

Forum:[Hack Forums](#)

Thread: [\[Tutorial\] REVERSE SHELL SERIES - The Basics \(Part 3\) \[Tutorial\]](#)

TheCyb3rAlpha - April 27, 2020, 6:20 pm

REVERSE SHELL SERIES - The Basics (Part 3)

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 \(Part 1\)](#)

Checkout the second part here: [REVERSE SHELL SERIES - The Basics \(Part 2\)](#)

In the previous tutorial, you learned how to start with a basic TCP reverse shell and what all happens during the connection process on a packet level. In this tutorial we'll focus on UDP protocol and getting shells over the UDP connection protocol.

In this tutorial, you'll learn about:

- [+] Learning about UDP Protocol
- [+] Introduction to Reverse UDP shell
- [+] Analysing UDP Reverse Shells using Wireshark

Learning about UDP Protocol

(Shamelessly copied this section from:

<https://searchnetworking.techtarget.com/definition/UDP-User-Datagram-Protocol>)

Spoiler ([Click to View](#)) UDP (User Datagram Protocol) is a communications protocol that is primarily used for establishing low-latency and loss-tolerating connections between applications on the internet. It speeds up transmissions by enabling the transfer of data before an agreement is provided by the receiving party. As a result, UDP is beneficial in time-sensitive communications, including voice over Internet Protocol (VoIP), domain name system (DNS) lookup, and video or audio playback. UDP is an alternative to Transmission Control Protocol (TCP).

UDP uses headers when packaging message data to transfer over network connections. UDP headers contain a set of parameters -- called fields -- defined by the technical specifications of the protocol.

User Datagram Protocol has attributes that make it beneficial for use with applications that can tolerate lost data. For example:

- [*] It allows packets to be dropped and received in a different order than they were transmitted, making it suitable for real-time applications where latency might be a concern.
- [*] It can be used for transaction-based protocols, such as DNS or Network Time Protocol (NTP).
- [*] It can be used where a large number of clients are connected and where real-time error correction isn't necessary, such as gaming, voice or video conferencing, and streaming media.

Introduction to Reverse UDP shell

Spoiler ([Click to View](#)) As you already know, a UDP protocol is a connectionless protocol, i.e. there won't be a three-way handshake or a retransmission packets communication between the server and the client. Due to the unreliable nature of this protocol, only a few selected protocols and network services use UDP as its base protocol on L4 (Refer to Layer 4 of the OSI Layer). However, the unreliable nature of this protocol makes it even faster and more efficient than TCP.

When talking about a reverse shell, the UDP protocol could help you out in tight situations where you're stuck with an egress filtration check in the network or an isolated environment from where an outside TCP reverse shell is blocked. There are Some network protocols such as NTP, DNS, DHCP, etc. that uses UDP protocol for communication. As these protocols are required by the network for maintenance (SNMP, DHCP, etc.) and queries (NNP, NTP, etc.), an organization has to allow some UDP based protocols to connect to the outside network. Using this exit door, we can bypass the egress filtration restrictions on the network and can get a reverse shell on our server.

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.

For starters, we can run ncat on our attacker machine to listen on port 3131/udp for the incoming connection from the target machine.

On the target machine, let's execute the command with the -u option to enable the support for UDP protocol during communication to the attacker machine.

If everything goes well you'll see a reverse connection on your system (attacker machine) with the opened command shell in it.

Note: As we're just learning the basics, there's no Deep Packet Inspection (DPI), hardware firewall, or any kind of protocol restriction in place right now.

Analysing UDP Reverse Shells using Wireshark

Spoiler ([Click to View](#)) Now that you have got some basic idea of how to get a reverse shell (cmd.exe) over UDP, let's dig deeper into the packet level to understand what actually happened. When the command is executed on the target machine, there won't be any kind of pre-connection packets for initiating the connection. You'll see only the UDP packets with the cmd.exe banner in it. No SYNs, No ACKs. Just two packets with the banner split into two due to CRLF.

On executing the 'whoami' command on this reverse shell, you can see in the following screenshot, a UDP packet (7 bytes) sent from the attacker machine to the target machine with the command in it.

The output for the executed command is then sent over to the attacker machine by the target in a single UDP packet (21 bytes)

References:

<https://searchnetworking.techtarget.com/definition/UDP-User-Datagram-Protocol>

Hope everyone likes it. Will post a bonus part soon. Stay connected and stay safe! Cheers ?

de_g3ntl3m4n - May 1, 2020, 12:48 am

Thanks again man keep it up cant wait to see more tuts man much respect

BlueSkull - July 2, 2020, 3:21 pm

That is an amazing tutorial. I really learned a lot from it.

Thanks for the amazing information

13uNkNoWn - July 11, 2020, 12:32 am

Great work overall!

However, similar to part two, as this is just someone manually executing NC on the server/client this is just a direct connection.

Part 3 is just a UDP connection instead of a TCP connection, but otherwise the same...

You are missing any explanation of "Step 1" from this image, which makes your example not a reverse shell:

You mention that a step 1 exists, but do not include it in the example.

More of a great example of the difference between TCP/UDP server/client connections.
But not really much about an actual reverse shell.

ceR0 - July 22, 2020, 8:25 am

Thanks for this guide men!

I would really love to see how to bypass IDS or IPS from outside the network.

Thanks!

TheCyb3rAlpha - August 2, 2020, 1:27 pm

(Jul 11, 2020 12:32 AM)13uNkNoWn Wrote: [\[CITE\]](#) Great work overall!

However, similar to part two, as this is just someone manually executing NC on the server/client this is just a direct connection.

Part 3 is just a UDP connection instead of a TCP connection, but otherwise the same...

You are missing any explanation of "Step 1" from this image, which makes your example not a reverse shell:

You mention that a step 1 exists, but do not include it in the example.

More of a great example of the difference between TCP/UDP server/client connections.
But not really much about an actual reverse shell.
Duly noted! Thanks for the feedback. Much appreciated!

SIM-4718838 - *December 22, 2020, 2:25 pm*

thank you for this quick series ?

SIM-5382664 - *July 2, 2023, 9:45 am*

thanks man appreciate it