr-xr-x 1根根539 9月13日09:18 main.py100% 1094 3.1MB/s

**脆弱性：不安全的焦油提取**

当我审查这个项目时，我脑海中浮现出一个问题。这些应用程序图标来自哪里？不可能通过一个Pardus软件包运送所有图标，是吗？

以下函数取自MainWindows.py文件，在第706-712行之间。所以看起来图标来自服务器。

**def** getIcons（self）：

**如果**self.Server.connection：

# self.splashbar.pulse()

print（“从服务器获取图标”）

self.splashlabel.set\_markup("<b>{}</b>".format(\_("从服务器获取图标")))

self.serverappicons = self.Server.getAppIcons()

self.servercaticons = self.Server.getCategoryIcons()

Let’s see how does it fetch and use the icons. getAppIcons() is defined within Server.py file.

**def** getAppIcons(self)：

如果不是self.isExist(self.cachedir + self.serverappicons)：

print（“试图让小伙子失望”+ self.serverappicons）

**尝试**：

response = request.get(self.serverurl + self.serverfiles + self.serverappicons + self.serverarchive)

**除了**：

打印（

“服务器错误获取”+ self.serverurl + self.serverfiles + self.serverappicons + self.serverarchive）

**返回错误**

**如果** response.status\_code == 200：

**if** self.createDir(self.cachedir)：

**以**open（self.cachedir + self.serverappicons + self.serverarchive，“wb”）**为**文件：

file.write(response.content)

**if** self.extractArchive(self.cachedir + self.serverappicons + self.serverarchive,

self.serverappicons）：

**返回True**

**返回错误**

**其他**：

print("{} : {}".format("错误获取应用程序图标，状态代码", response.status\_code))

**返回错误**

**其他**：

**返回True**

简而言之，Pardus软件中心获取图标作为存档，并在第15行提取它们。那听起来不错。但我必须问两个问题：

1 – 任何形式的网络级加密？  
2 – 它如何提取档案？

让我们看看服务器的类变量，以回答第一个问题。事实上，这应该不成问题。因为它获取了一个包含大量图像的无辜存档文件。

**class** Server（对象）：

**def** \_\_init\_\_(self):

self.serverurl =“http://store.pardus.org.tr”

self.serverapps = "/api/v2/apps/"

self.servercats = "/api/v2/cats/"

self.serverhomepage = "/api/v2/homepage"

self.serversendrate = "/api/v2/rate"

self.serversenddownload = "/api/v2/download"

self.serversendsuggestapp = "/api/v2/suggestapp"

self.serverfiles = "/files/"

self.serverappicons =“应用程序”

self.servercaticons =“categoryicons”

self.serverarchive = ".tar.gz"

self.serversettings = "/api/v2/settings"

self.settingsfile = "serversettings.ini"

让我们看看它如何提取存档文件。以下函数取自Server.py文件，行介于163-173之间。

**def** extractArchive（自我、存档、类型）：

如果不是Path(self.cachedir + type).exist():

**尝试**：

tar = tarfile.open(archive)

tar.extractall(path=self.cachedir)

tar.close()

**返回True**

**除了**：

打印（“tarfile错误”）

**返回错误**

**返回True**

我没有看到对存档文件中的文件名称的任何验证。当文件具有目录遍历目标时，tarfile package的提取all()方法不会将文件提取限制在不同路径上。（例如：.../../../backdoor.sh) .存档文件将接管HTTP流量。你看我要去哪了。。

**剥削**

1 – 执行ARP中毒攻击，以成为中间人。

2 – 当Pardus用户打开Pardus软件中心应用程序时

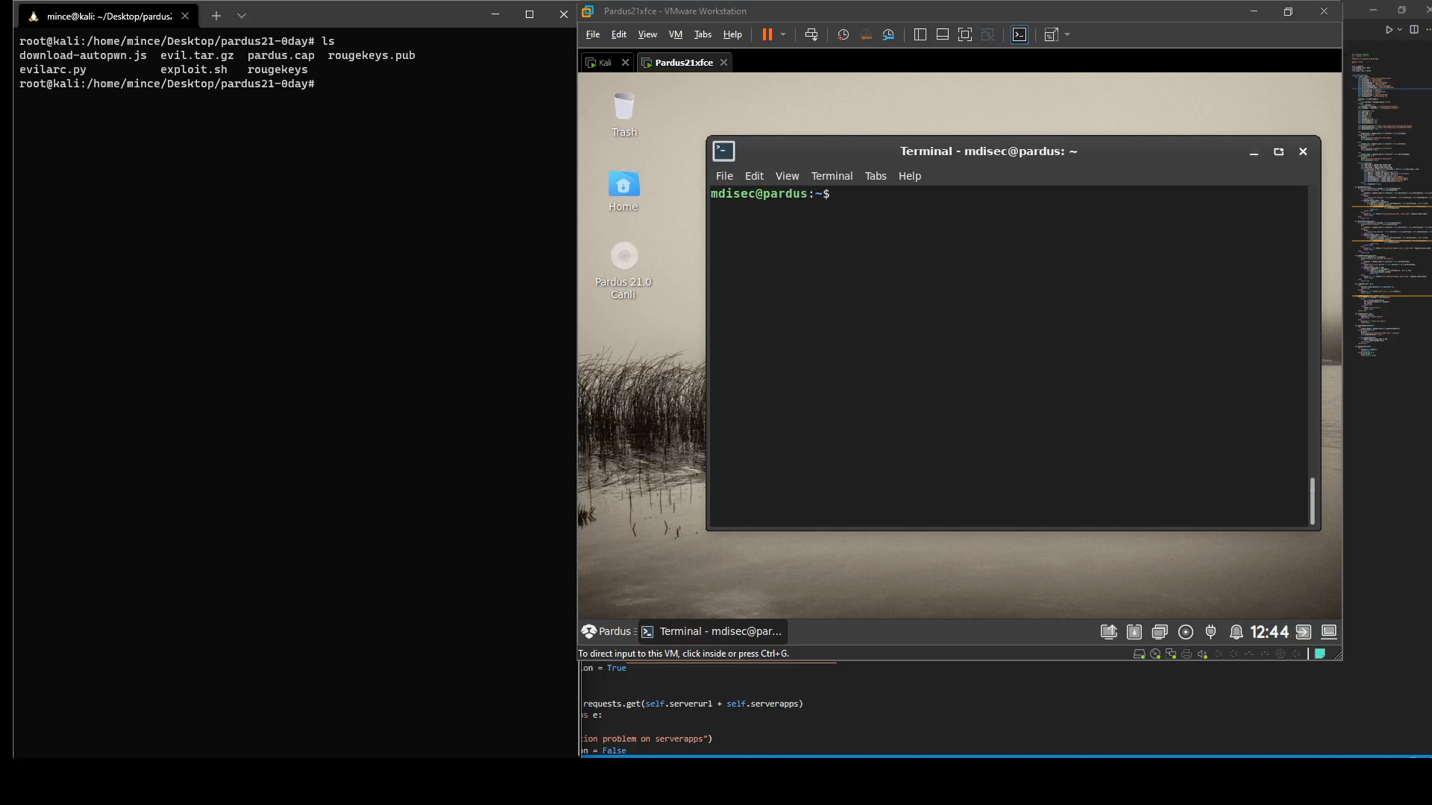
3 – 该应用程序将生成以下HTTP请求（<http://store.pardus.org.tr/files/appicons.tar.gz>）

4 – 拦截tar.gz文件，并将其替换为您自己的tar.gz文件，该文件包含.../../.ssh/authorized\_keys，带有您自己的ssh文件的公钥。

5- extractAll()方法将使用我们自己的tar.gz文件执行。

6 – 当不安全的焦油提取完成后，您的公共ssh密钥将存储在用户的个人文件夹下。这将给我们机会向目标打球！

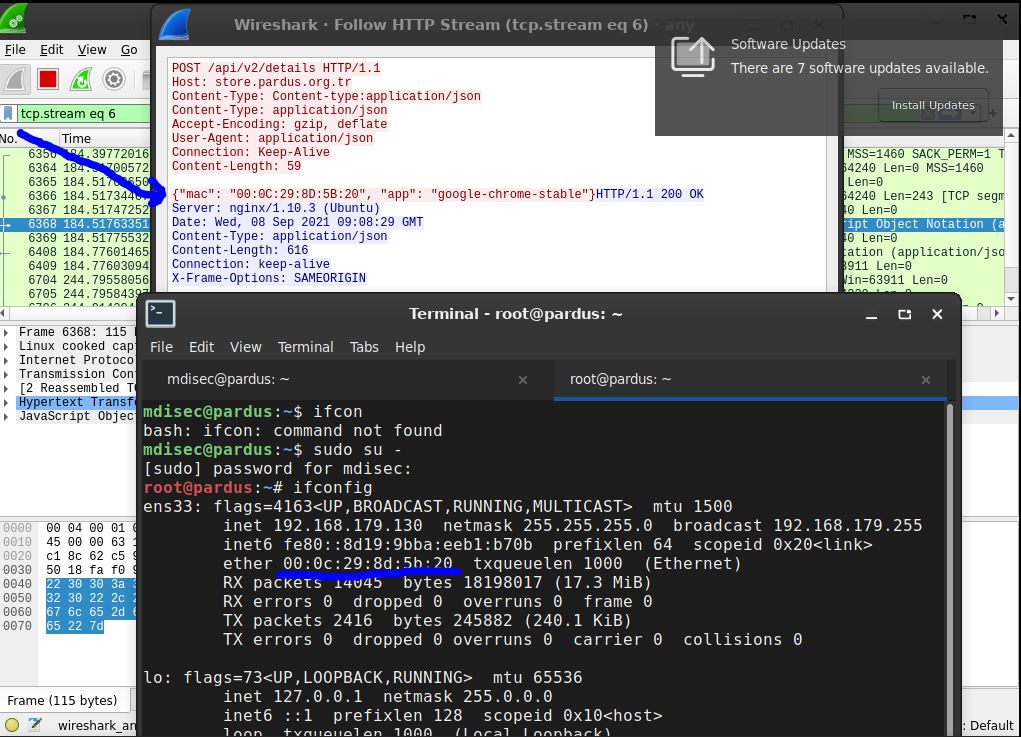
7 – SSH到您的公钥框中。

**概念证明**

我实现了bash脚本，可以自动执行所有步骤。

**奖金发现：隐私问题：Mac地址HTTP泄露**

在研究期间，我发现当用户单击应用程序图标时，Pardus软件中心正在发送包含计算机MAC地址的HTTP POST请求（至store.pardus.org.tr）。

* 

这个问题与脆弱性一起也得到了解决。

**时间表**

2021年9月07日上午01:15 – 研究开始

2021年9月07日上午02:10——发现漏洞。

2021年9月07日上午03:37——PoC已实施并录制视频。

2021年9月08日上午12:00——向帕杜斯团队报告。

2021年9月08日下午13:15 – 供应商验证了漏洞。

2021年9月09日上午7:17 – 补丁发布。