

Technitium MAC Address Changer

- What is a MAC address:

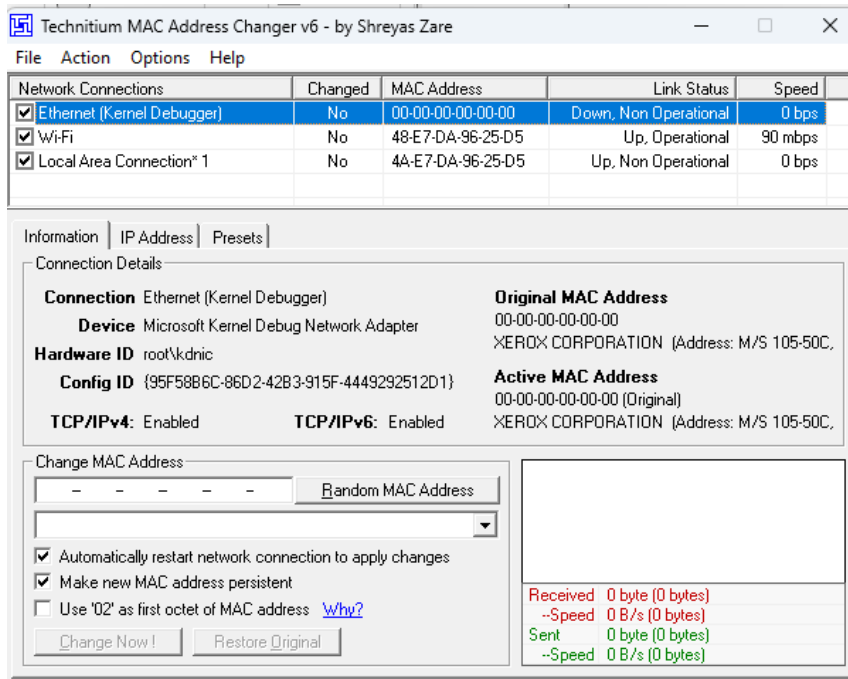
A MAC (Media Access Control) address, sometimes referred to as a physical address, is used to identify each device that has a network interface. This address is on the Network Interface Card (NIC) that a device uses to connect to a network. Each network interface on a device utilizes either multiple NICs or a single NIC. Most devices have a single NIC with multiple logical interfaces on it, and each logical interface usually has a separate MAC address unless configured differently. MAC addresses are used on layer 2 of the OSI model. This layer is known as the data link layer and consists of devices such as switches that forward traffic within a network based on the source and destination MAC addresses. Some network security systems also only allow certain MAC addresses to utilize the network. Thus, having a device with one of these MAC addresses is attractive to attackers.

- What is MAC address spoofing:

As stated above, MAC addresses are very important to computer networks and the security of these networks, which is why some attacks are based on masquerading as a different device by using that device's MAC address. This type of attack is known as MAC spoofing. An attacker will sometimes gain knowledge of what MAC addresses are allowed on a network, find a way to spoof one of those MAC addresses and infiltrate the network. However, two devices with the same MAC address cannot be on the network at the same time, making choosing the right MAC address at the right time a vital part of a MAC spoofing attack. It is also harder to track devices with a spoofed MAC address, which makes the identity of the attacker harder to obtain.

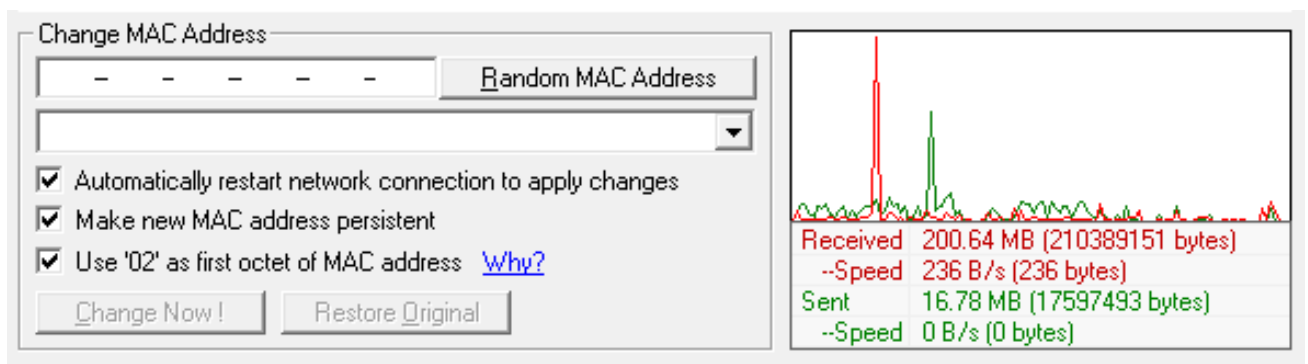
- How to spoof a MAC address using Technitium MAC Address Changer:

There are several programs that allow you to spoof MAC addresses, but one of the best ones is called Technitium MAC Address Changer (TMAC). TMAC has an extremely user-friendly interface that allows users to change their MAC addresses, view their different network interfaces, and provides other helpful tools. After opening the software, this is what is shown:

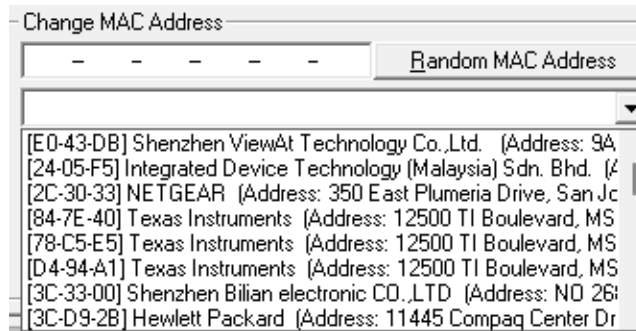


The different network interfaces are shown at the top, three tabs titled “Information”, “IP Address”, and “Presets” are below that. In the information tab, Connection details are shown including the original MAC address for the selected interface and the Active MAC address on that interface. Finally, the bottom of the window allows the user to change their MAC address in multiple ways.

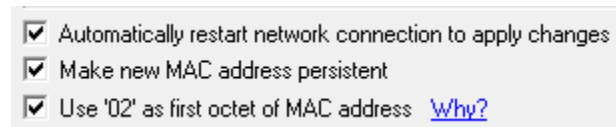
After selecting the interface that is being used, the user can either enter a custom MAC address (helpful if trying to spoof a specific address to bypass security mechanisms) or have a random MAC address generated. This would be helpful if the user simply wanted to conceal their identity.



By clicking the drop-down menu, the user can generate a MAC address from a specific vendor:



The three checkboxes below the drop-down menu allow for greater ease of use:



- The first checkbox automatically resets the network connection when the MAC address is changed when checked. This allows DHCP to provide a different IP address to the new MAC address.
- The second checkbox causes the new MAC address to be persistent when checked, which means that the interface will retain the new address if the device is restarted or turned off.
- Finally, the last checkbox makes the first octet of a randomly generated MAC address 02 when checked. This helps resolve certain issues when changing the MAC address on wireless internet adapters.

The following demonstration will show the process of changing a MAC address using Technitium:

MAC address after running ipconfig /all before the MAC change:

```
Wireless LAN adapter Wi-Fi:

Connection-specific DNS Suffix . : home
Description . . . . . : Realtek RTL8821CE 802.11ac PCIe Adapter
Physical Address. . . . . : 48-E7-DA-96-25-D5
DHCP Enabled. . . . . : Yes
Autoconfiguration Enabled . . . . : Yes
Link-local IPv6 Address . . . . . : fe80::5910:16b1:3ecb:ffb8%17(Preferred)
IPv4 Address. . . . . : 192.168.1.177(Preferred)
Subnet Mask . . . . . : 255.255.255.0
Lease Obtained. . . . . : Tuesday, October 22, 2024 4:42:04 PM
Lease Expires . . . . . : Wednesday, October 23, 2024 6:50:02 AM
Default Gateway . . . . . : 192.168.1.1
DHCP Server . . . . . : 192.168.1.1
DHCPv6 IAID . . . . . : 172550106
DHCPv6 Client DUID. . . . . : 00-01-00-01-2E-A9-B5-B8-02-D5-78-DD-DA-B7
DNS Servers . . . . . : 192.168.1.1
NetBIOS over Tcpip. . . . . : Enabled
```

After selecting Wi-Fi in the Network connections section and clicking “Random MAC address”:

The screenshot shows the Technitium MAC Address Changer v6 application window. The 'Network Connections' table lists three connections: Ethernet (Kernel Debugger), Wi-Fi, and Local Area Connection* 1. The Wi-Fi connection is selected, and its details are shown in the 'Connection Details' section. The 'Original MAC Address' is 48-E7-DA-96-25-D5. The 'Active MAC Address' is also 48-E7-DA-96-25-D5 (Original). The 'Change MAC Address' section shows a dropdown menu with 'Random MAC Address' selected. Below the dropdown, there are checkboxes for 'Automatically restart network connection to apply changes', 'Make new MAC address persistent', and 'Use '02' as first octet of MAC address'. The 'Change Now!' button is highlighted. On the right, a network graph shows data transfer statistics: Received 234.28 KB (239904 bytes), --Speed 37.13 KB/s (38024 bytes), Sent 176.74 KB (180979 bytes), and --Speed 23.05 KB/s (23604 bytes).

Network Connections	Changed	MAC Address	Link Status	Speed
<input checked="" type="checkbox"/> Ethernet (Kernel Debugger)	No	00-00-00-00-00-00	Down, Non Operational	0 bps
<input checked="" type="checkbox"/> Wi-Fi	No	48-E7-DA-96-25-D5	Up, Operational	150 mbps
<input checked="" type="checkbox"/> Local Area Connection* 1	No	4A-E7-DA-96-25-D5	Up, Non Operational	0 bps

Information | IP Address | Presets

Connection Details

Connection Wi-Fi

Device Realtek RTL8821CE 802.11ac PCIe Adapter

Hardware ID PCI\VEN_10EC&DEV_C821&SUBSYS_884D10C

Config ID {F7CB3F2A-9BED-418B-B8EA-3BD3D795980A}

TCP/IPv4: Enabled **TCP/IPv6:** Enabled

Original MAC Address 48-E7-DA-96-25-D5
Unknown Vendor

Active MAC Address 48-E7-DA-96-25-D5 (Original)
Unknown Vendor

Change MAC Address

02 - 02 - F8 - 49 - 73 - CE

[00-02-F8] SEAKR Engineering, Inc. (Address: 12847 E. Peakview)

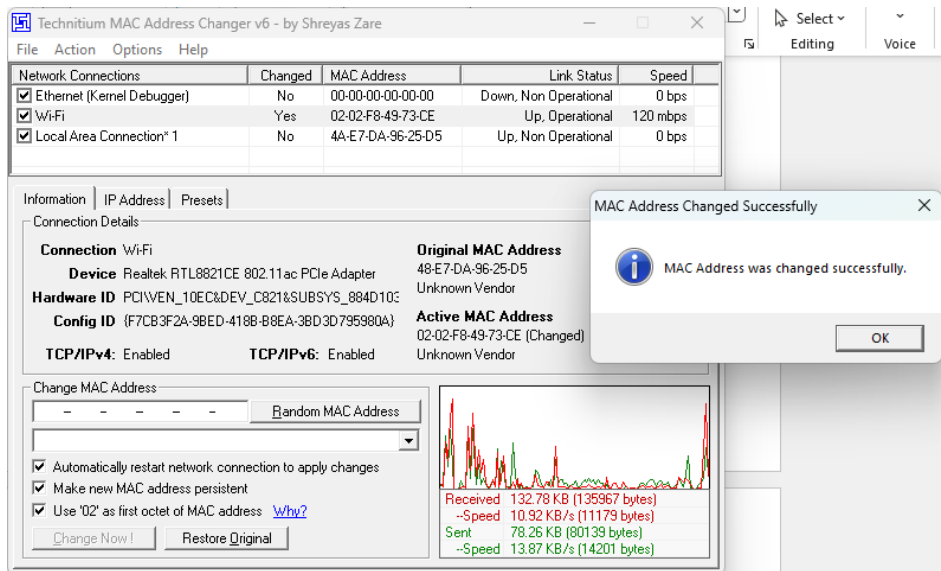
☒ Automatically restart network connection to apply changes

☒ Make new MAC address persistent

☒ Use '02' as first octet of MAC address [why?](#)

Received 234.28 KB (239904 bytes)
--Speed 37.13 KB/s (38024 bytes)
Sent 176.74 KB (180979 bytes)
--Speed 23.05 KB/s (23604 bytes)

After clicking “Change Now!”:

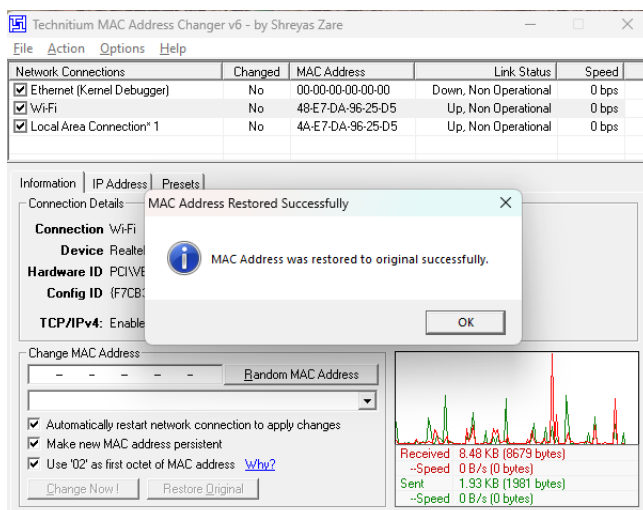


This change is reflected when running the ipconfig /all command in command prompt again:

```
Wireless LAN adapter Wi-Fi:

Connection-specific DNS Suffix . : home
Description . . . . . : Realtek RTL8821CE 802.11ac PCIe Adapter
Physical Address. . . . . : 02-02-F8-49-73-CE
DHCP Enabled. . . . . : Yes
Autoconfiguration Enabled . . . . : Yes
Link-local IPv6 Address . . . . . : fe80::e315:6c75:8af:e57b%17(Preferred)
IPv4 Address. . . . . : 192.168.1.228(Preferred)
Subnet Mask . . . . . : 255.255.255.0
Lease Obtained. . . . . : Tuesday, October 22, 2024 7:57:21 PM
Lease Expires . . . . . : Wednesday, October 23, 2024 7:57:20 AM
Default Gateway . . . . . : 192.168.1.1
DHCP Server . . . . . : 192.168.1.1
DHCPv6 IAID . . . . . : 172550106
DHCPv6 Client DUID. . . . . : 00-01-00-01-2E-A9-B5-B8-02-D5-78-DD-DA-B7
DNS Servers . . . . . : 192.168.1.1
NetBIOS over Tcpip. . . . . : Enabled
```

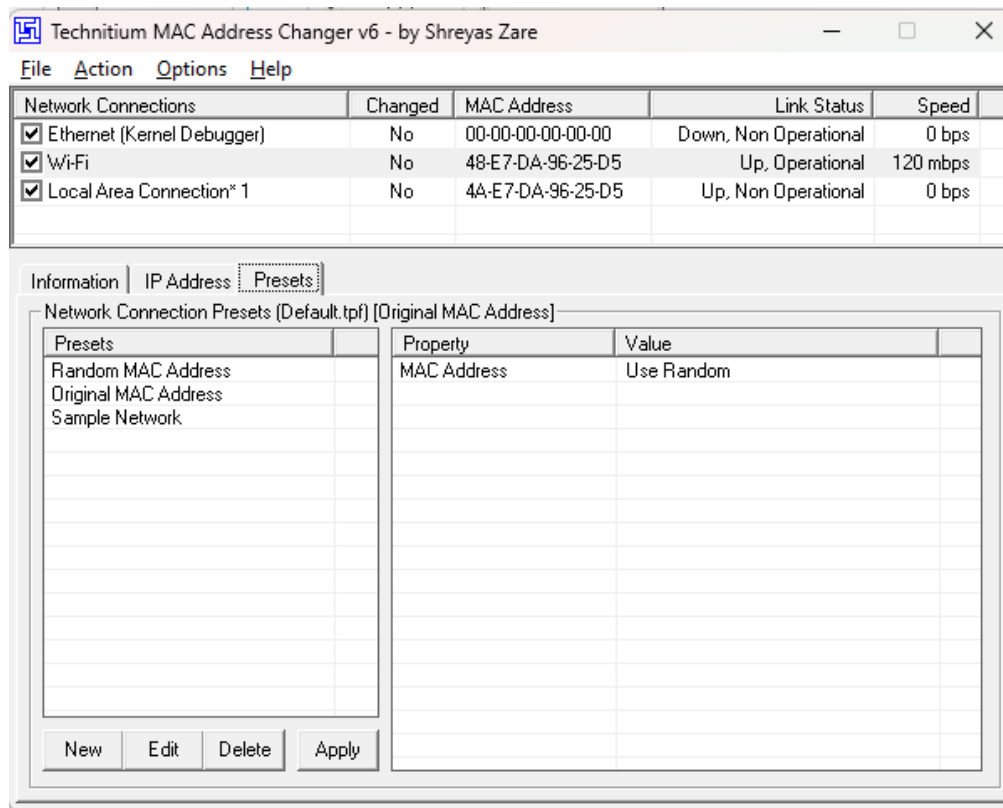
The original MAC address can then be restored by simply clicking “Restore Original”:



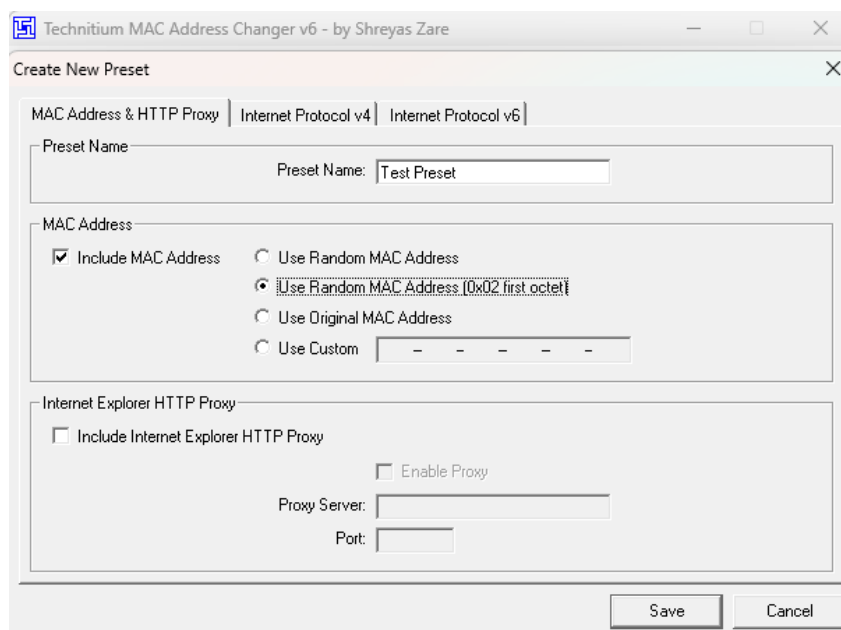
- By moving over to the IP Address tab, some of the details of the selected interface are shown, including the current IP address (IPv4 or IPv6), the gateway (usually the IP address of a router), and the IP address of the DNS server.

All of these sections also allow the user to add or remove rows to consider other configurations.

By moving to the Presets tab, the user can create presets to be used whenever necessary:



When creating a preset by clicking “New”, the user has the same options for the MAC address they want to spoof along with the options to name the preset and use an internet proxy if they wish, which could help conceal their identity even more.



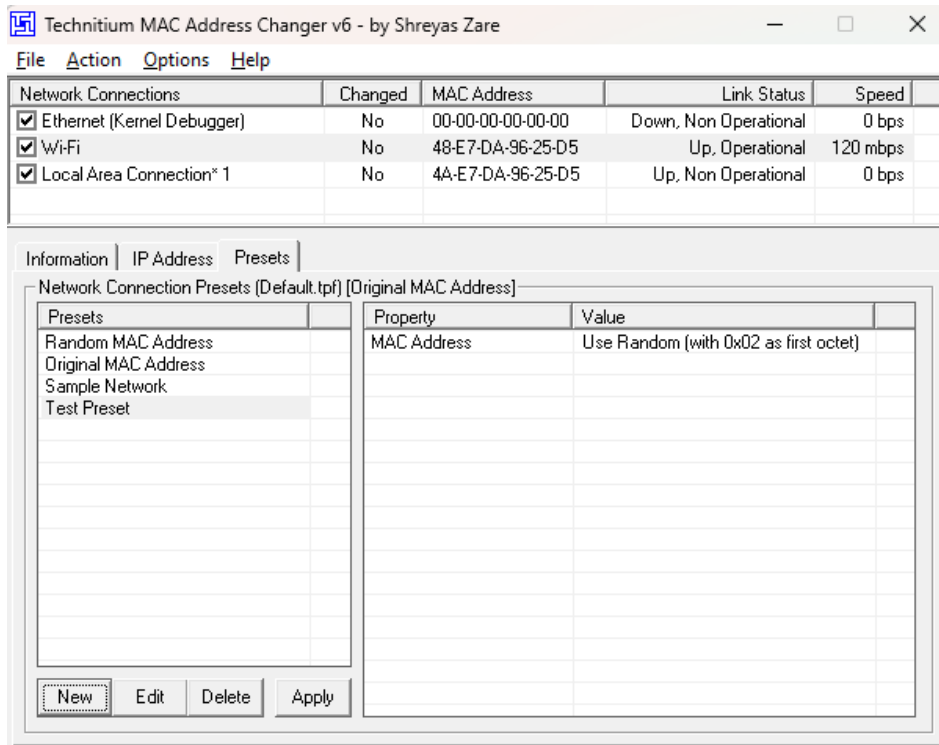
Then, the user has the option to specify how an IP address will be obtained when the preset is active. They can choose to obtain an IP address via DHCP or by supplying a static IP address. A static IP address could be helpful to avoid some conflicts and to blend into certain networks more seamlessly. The user also has the option to specify the IP address of the gateway and the DNS server.

The screenshot shows the 'Create New Preset' dialog box with the 'Internet Protocol v4' tab selected. The 'MAC Address & HTTP Proxy' tab is also visible. The 'Internet Protocol v4 Parameters' section contains a checkbox for 'Include Internet Protocol v4' and four sub-options: 'DHCPv4', 'IPv4 Address', 'IPv4 Gateway', and 'IPv4 DNS Server'. The 'Internet Protocol v4 [Not Selected]' section contains three input fields: 'IP Address (0)' and 'Subnet Mask' (grouped), 'Gateway (0)' and 'Metric' (grouped), and 'DNS Server (0)'. The 'Save' and 'Cancel' buttons are at the bottom right.

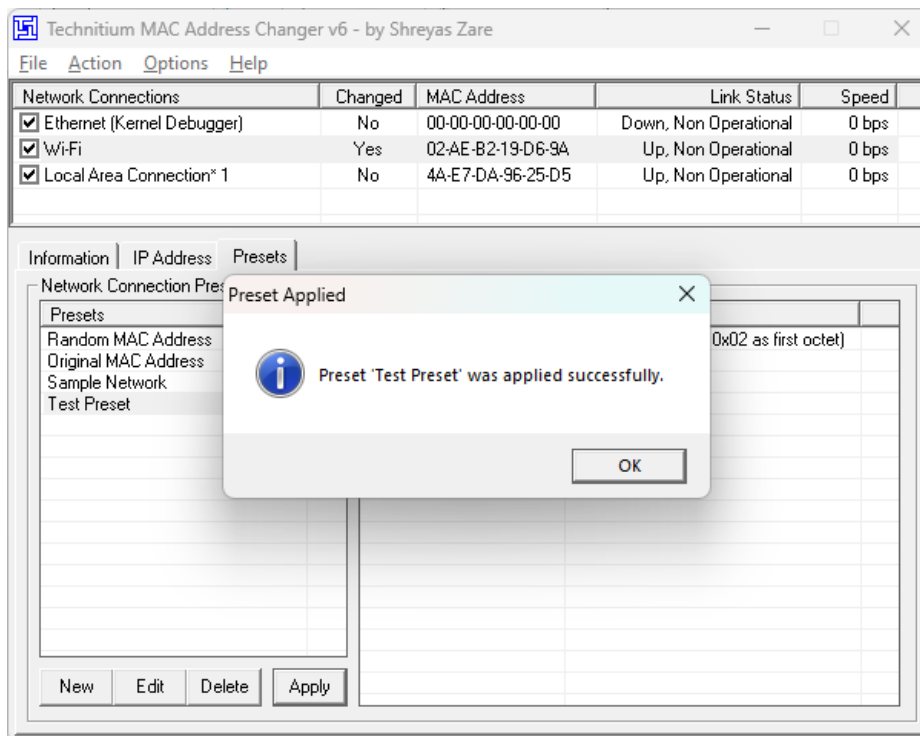
All of these options are also available using IPv6:

The screenshot shows the 'Create New Preset' dialog box with the 'Internet Protocol v6' tab selected. The 'MAC Address & HTTP Proxy' and 'Internet Protocol v4' tabs are also visible. The 'Internet Protocol v6 Parameters' section contains a checkbox for 'Include Internet Protocol v6' and four sub-options: 'DHCPv6', 'IPv6 Address', 'IPv6 Gateway', and 'IPv6 DNS Server'. The 'Internet Protocol v6 [Not Selected]' section contains three input fields: 'Unicast IP Address (0)' and 'Prefix' (grouped), 'Gateway/Next Hop (0)' and 'Metric' (grouped), and 'DNS Server (0)'. The 'Save' and 'Cancel' buttons are at the bottom right.

By clicking “Save”, the preset is added to the list of presets that can be used, edited, or deleted later:



A preset can be activated by clicking the desired preset and then clicking “Apply”:



This change is reflected on the Active MAC Address section of the information tab and by running the ipconfig /all command in command prompt:

Technitium MAC Address Changer v6 - by Shreyas Zare

File Action Options Help

Network Connections	Changed	MAC Address	Link Status	Speed
<input checked="" type="checkbox"/> Ethernet (Kernel Debugger)	No	00-00-00-00-00-00	Down, Non Operational	0 bps
<input checked="" type="checkbox"/> Wi-Fi	Yes	02-AE-B2-19-D6-9A	Up, Operational	120 mbps
<input checked="" type="checkbox"/> Local Area Connection* 1	No	4A-E7-DA-96-25-D5	Up, Non Operational	0 bps

Information | IP Address | Presets

Connection Details

Connection Wi-Fi

Device Realtek RTL8821CE 802.11ac PCIe Adapter

Hardware ID PCI\VEN_10EC&DEV_C821&SUBSYS_884D10C

Config ID {F7CB3F2A-9BED-418B-B8EA-3BD3D795980A}

TCP/IPv4: Enabled **TCP/IPv6:** Enabled

Original MAC Address
48-E7-DA-96-25-D5
Unknown Vendor

Active MAC Address
02-AE-B2-19-D6-9A (Changed)
Unknown Vendor

Change MAC Address

- - - - - Random MAC Address

☒ Automatically restart network connection to apply changes

☒ Make new MAC address persistent

☒ Use '02' as first octet of MAC address [Why?](#)

Received 422.73 KB (432874 bytes)
--Speed 172 B/s (172 bytes)
Sent 483.78 KB (495389 bytes)
--Speed 0 B/s (0 bytes)

Wireless LAN adapter Wi-Fi:

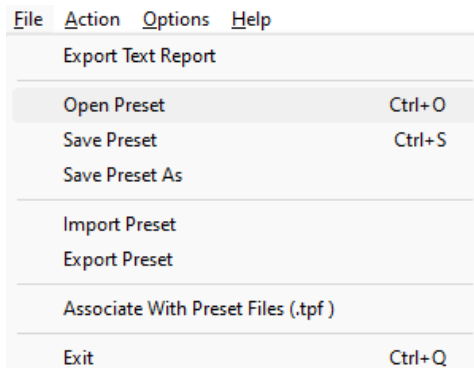
```

Connection-specific DNS Suffix . : home
Description . . . . . : Realtek RTL8821CE 802.11ac PCIe Adapter
Physical Address. . . . . : 02-AE-B2-19-D6-9A
DHCP Enabled. . . . . : Yes
Autoconfiguration Enabled . . . . : Yes
Link-local IPv6 Address . . . . . : fe80::a1a0:baa5:2cd4:5f99%17(Preferred)
IPv4 Address. . . . . : 192.168.1.197(Preferred)
Subnet Mask . . . . . : 255.255.255.0
Lease Obtained. . . . . : Tuesday, October 22, 2024 8:14:31 PM
Lease Expires . . . . . : Wednesday, October 23, 2024 8:14:31 AM
Default Gateway . . . . . : 192.168.1.1
DHCP Server . . . . . : 192.168.1.1
DHCPv6 IAID . . . . . : 172550106
DHCPv6 Client DUID. . . . . : 00-01-00-01-2E-A9-B5-B8-02-D5-78-DD-DA-B7
DNS Servers . . . . . : 192.168.1.1
NetBIOS over Tcpip. . . . . : Enabled

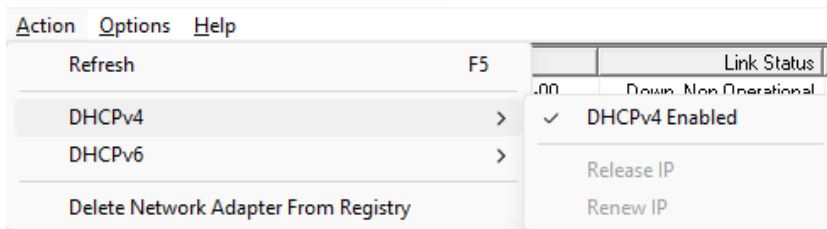
```

- Other helpful Features:

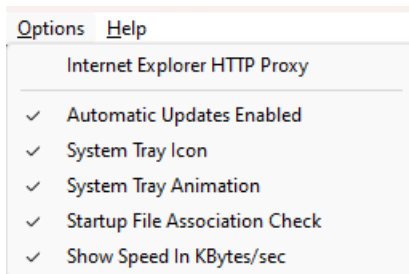
By selecting “File” at the top of the page, presets can be opened from another location:



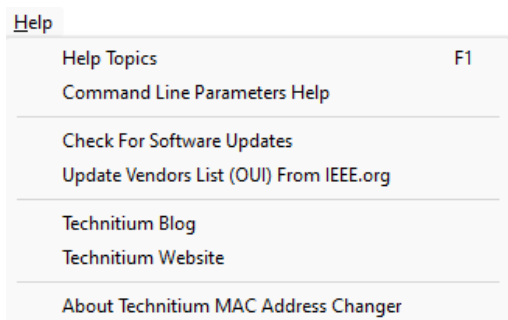
By selecting “Action” at the top of the page, some IP functions are available including releasing and renewing an IP address:



By selecting “Options” at the top of the page, the user can specify a proxy server or change viewing preferences:



By selecting “Help” at the top of the page, the user can find helpful articles to guide them:



- Demonstration of how MAC spoofing can be used:

Many networks have security features that limit network access to certain approved MAC addresses. If an attacker is able to somehow obtain the list of permitted MAC addresses or even one permitted MAC address, they can bypass this security feature. However, the attacker has to make sure that the device that actually has that MAC address is not on the network at the same time.

The following Python program simulates this type of security feature:

```
# This program utilizes the psutil library to obtain the MAC address of the network interface
# of the device running it. If the MAC address is an approved MAC address it can "access the network"
# If the MAC address isn't approved, the program will display "Access Denied"
import psutil

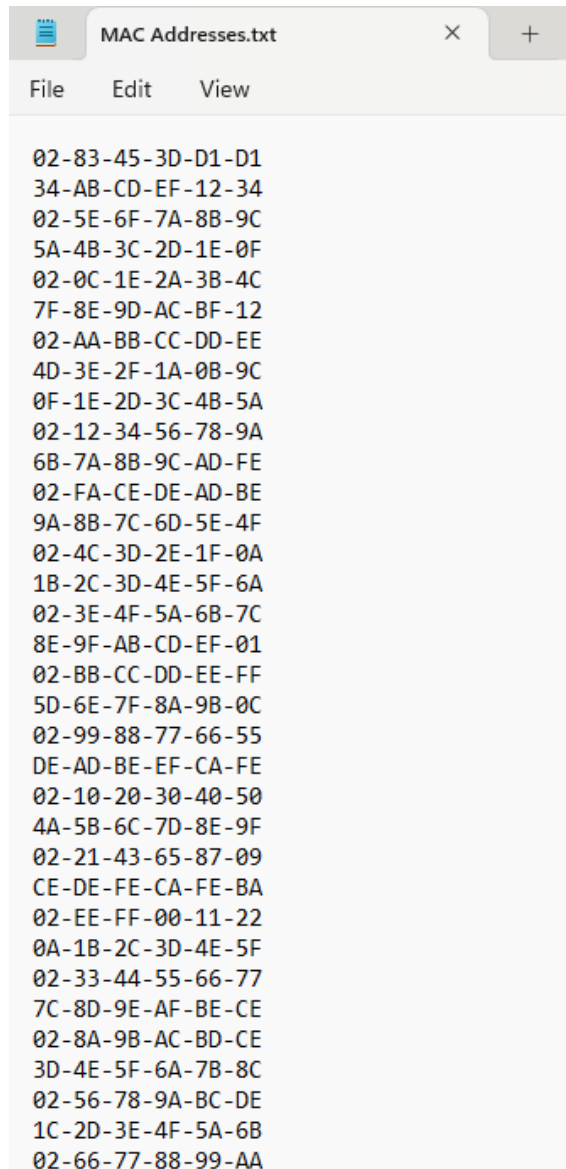
def get_mac():
    # Get all network interfaces
    interfaces = psutil.net_if_addrs()
    # psutil.net_if_addrs() returns a dictionary with keys of adapter names and values
    # of MAC addresses and ip addresses associated with the interface
    interface_name = input('Enter the name of your network interface: ')
    # User must enter the name of their network interface
    for interface, addrs in interfaces.items():
        # Check for specified interface (My wireless adapter's name is "Wi-Fi")
        if interface_name in interface:
            for addr in addrs:
                if addr.family == psutil.AF_LINK: # Check for MAC address family
                    # addr.family refers to the family attribute associated with the
                    # addresses of the interface (such as MAC, IPv4, and IPv6)
                    # psutil.AF_LINK refers to link layer addresses (MAC addresses)
                    return addr.address #returns MAC address
    return None

print('Welcome to the program!')
choice = ''
while choice.lower().strip() not in ['y', 'n']:
    choice = input('Would you like to access the network (Y/n): ')
    if choice.lower().strip() not in ['y', 'n']:
        print('Invalid input')

if choice.lower().strip() == 'y':
    # Obtain MAC address of adapter
    mac_address = get_mac()
    if mac_address:
        print(f'MAC address is: {mac_address}')
    else:
        print('Adapter not found.')
    # Grant access if MAC address is in the list of approved MAC addresses
    with open('MAC Addresses.txt', 'r') as file:
        contents = file.read()
    if mac_address in contents:
        print('Access Granted')
    else:
        print('Access Denied')
```

The program asks the user if they want to access the network, and if they want to, it asks the user to provide the network interface that they are using inside of the `get_mac()` function. The function retrieves all network interfaces and finds the MAC address of the interface specified by the user attempting to access the network. If their MAC address matches one of the MAC addresses in the file that stores the allowed MAC addresses, an “Access Granted” message is shown. Otherwise, the user sees “Access Denied”. The code is explained in further detail in the comments of the code.

The file that contains the allowed Mac addresses looks like this:



```
02-83-45-3D-D1-D1
34-AB-CD-EF-12-34
02-5E-6F-7A-8B-9C
5A-4B-3C-2D-1E-0F
02-0C-1E-2A-3B-4C
7F-8E-9D-AC-BF-12
02-AA-BB-CC-DD-EE
4D-3E-2F-1A-0B-9C
0F-1E-2D-3C-4B-5A
02-12-34-56-78-9A
6B-7A-8B-9C-AD-FE
02-FA-CE-DE-AD-BE
9A-8B-7C-6D-5E-4F
02-4C-3D-2E-1F-0A
1B-2C-3D-4E-5F-6A
02-3E-4F-5A-6B-7C
8E-9F-AB-CD-EF-01
02-BB-CC-DD-EE-FF
5D-6E-7F-8A-9B-0C
02-99-88-77-66-55
DE-AD-BE-EF-CA-FE
02-10-20-30-40-50
4A-5B-6C-7D-8E-9F
02-21-43-65-87-09
CE-DE-FE-CA-FE-BA
02-EE-FF-00-11-22
0A-1B-2C-3D-4E-5F
02-33-44-55-66-77
7C-8D-9E-AF-BE-CE
02-8A-9B-AC-BD-CE
3D-4E-5F-6A-7B-8C
02-56-78-9A-BC-DE
1C-2D-3E-4F-5A-6B
02-66-77-88-99-AA
```

This is just part of the file, but any of these MAC addresses will do.

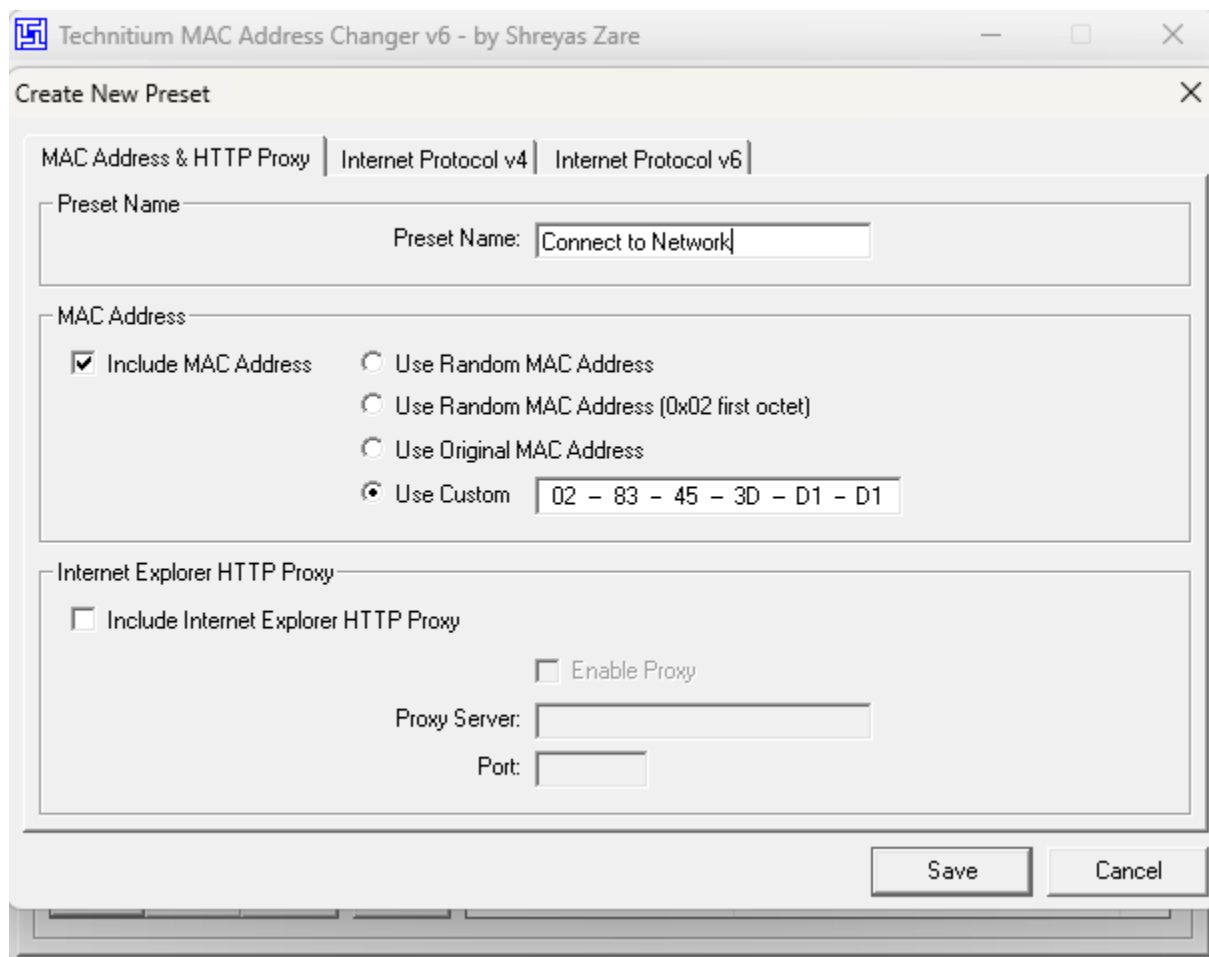
Note: The octets in MAC addresses are often separated by a colon instead of a dash, but the `psutil.AF_LINK` family in the `psutil` library returns the address with dashes.

By running the program with the MAC address on my PC (which is not in the list of allowed MAC addresses), this output is shown:

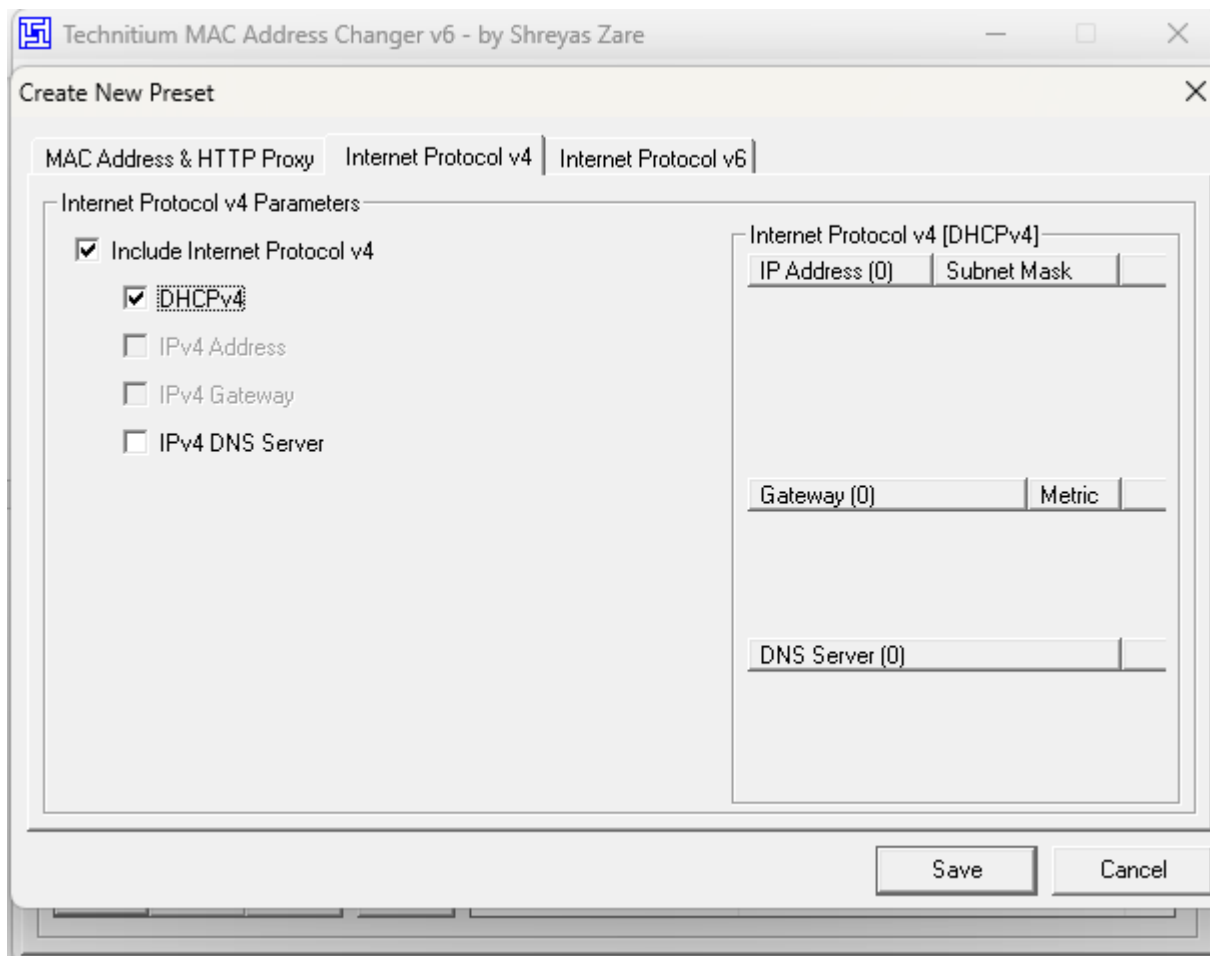
```
C:\WINDOWS\system32\cmd.exe
Welcome to the program!
Would you like to access the network (Y/n): Y
Enter the name of your network interface: Wi-Fi
MAC address is: 48-E7-DA-96-25-D5
Access Denied
Press any key to continue . . .
```

TMAC can be used to spoof a MAC address that will be able to access the network.

A preset can be created in TMAC so that the network can be accessed by the attacker whenever they want by clicking “New” in the “Presets” tab, naming the preset, and checking “Include MAC Address”. Then, the attacker can simply type in an allowed MAC address (the one at the top of the allowed MAC address list is used here):

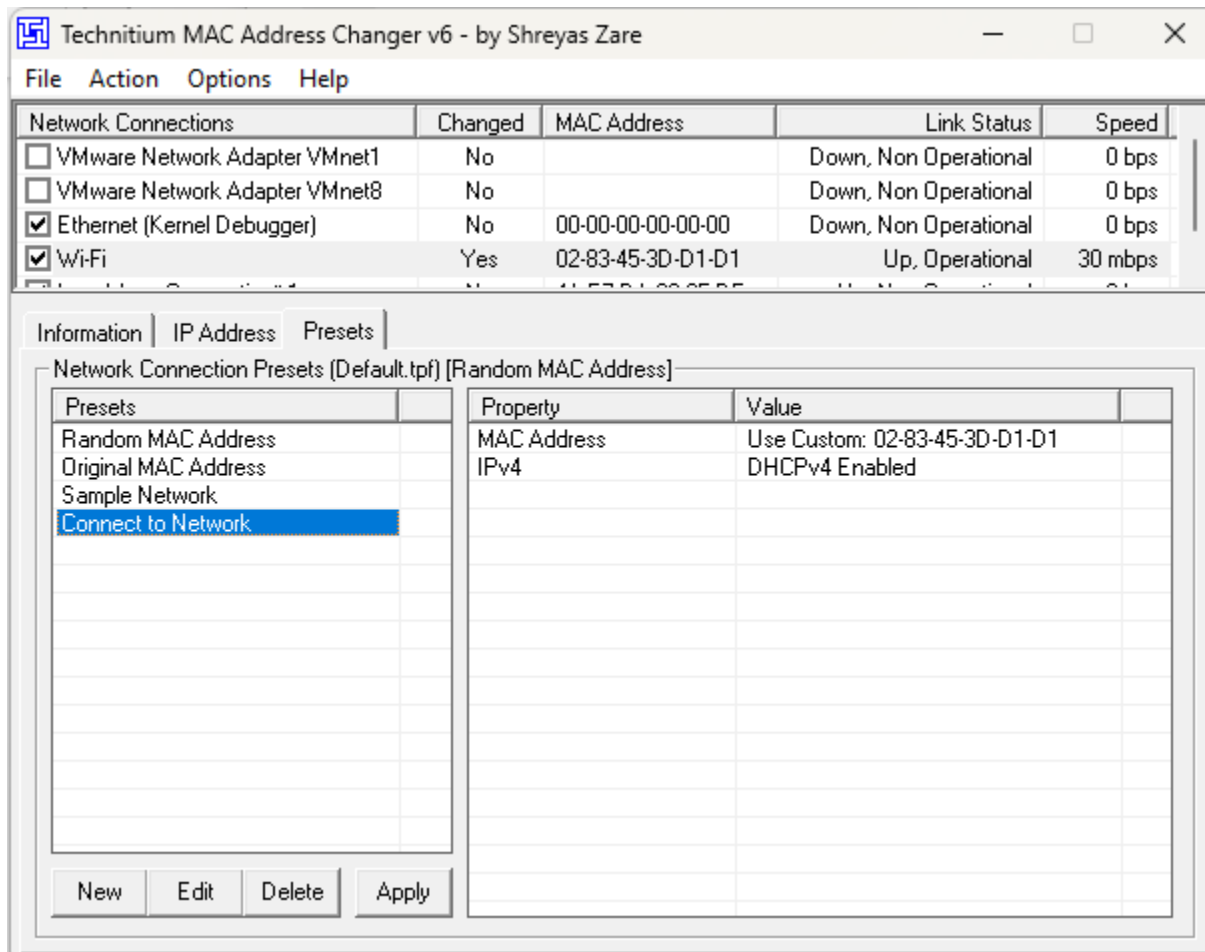


An IP address can also be obtained for the new MAC address by switching to the “Internet Protocol v4” tab (or v6 if needed), checking “Include Internet Protocol v4”, and checking “DHCPv4”.



The preset can then be created by clicking “Save”.

To apply the preset, select the preset and click “Apply” (the new MAC address is also visible at the top of the screen next to the adapter being used):



Now that a MAC address from the list of allowed MAC addresses has been spoofed, the attacker can run the program again and be granted access to the network:

```
C:\WINDOWS\system32\cmd.exe
Welcome to the program!
Would you like to access the network (Y/n): Y
Enter the name of your network interface: Wi-Fi
MAC address is: 02-83-45-3D-D1-D1
Access Granted
Press any key to continue . . .
```

This is a very oversimplified example of how a MAC spoofing attack works, but the general concept of changing a device’s MAC address to gain access to something that the device should not have access to remains the same.