

Forum:[Hack Forums](#)

Thread: [\[Tutorial\] REVERSE SHELL SERIES - The Basics \(Part 2\) \[Tutorial\]](#)

TheCyb3rAlpha - April 26, 2020, 6:18 pm

REVERSE SHELL SERIES - The Basics (Part 2)

Greetings to all HF members,

I'm introducing the reverse shell series for everyone who wants to learn about the reverse shell in depth. As me and my partner [??\\$???](#) have already written a book - [Hands On - Red Team Tactics](#) that has this chapter, we would like to expand the horizon for everyone to learn more about reverse shells with different case scenarios.

Checkout the first part here: [REVERSE SHELL SERIES - The Basics \(Part 1\)](#)

In this tutorial, you'll learn about:

- [+] Getting started with Reverse TCP Shell
- [+] Analysing Reverse TCP Shell using Wireshark
- [+] Issues with the basic reverse shell

Starting with reverse TCP shell

Spoiler ([Click to View](#))

As the name suggests, the reverse TCP shell uses the Transmission Control Protocol (TCP) as its main connection protocol. To get a reverse TCP shell, a command needs to be executed on the target machine that will do the following:

- [*]Initiate a TCP three-way handshake to the attacker machine,
- [*]Redirects the input and output of a command shell (shovel a shell) using existing or custom file descriptors (input, output & error file descriptors for Linux and file handles [child_in_r, child_in_w, child_out_r, child_out_w] for Windows)
- [*]Connects with the attacker machine and listen for input commands,
- [*]Gets the command using the input file descriptor from the attacker machine, executes the command, and sends the command output to the attacker machine.

Let us understand using the example machines. The details for the machines are as follows:

Attacker Machine: 192.168.2.19

Target Server: 192.168.2.12

Note: As both the machines are in the same network, we don't have to do any egress test or port forwarding.

Before running any kind of command, a listener should be running on the attacker machine so lets open up port 3131/tcp here.

On executing the command on the target machine, it should be initiating a connection to the attacker machine. Note: We have used ncat.exe simply for the readers to understand.

Now on the attacker machine, we can see the Command Shell connected.

We can execute a command on the connected Command Shell just like a normal command in cmd.exe. In this case, we have executed 'whoami' command

Analyzing Reverse TCP Shell using Wireshark

Spoiler ([Click to View](#)) Even though this section is optional and only gives you a more in-depth understanding of how actually a reverse shell works on a network packet level. From the previous setup, when the attacker executed the ncat.exe command on the target machine. Let's understand what all happened from there onwards. Target Machine initiated a three-way handshake (SYN, SYN-ACK, ACK) with the attacker machine (The first 3 packets shows the handshake).

After the three-way handshake is completed, the attacker machine sends a [TCP Window Update] for the target machine to set the TCP Window size for the transfer.

When this is done, the target machine will send the Command Shell Banner to the attacker machine. The next 4 packets are about the transfer of cmd.exe banner to the attacker machine.

Here the Banner is split into two packets: Packet 1 (38 bytes)

and Packet 2 (86 bytes). The reason for the split is 0x0a (line feed) and 0x0d (carriage return) in the banner. You can see the CRLF in the first two bytes of the data.

Issues with the basic reverse shell

Spoiler ([Click to View](#)) Un-encrypted data. TCP reverse shells without any kind of encryption may end up alerting the monitoring team. E.g. Sonicwall detects the presence of a generic reverse shell by following the signature (

http://help.sonicwall.com/help/sw/eng/9530/26/2/3/content/Application_Control.065.23.htm)

If exfiltration is done using a generic reverse shell, the number of packets will be alarming itself.

Hope everyone likes it.

Checkout the third part here: [REVERSE SHELL SERIES - The Basics \(Part 3\)](#)

Stay connected and stay safe! Cheers ?

??¢??? - April 30, 2020, 11:06 pm

Great tutorial. I wonder why no idiots on this forum are learning from this.

de_g3ntl3m4n - April 30, 2020, 11:13 pm

bahhaha nobody likes to read but keep it up cant wait to see part 3 thank you for the tuts very nice

??¢??? - May 1, 2020, 12:37 am

(Apr 30, 2020 11:13 PM)de_g3ntl3m4n Wrote: [\[CITE\]](#) bahhaha nobody likes to read but keep it up cant wait to see part 3 thank you for the tuts very nice

There is already a part 3 :

<https://hackforums.net/showthread.php?tid=6072215>

de_g3ntl3m4n - May 1, 2020, 12:47 am

(May 1, 2020 12:37 AM)??¢??? Wrote: [\[CITE\]](#) There is already a part 3 :

<https://hackforums.net/showthread.php?tid=6072215>

im dead thanks for the link ahahahahaha much respect

13uNkNoWn - July 10, 2020, 8:49 pm

First I want to say, I've read part one, and now part 2.

Part one, was a great introduction to the idea, good data/information.

Here in part two, more good data, but I have a bit of criticism of the example.

You made it too basic for users understanding, and actually it's no longer a valid example of reverse shell.

I feel like to really show an example of a reverse shell it would need to give an example of the "Attacker Machine" making use of some vulnerability to force the "Target Server" to connect back to the "Attacker Machine".

It's a very small distinction, but very important to point out in your instruction, otherwise what you have shown an example of is more of a direct connection.

In this case the "Attacker Machine" acts as the server, and has the server(application) running with network port listening.

The "Target server" becomes the actual client, and connects to the server manually, nothing more, nothing reverse about this example.

Part 3 after I eat!

But really great that you are putting the effort to share the knowledge and format things well with examples and screen shots!

Thanks!

ceR0 - July 22, 2020, 8:09 am

Great stuff mate keep it up! much appreciated !

TheCyb3rAlpha - August 2, 2020, 1:24 pm

(Jul 10, 2020 08:49 PM)13uNkNoWn Wrote: [\[CITE\]](#) First I want to say, I've read part one, and now part 2.

Part one, was a great introduction to the idea, good data/information.

Here in part two, more good data, but I have a bit of criticism of the example.

You made it too basic for users understanding, and actually it's no longer a valid example of reverse shell.

I feel like to really show an example of a reverse shell it would need to give an example of the "Attacker Machine" making use of some vulnerability to force the "Target Server" to connect back to the "Attacker Machine".

It's a very small distinction, but very important to point out in your instruction, otherwise what you have shown an example of is more of a direct connection.

In this case the "Attacker Machine" acts as the server, and has the server(application) running with network port listening.

The "Target server" becomes the actual client, and connects to the server manually, nothing more, nothing reverse about this example.

Part 3 after I eat!

But really great that you are putting the effort to share the knowledge and format things well with examples and screen shots!

Thanks!

Thanks for the feedback. I actually wanted to cover the very basics first as there was already a lot of topics (Meterpreter, tunnels, encryption, etc.) lined up but I'm not getting the time at all.

Anyways, Much appreciated mate!