

Intro to Cyber

XE101

v1.3



THINKCYBER

Table of Contents

The Language of Computers: Understanding Binary	5
Digital Sizes	5
Binary System.....	6
Introduction to ASCII.....	7
Converting Binary to ASCII	10
Converting ASCII to Binary	11
IP Addresses Made Simple.....	12
What is IANA?	12
How are IPv4 addresses allocated?.....	13
Why is IPv4 address space running out?.....	13
IPv4 Address Structure.....	13
Public IP Addresses	13
Private IP Addresses.....	13
Automatic Private IP Addressing (APIPA).....	14
Loopback Addresses.....	14
The Classes of IP addresses.....	15
NAT (Network Address Translation)	15
MAC Addresses	17
MAC Address Structure.....	17
Finding the Vendor Using the MAC Address.....	17
Subnet Mask	18
Subnet to CIDR	18
The Essentials of Network.....	20
Network Services	20
Network Protocols	21
Port Numbers.....	22
Default Gateway vs. Router: Understanding the Difference	23
Switch vs. Hub: Understanding the Differences.....	25
Understanding the Fundamentals of TCP and UDP	26
What are TCP and UDP?.....	26
What are the key differences between TCP and UDP?	26
What are the advantages of using TCP?.....	27
What are the advantages of using UDP?.....	27
Which protocol should I use?.....	27
3-Way Handshake	27

Shodan: The Search Engine for Hackers.....	29
What is Shodan Used For?	29
How Shodan Works.....	29
Crafting Shodan Queries and Examples	30
Working with WHOIS Queries.....	32
Unlocking the Power of Google Dorks	34
Introduction	34
Understanding Google Dorks	34
Basic Google Search Operators.....	34
Combining Google Search Operators.....	35
Ethical Considerations and Best Practices.....	35
Understanding Hex, Base64, and Hashing	36
Hexadecimal.....	36
Hexadecimal Basics	36
Hexadecimal Encoding and Decoding Examples.....	37
Base64 Encoding and Decoding	37
Cryptographic Hash Functions: Exploring MD4, MD5, and SHA Algorithms.....	38
MD4 and MD5.....	39
SHA (Secure Hash Algorithm) Family	39
Practical Uses of Hash Functions.....	39
The Differences	39
An Introduction to Windows Command-Line (CMD).....	41
Understanding of Windows Command Prompt.....	41
Basic Usage of Windows Command Prompt.....	41
Internet Commands in Windows Command Prompt.....	42
Common File Extensions	47
Understanding the OSI Model.....	48
Layer 1: Physical Layer.....	48
Layer 2: Data Link Layer	48
Layer 3: Network Layer	48
Layer 4: Transport Layer.....	49
Layer 5: Session Layer	49
Layer 6: Presentation Layer	49
Layer 7: Application Layer	49
Inspecting Network Traffic using Wireshark	50
What is Wireshark?.....	50

How does Wireshark work?	50
What can you do with Wireshark?.....	50
How to use Wireshark.....	50
Basic Filters to Know	52
Extracting Objects and Files from Pcap Files.....	52
Remote Services.....	53
Introduction	53
HTTP (HyperText Transfer Protocol).....	53
FTP (File Transfer Protocol)	53
RDP (Remote Desktop Protocol)	54
SSH (Secure Shell)	54
How to Connect to Remote Services	55
Connecting to FTP	55
Connecting to RDP	59
Connecting to SSH.....	62

The Language of Computers: Understanding Binary

In the realm of computer science, numeral systems play a crucial role in representing and manipulating data. The most commonly used numeral systems in computing are binary, octal, and hexadecimal. These systems are essential for understanding how computers process and store information.

Digital Sizes

Bit

The smallest unit of digital information is called a bit, which can have a value of either 0 or 1. It's used to represent the most basic information in computers and digital communication.

Byte

A byte is a unit of digital information that consists of 8 bits. It's the most common unit of measurement for digital storage and data transfer. Bytes are used to measure file sizes, memory capacity, and network bandwidth.

Kilobyte

A kilobyte (KB) is a unit of digital information equal to 1,024 bytes. It's often used to measure the size of small files, such as documents, images, and songs.

Megabyte

A megabyte (MB) is a unit of digital information equal to 1,024 kilobytes or 1,048,576 bytes. It's commonly used to measure the size of larger files, such as videos, high-resolution images, and software programs.

Gigabyte

A gigabyte (GB) is a unit of digital information equal to 1,024 megabytes, or 1,073,741,824 bytes. It's commonly used to measure the size of very large files, such as high-definition movies, large databases, and video games.

Terabyte

A terabyte (TB) is a unit of digital information equal to 1,024 gigabytes, or 1,099,511,627,776 bytes. It's commonly used to measure the size of extremely large files, such as 4K videos, scientific data, and enterprise-level storage systems.

Binary System

The binary system, also known as the base-2 numeral system, uses only two symbols: 0 and 1. It forms the foundation of digital computing, as it corresponds directly to the two states of a digital circuit: on (1) and off (0). The binary system is used to represent data, perform calculations, and control logic operations in computers.

In the binary system, each digit's position corresponds to a power of 2, with the rightmost digit representing 2^0 (1), the next digit to the left representing 2^1 (2), and so on. To convert a binary number to its decimal equivalent, we simply add the products of each binary digit and its corresponding power of 2. For example, the binary number 1101 can be converted to decimal as follows: $(1 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (1 \times 2^0) = 8 + 4 + 0 + 1 = 13$.

Example: Let's take the binary number 1101 and convert it to decimal. To do this, we will multiply each digit by the power of 2 corresponding to its position, starting from the rightmost digit (the least significant bit). Here's the process:

$$\begin{aligned} &(1 * 2^3) + (1 * 2^2) + (0 * 2^1) + (1 * 2^0) \\ &= (1 * 8) + (1 * 4) + (0 * 2) + (1 * 1) \\ &= 8 + 4 + 0 + 1 \\ &= 13 \end{aligned}$$

So, the binary number 1101 is equal to the decimal number 13.

Example 2: Take the binary number 101010 and convert it to decimal. To do this, we will multiply each digit by the power of 2 corresponding to its position, starting from the rightmost digit (the least significant bit). Here's the process:

$$\begin{aligned} &(1 * 2^5) + (0 * 2^4) + (1 * 2^3) + (0 * 2^2) + (1 * 2^1) + (0 * 2^0) \\ &= (1 * 32) + (0 * 16) + (1 * 8) + (0 * 4) + (1 * 2) + (0 * 1) \\ &= 32 + 0 + 8 + 0 + 2 + 0 \\ &= 42 \end{aligned}$$

So, the binary number 101010 is equal to the decimal number 42.

Introduction to ASCII

The American Standard Code for Information Interchange (ASCII) is a character encoding standard that represents text in computers, communication equipment, and other devices that use text. It was first developed in the 1960s as a standardized way to represent characters on electronic devices, including letters, numbers, punctuation marks, and some control characters.

ASCII is a 7-bit character encoding, which means it can represent 2^7 or 128 different characters. The characters are assigned numeric values, ranging from 0 to 127. These values are used to represent the characters in binary form, with each value represented by a unique 7-bit binary number.

Dec	Char	Dec	Char	Dec	Char	Dec	Char
0	NUL (null)	32	SPACE	64	@	96	`
1	SOH (start of heading)	33	!	65	A	97	a
2	STX (start of text)	34	"	66	B	98	b
3	ETX (end of text)	35	#	67	C	99	c
4	EOT (end of transmission)	36	\$	68	D	100	d
5	ENQ (enquiry)	37	%	69	E	101	e
6	ACK (acknowledge)	38	&	70	F	102	f
7	BEL (bell)	39	'	71	G	103	g
8	BS (backspace)	40	(72	H	104	h
9	TAB (horizontal tab)	41)	73	I	105	i
10	LF (NL line feed, new line)	42	*	74	J	106	j
11	VT (vertical tab)	43	+	75	K	107	k
12	FF (NP form feed, new page)	44	,	76	L	108	l
13	CR (carriage return)	45	-	77	M	109	m
14	S0 (shift out)	46	.	78	N	110	n
15	SI (shift in)	47	/	79	O	111	o
16	DLE (data link escape)	48	0	80	P	112	p
17	DC1 (device control 1)	49	1	81	Q	113	q
18	DC2 (device control 2)	50	2	82	R	114	r
19	DC3 (device control 3)	51	3	83	S	115	s
20	DC4 (device control 4)	52	4	84	T	116	t
21	NAK (negative acknowledge)	53	5	85	U	117	u
22	SYN (synchronous idle)	54	6	86	V	118	v
23	ETB (end of trans. block)	55	7	87	W	119	w
24	CAN (cancel)	56	8	88	X	120	x
25	EM (end of medium)	57	9	89	Y	121	y
26	SUB (substitute)	58	:	90	Z	122	z
27	ESC (escape)	59	;	91	[

The ASCII character set is divided into two main categories:

- Control characters (0-31): These characters are non-printable and used to control devices, such as printers and terminals. Examples include the carriage return (CR), line feed (LF), and tab (TAB).
- Printable characters (32-127): These characters include uppercase and lowercase letters, digits, punctuation marks, and special characters like the space, exclamation mark, and at symbol (@).

ASCII Table

An ASCII table is a visual representation of the ASCII character set, displaying the characters along with their corresponding decimal, hexadecimal, and binary values.

For example:

The character 'A' has a decimal value of 65, a hexadecimal value of 41, and a binary value of 0100001. The character 'a' has a decimal value of 97, a hexadecimal value of 61, and a binary value of 1100001.

Example:

To convert the word "Cyber" into binary using ASCII, we'll first find the ASCII decimal value of each character, then convert those decimal values into binary. Here's the step-by-step process.

1. Determine the ASCII decimal values of each character.

- **C:** 67
- **y:** 121
- **b:** 98
- **e:** 101
- **r:** 114

2. Convert the decimal values to binary.

- C (67): 1000011
- y (121): 1111001
- b (98): 1100010
- e (101): 1100101
- r (114): 1110010

3. Combine the binary values.

Cyber: 1000011 1111001 1100010 1100101 1110010

Here's a detailed explanation of the steps:

- **Step 1:** Consult an ASCII table to find the decimal values of each character. In this case, we looked up 'C', 'y', 'b', 'e', and 'r' and found their corresponding decimal values as 67, 121, 98, 101, and 114.
- **Step 2:** Convert the decimal values to their binary equivalents. To do this, we'll use the method of dividing the decimal number by 2 and keeping track of the remainder.

For example, let's convert the decimal value 67 (for the character 'C') to binary:

- 67 divided by 2 is 33 with a remainder of 1. Write down the remainder (1).
- 33 divided by 2 is 16 with a remainder of 1. Write down the remainder (1).
- 16 divided by 2 is 8 with a remainder of 0. Write down the remainder (0).
- 8 divided by 2 is 4 with a remainder of 0. Write down the remainder (0).
- 4 divided by 2 is 2 with a remainder of 0. Write down the remainder (0).
- 2 divided by 2 is 1 with a remainder of 0. Write down the remainder (0).

We're left with 1, which is less than 2, so write down the final value (1).

Now read the remainder from bottom to top: 1000011. This is the binary representation of the decimal value 67.

Repeat this process for the remaining characters: 'y' (121), 'b' (98), 'e' (101), and 'r' (114).

- **Step 3:** Combine the binary values of each character to represent the word "Cyber" in binary. Separate each character's binary value with space for readability:

Cyber: 1000011 1111001 1100010 1100101 1110010

Converting Binary to ASCII

Imagine you receive a secret message made up of a series of 1s and 0s, and you need to decode it into a readable text. The process of converting binary (the 1s and 0s) to text, specifically using the ASCII system, involves a few simple steps.

ASCII is like a secret decoder ring that computers use to represent text. It assigns a unique number to each character, like letters, numbers, and punctuation marks. By converting the binary code to these numbers, we can figure out which characters they represent.

Here's a simplified explanation of the process for non-technical people:

1. **Break the binary code into groups:** First, we separate the long string of 1s and 0s into smaller groups. Each group will have 7 or 8 digits, depending on the encoding (for simplicity, let's assume 8-digit groups). For example, if we have the binary code "1000011 1111001", we'll break it into two groups: "1000011" and "1111001".
2. **Convert binary groups to numbers:** Now we need to convert each binary group into a number. To do this, we assign a value to each position in the group, starting from the right. The rightmost position has a value of 1, the next one to the left has a value of 2, then 4, 8, 16, and so on. We add up the values of the positions that have a "1" in them.

For example, let's convert "1000011" to a number:

$$(1 * 64) + (0 * 32) + (0 * 16) + (0 * 8) + (0 * 4) + (1 * 2) + (1 * 1) = 64 + 2 + 1 = 67$$

3. **Use the ASCII table to find the characters:** Now that we have numbers for each binary group, we can use the ASCII table to find the corresponding characters. For example, the number 67 in the ASCII table represents the letter "C".

Using this process, we can decode the entire binary message into text. In our example, "1000011 1111001" would be translated to "Cy" using the ASCII system.

Converting ASCII to Binary

Binary code uses only two digits - 0 and 1 - to represent data. It is the language that computers understand, and all data is ultimately represented in binary form. In binary code, each digit is called a bit, and 8 bits make up a byte.

To convert an ASCII character to binary, follow these steps:

Step 1: Determine the ASCII code for the character you want to convert. You can find ASCII code charts online.

For example, let's say we want to convert the letter "B" to binary. The ASCII code for "B" is 66.

Step 2: Convert the ASCII code to binary. To do this, divide the ASCII code by 2 repeatedly, recording the remainder each time. Keep going until the quotient is 0.

For "B," we divide 66 by 2:

$$\begin{array}{l} 66 / 2 = 33 \text{ remainder } 0 \\ 33 / 2 = 16 \text{ remainder } 1 \\ 16 / 2 = 8 \text{ remainder } 0 \\ 8 / 2 = 4 \text{ remainder } 0 \\ 4 / 2 = 2 \text{ remainder } 0 \\ 2 / 2 = 1 \text{ remainder } 0 \\ 1 / 2 = 0 \text{ remainder } 1 \end{array}$$

So the binary code for "B" is 01000010.

Another example would be the character "@" which has an ASCII code of 64. To convert it to binary, we would divide 64 by 2:

$$\begin{array}{l} 64 / 2 = 32 \text{ remainder } 0 \\ 32 / 2 = 16 \text{ remainder } 0 \\ 16 / 2 = 8 \text{ remainder } 0 \\ 8 / 2 = 4 \text{ remainder } 0 \\ 4 / 2 = 2 \text{ remainder } 0 \\ 2 / 2 = 1 \text{ remainder } 0 \\ 1 / 2 = 0 \text{ remainder } 1 \end{array}$$

So the binary code for "@" is 01000000.

Using Binary Code

Now that we have our binary code, what can we do with it? Binary code is used in a variety of computing operations, including arithmetic and logic operations. By representing data in binary form, computers can perform calculations quickly and accurately.

In addition, binary code is used for data storage and transmission. When you save a file on your computer, it is stored in binary form on the hard drive. When you send an email, the text is converted to binary code and transmitted over the internet.

IP Addresses Made Simple

In the world of the internet, every device that connects to the internet is identified by a unique IP address. This is how data is sent and received across the internet - each device has its own address, and data is sent from one address to another.

IPv4 (Internet Protocol version 4) is one of the most widely used versions of the internet protocol, and it is what most people are referring to when they talk about IP addresses. IPv4 addresses are 32-bit addresses, which means that there are 2^{32} possible addresses in the IPv4 space.

But with so many devices connecting to the internet, how does the internet keep track of all these addresses? This is where the IANA (Internet Assigned Numbers Authority) comes in.

What is IANA?

IANA stands for Internet Assigned Numbers Authority. It is a department of ICANN (Internet Corporation for Assigned Names and Numbers) that is responsible for managing the global allocation of IP addresses, domain names, protocol numbers, and other unique identifiers that are used on the Internet.



The main functions of IANA include:

- **IP address allocation:** IANA allocates IP addresses to the five Regional Internet Registries (RIRs) that manage IP addresses for different regions of the world.
- **Domain name system (DNS) management:** IANA manages the root zone of the DNS, which includes the delegation of top-level domains (TLDs) such as .com, .org, and .net.
- **Protocol number allocation:** IANA assigns unique numbers to Internet protocols such as TCP/IP, HTTP, and SMTP.
- **Maintaining registries:** IANA maintains various registries of Internet parameters, such as the list of TLDs, port numbers, and MIME types.

IANA plays a crucial role in ensuring the stability and security of the Internet. Without IANA's management and allocation of IP addresses, domain names, and protocol numbers, it would be difficult for devices and systems to communicate with each other effectively on the global network.

How are IPv4 addresses allocated?

The IANA allocates IPv4 addresses to the RIRs in blocks of /8, which contain approximately 16.8 million addresses each. The RIRs then allocate these blocks to ISPs and other organizations within their region, who in turn allocate smaller blocks to their customers.

IPv4 addresses are assigned based on geographic regions, and each region has its own RIR. The five RIRs are:

- AfriNIC (African Network Information Center)
- APNIC (Asia-Pacific Network Information Centre)
- ARIN (American Registry for Internet Numbers)
- LACNIC (Latin American and Caribbean IP Address Regional Registry)
- RIPE NCC (Réseaux IP Européens Network Coordination Centre)

Why is IPv4 address space running out?

One of the biggest challenges facing the internet today is the exhaustion of the IPv4 address space. With so many devices connecting to the internet, the demand for IP addresses has grown exponentially, and the 32-bit IPv4 address space is simply not enough to meet this demand.

As a result, the internet has been transitioning to IPv6, which uses 128-bit addresses and has a much larger address space. However, the transition to IPv6 is a slow process, and IPv4 will likely continue to be used for many years to come.

IPv4 Address Structure

An IPv4 address is a 32-bit numerical identifier, typically displayed as four decimal numbers separated by periods (e.g., 192.168.0.1). Each decimal number, known as an octet, represents 8 bits and can range from 0 to 255. The address space of IPv4 allows for approximately 4.3 billion unique addresses.

Public IP Addresses

Public IP addresses are assigned by the Internet Assigned Numbers Authority (IANA) and regional registries. They are globally unique and routable on the internet, enabling devices to communicate with each other across different networks. Public IPs are typically assigned to web servers, email servers, or devices that require direct access to the internet.

Private IP Addresses

Private IP addresses are used for internal communication within private networks, such as home or office networks. They are not routable on the internet and must be translated into public IP addresses using Network Address Translation (NAT) to communicate with devices on the internet.

The following IP ranges are reserved for private use:

- 10.0.0.0 to 10.255.255.255
- 172.16.0.0 to 172.31.255.255
- 192.168.0.0 to 192.168.255.255

Automatic Private IP Addressing (APIPA)

APIPA is a feature in Windows operating systems that automatically assigns a private IP address to a device when it cannot obtain one from a Dynamic Host Configuration Protocol (DHCP) server. APIPA addresses are self-assigned and fall within the range 169.254.0.1 to 169.254.255.254, with a subnet mask of 255.255.0.0.

Although APIPA allows devices to communicate within a local network, it does not provide internet connectivity. If the device later obtains a valid IP address from a DHCP server, it will replace the APIPA address.

Loopback Addresses

Loopback addresses are a special type of IP address that is used to send data to the same device that generated it. Loopback addresses are a fundamental part of networking, and they are used for a variety of purposes.

What is Loopback Used For?

Loopback addresses are used for a variety of purposes in networking. One of the most common uses of loopback addresses is for testing and troubleshooting network connectivity. By sending data to a loopback address, a device can test whether its network interface is functioning properly.

Loopback addresses are also used in applications that require a network connection but do not need to communicate with other devices on the network. For example, some applications use a loopback address to communicate with a local service or database.

Loopback Commands

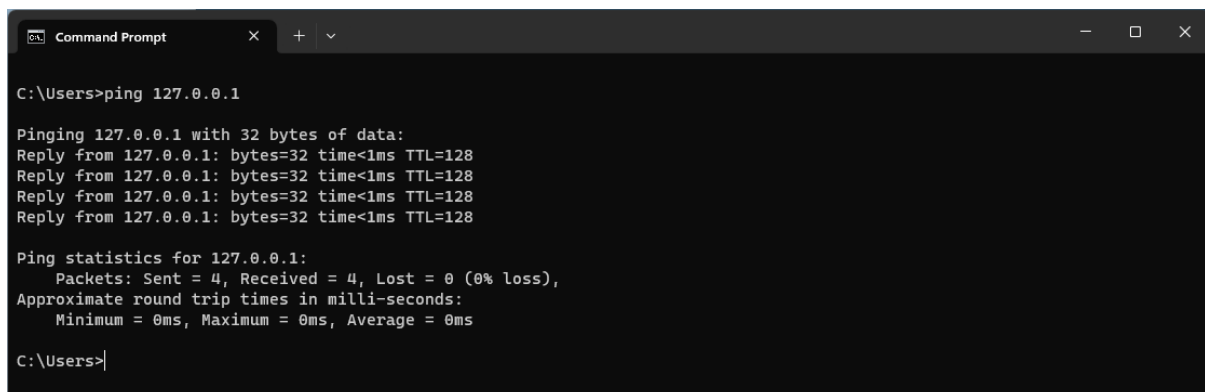
There are several commands that are used to work with loopback addresses on different operating systems. Here are some of the most used commands for Windows:

ipconfig: This command displays information about the network interfaces on a Windows system, including any loopback interfaces that are configured.

ping 127.0.0.1: This command sends a ping request to the loopback address, allowing you to test the connectivity of the loopback interface.

Loopback Address Ranges

Loopback addresses are assigned to a special range of IP addresses that are reserved for this purpose. In IPv4, the loopback address range is 127.0.0.0/8, which includes all addresses from 127.0.0.1 to 127.255.255.255.



```
Command Prompt
C:\Users>ping 127.0.0.1

Pinging 127.0.0.1 with 32 bytes of data:
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128

Ping statistics for 127.0.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users>
```


The Classes of IP addresses

Class A: This class of IP addresses uses the first octet to represent the network portion, and the remaining three octets to represent the host portion. Class A addresses range from 1.0.0.0 to 126.0.0.0 and are intended for very large networks.

Class B: This class of IP addresses uses the first two octets to represent the network portion, and the remaining two octets to represent the host portion. Class B addresses range from 128.0.0.0 to 191.255.0.0 and are intended for medium-sized networks.

Class C: This class of IP addresses uses the first three octets to represent the network portion, and the remaining octet to represent the host portion. Class C addresses range from 192.0.0.0 to 223.255.255.0 and are intended for small networks.

Class D: This class of IP addresses is used for multicasting. Multicast addresses begin with the binary pattern of 1110 in the first four bits of the first octet. Class D addresses range from 224.0.0.0 to 239.255.255.255.

Class E: This class of IP addresses is reserved for experimental use and is not used for general networking purposes. Class E addresses range from 240.0.0.0 to 255.255.255.255.

Max Hosts per Class Network

The three original classes of IP addresses are Class A, Class B, and Class C, each of which has a different range of IP addresses and a different number of hosts that can be assigned to each network.

Class A networks have a range of IP addresses from 1.0.0.0 to 126.0.0.0. The first octet is reserved for the network identifier, and the remaining three octets are available for hosts. This means that a Class A network can have up to 16,777,214 ($2^{24} - 2$) hosts.

Class B networks have a range of IP addresses from 128.0.0.0 to 191.255.0.0. The first two octets are reserved for the network identifier, and the remaining two octets are available for hosts. This means that a Class B network can have up to 65,534 ($2^{16} - 2$) hosts.

Class C networks have a range of IP addresses from 192.0.0.0 to 223.255.255.0. The first three octets are reserved for the network identifier, and the remaining octet is available for hosts. This means that a Class C network can have up to 254 ($2^8 - 2$) hosts.

NAT (Network Address Translation)

Network Address Translation is a technique used to allow multiple devices in a home or office network to share a single internet connection.

Imagine that your home network is like a private community with each device having its own address. When you want to access the internet, your request needs to be sent out using a unique address that is recognized by the internet. But, your internet service provider (ISP) only provides you with one public address, and it's not enough for all of your devices to use.

This is where NAT comes in. NAT allows your home network to communicate with the internet using a single public address, provided by your ISP. The NAT device acts as a middleman between your devices and the internet.

When one of your devices sends a request to the internet, the NAT device replaces the private address of your device with the public address provided by your ISP. This allows your request to be routed over the internet using the public address, which is recognized by the internet. When a response is received from the internet, the NAT device sends it back to the requesting device on your home network.

NAT provides several benefits, including improved security by hiding the private addresses of devices on the network from the internet, conservation of public addresses, and simplification of network configuration.

In summary, NAT is a technique that allows multiple devices in your home or office network to access the internet using a single public address. This makes it possible for you to share your internet connection among several devices without needing multiple public addresses.

MAC Addresses

Media Access Control (MAC) addresses are unique identifiers assigned to network interface cards (NICs) by their manufacturers. They play a crucial role in communication within local network segments, allowing devices to identify each other and ensure that data packets are transmitted to their intended recipients.

MAC Address Structure

A MAC address consists of 48 bits, typically displayed in hexadecimal format and separated by colons or hyphens (e.g., 01:23:45:67:89:AB).

Finding the Vendor Using the MAC Address

To identify the vendor of a device based on its MAC address, you can use the OUI portion of the address. By looking up the OUI in an online database or using a tool that has access to an updated OUI list, you can determine which company manufactured the network interface card.

Here are a few methods to find the vendor using a MAC address:

- **IEEE OUI Lookup:** The IEEE maintains an up-to-date list of OUI assignments, which can be accessed through their website. By entering the OUI portion of the MAC address, you can retrieve information about the manufacturer.
- **Online MAC Address Lookup Tools:** Several websites and online tools allow you to enter a MAC address and quickly retrieve information about the vendor. Examples include Wireshark's OUI Lookup Tool and MAC Vendor Lookup.

To view the vendor of a device using its MAC address is to use an online MAC address lookup tool. These tools allow you to enter the MAC address and then display information about the device, including the vendor. For example, let's say we want to find the vendor of the MAC address "00:11:22:33:44:55". We can use an online MAC address lookup tool, such as <https://macaddress.io> and enter the MAC address into the search field. The tool will then display information about the device, including the vendor, which in this case is "Cimsys Inc".

Vendor details		Block details		MAC address details	
OUI	00:11:22	Is registered	True	Is valid	True
Is private	False	Border left	00:11:22:00:00:00	Virtual Machine	Not detected
Company name	Cimsys Inc	Border right	00:11:22:FF:FF:FF	Transmission type	Unicast
Company address	#301,Sinsung-clean BLDG,140, Nongseo- Ri,Kiheung-Eup Yongin- City Kyunggi-Do 449- 711 KR	Block size	16,777,216	Administration type	UAA
Country code	KR	Assignment block size	MA-L	Applications	Not detected
		Date created	05 June 2004	Wireshark notes	No details
		Date updated	27 September 2015		

Subnet Mask

A subnet mask is a 32-bit number that is used to determine which part of an IP address belongs to the network and which part belongs to the host. It works by masking off the network portion of an IP address to identify the host portion. In IPv4, subnet masks are commonly used to divide a network into smaller sub-networks, allowing for better organization and management of network resources.

CIDR (Classless Inter-Domain Routing) notation is a compact representation of an IP address and subnet mask. CIDR notation consists of the IP address followed by a slash (/) and a number that indicates the number of network bits in the subnet mask. For example, 192.168.0.0/24 represents an IPv4 address with a subnet mask of 255.255.255.0.

/8, /16, and /24 are common subnet mask values used to create smaller sub-networks within a larger network.

A subnet mask of /8 (255.0.0.0) creates a very large network with up to 16,777,216 hosts. This type of network is typically used by large organizations or internet service providers.

A subnet mask of /16 (255.255.0.0) creates a moderately sized network with up to 65,536 hosts. This type of network is often used by mid-sized organizations or for regional network deployments.

A subnet mask of /24 (255.255.255.0) creates a smaller network with up to 256 hosts. This type of network is commonly used for small office or home office (SOHO) networks, as well as for point-to-point connections.

Subnet to CIDR

When connecting devices to a network, it's essential to organize them efficiently, so they can communicate with each other and share resources. To do this, we use a concept called subnetting, which helps us divide a network into smaller parts called subnets. One way to represent these subnets is by using CIDR notation.

CIDR stands for Classless Inter-Domain Routing, and it's a shorthand way of representing a subnet. In simpler terms, it tells us how many devices can fit in a subnet without getting into the technical details of binary and network masks.

Example 1: Subnet mask: 255.255.255.0

To convert this subnet mask to CIDR notation, we need to count the number of consecutive '1's in the binary representation of the subnet mask. Let's break this down:

1. Convert each decimal number in the subnet mask to binary:

255: 11111111

255: 11111111

255: 11111111

0: 00000000

2. Combine the binary numbers: 11111111 11111111 11111111 00000000
3. Count the number of consecutive '1's: There are 24 consecutive '1's.
4. Write the CIDR notation: /24

So, the subnet mask 255.255.255.0 can be represented as CIDR notation /24.

Example 2: Subnet mask: 255.255.254.0

1. Following the same process as before - convert each decimal number in the subnet mask to binary:

255: 11111111

255: 11111111

254: 11111110

0: 00000000

2. Combine the binary numbers: 11111111 11111111 11111110 00000000
3. Count the number of consecutive '1's: There are 23 consecutive '1's.
4. Write the CIDR notation: /23

So, the subnet mask 255.255.254.0 can be represented as CIDR notation /23.

In summary, converting a subnet mask to CIDR notation involves converting the subnet mask to binary, counting the number of consecutive '1's, and representing that count with a slash followed by the number. This simplified notation helps us understand and organize networks more easily, even for those who are not familiar with the technical aspects of networking.

The Essentials of Network

In today's world, where we rely heavily on the internet, there are several technical terms that we might encounter. These terms might sound like jargon or alphabet soup to some people, but they are actually important components that help our devices connect to the internet and communicate with each other.

Network Services

Let's start with DNS, which stands for Domain Name System. Every website you visit has a unique IP address, which is a series of numbers that identifies the server where the website is hosted. However, IP addresses can be difficult to remember, especially when there are millions of websites on the internet. That's where DNS comes in. DNS translates human-readable domain names (like google.com) into IP addresses that computers can understand. When you type a domain name into your web browser, your computer sends a DNS query to a DNS server, which responds with the IP address of the website you want to visit. This allows your computer to connect to the correct server and load the website you requested.

In simpler terms, DNS acts as a phone book for the internet. Just like how you look up a phone number in a phone book to call someone, your computer looks up a website's IP address in DNS to connect to it.

Moving on to DHCP, which stands for Dynamic Host Configuration Protocol. DHCP is a protocol that automatically assigns IP addresses to devices on a network. In the early days of networking, devices had to be manually configured with IP addresses, which was a time-consuming and error-prone process. With DHCP, devices can automatically obtain an IP address from a DHCP server when they connect to a network. This simplifies the process of setting up and managing a network, especially in large organizations with many devices.

To put it simply, DHCP assigns an IP address to your device when you connect to a network. It's like checking into a hotel and being assigned a room number. With DHCP, you don't have to manually configure your device's IP address every time you connect to a new network.

Finally, let's talk about ARP, which stands for Address Resolution Protocol. ARP is a protocol that helps devices on a network find each other. When a device wants to communicate with another device on the same network, it needs to know the other device's MAC address, which is a unique identifier for the device's network interface. However, devices typically only know each other's IP addresses, not their MAC addresses. ARP allows a device to send a broadcast message on the network asking which device has a particular IP address. The device with that IP address responds with its MAC address, and the requesting device can then use that MAC address to communicate with the other device.

In simpler terms, ARP helps devices on a network find each other. It's like asking a friend for someone's phone number so you can call or text them.

Network Protocols

The internet is a complex ecosystem, where various protocols work together to ensure seamless communication, data exchange, and network management. These protocols provide the foundation for the services and applications we use daily.

SSL (Secure Sockets Layer)

General Use: SSL is a security protocol used to establish encrypted links between a web server and a browser, ensuring secure data transmission. It's widely used to protect sensitive information, such as login credentials, credit card information, and personal data, as it travels across the internet.

Port: SSL uses port 443.

HTTP (Hypertext Transfer Protocol)

General Use: HTTP is the foundation for data communication on the World Wide Web. It enables the transfer of hypertext (text with links to other resources) between a client (usually a web browser) and a server. HTTP is a stateless protocol, meaning each request is treated independently, without any knowledge of previous requests.

Port: HTTP uses port 80.

IMAP (Internet Message Access Protocol)

General Use: IMAP is an email retrieval protocol that enables users to access and manage their email messages on a mail server. Unlike the older POP3 protocol, IMAP allows for multiple clients to access the same mailbox simultaneously, supports folders and message flagging, and keeps messages on the server, making it ideal for managing email across multiple devices.

Port: IMAP uses port 143 for unencrypted connections and port 993 for encrypted (IMAP over SSL/TLS) connections.

ICMP (Internet Control Message Protocol)

General Use: ICMP is a network-layer protocol used primarily for diagnostic and error-reporting purposes. It provides tools like ping and traceroute, which help administrators and users troubleshoot network issues by sending and receiving control messages. For example, ICMP can report when a requested IP address is unreachable or when a packet's time-to-live (TTL) has expired.

Port: As a network-layer protocol, ICMP does not use a specific port but is encapsulated within IP packets.

SNMP (Simple Network Management Protocol)

General Use: SNMP is an application-layer protocol for managing and monitoring network devices, such as routers, switches, and servers. It enables administrators to collect performance data, configure devices, and receive notifications when certain events occur, all via a centralized management system.

Port: SNMP uses port 161 for queries and port 162 for receiving notifications (traps).

NTP (Network Time Protocol)

General Use: NTP is a protocol for synchronizing the clocks of computers and network devices over the internet. Accurate timekeeping is essential for various applications, such as distributed databases, authentication systems, and log file management. NTP uses a hierarchical system of time sources, with more accurate sources at the top.

Port: NTP uses port 123.

SMB (Server Message Block)

General Use: SMB is a network file-sharing protocol that enables applications to read and write to files and request services from server programs on a computer network. It's primarily used on Windows-based networks for sharing files, printers, and other resources, but it's also implemented on other platforms, such as macOS and Linux.

Port: SMB uses port 445 over TCP for direct connections, while legacy systems may use port 139 over NetBIOS.

Understanding these key internet protocols and their associated ports is essential for anyone working with networks, servers, or security. Each protocol plays a critical role in the overall functioning of the internet, enabling seamless communication, data exchange, and network management. As technology continues to evolve, these protocols may be updated or new ones may emerge to meet the demands of increasingly complex and diverse network environments. By staying informed about these protocols, IT professionals and users alike can better understand and navigate the intricate landscape of the internet.

Port Numbers

SSH, which stands for Secure Shell. SSH is a secure protocol for accessing a remote computer over a network. It uses encryption to protect the communication between the two devices, making it a popular choice for system administrators and developers who need to access servers and other network devices remotely. The default port for SSH is 22, but it can be changed to any other available port.

Next is HTTP, which stands for Hypertext Transfer Protocol. HTTP is the foundation of the World Wide Web and is used to transfer web pages and other resources between servers and clients. When you type a URL into your web browser, your computer sends an HTTP request to the server hosting the website, which responds with the requested resource. The default port for HTTP is 80, but it is often encrypted using HTTPS (HTTP Secure), which uses port 443.

RDP, or Remote Desktop Protocol, is a protocol used to access and control a remote computer. It is commonly used by system administrators and support technicians to troubleshoot and fix issues on remote computers. RDP uses port 3389 by default, but it is often used with a VPN (Virtual Private Network) to provide an extra layer of security.

DHCP, or Dynamic Host Configuration Protocol, is a protocol used to automatically assign IP addresses to devices on a network. It simplifies the process of configuring network settings by allowing devices to obtain an IP address, subnet mask, default gateway, and other network settings automatically. DHCP uses ports 67 and 68 for communication between clients and servers.

SMTP, or Simple Mail Transfer Protocol, is a protocol used for sending email messages between servers. When you send an email, your email client communicates with an SMTP server to send the message. SMTP uses port 25 by default, but many email providers also support ports 465 and 587 for encrypted communication.

FTP, or File Transfer Protocol, is a protocol used for transferring files between servers and clients. It is commonly used by web developers to upload and download files to and from web servers. FTP uses ports 20 and 21 for communication, but it is often used with SSL/TLS (Secure Sockets Layer/Transport Layer Security) to encrypt the data being transferred.

Here is a list of the most used ports:

- **Port 80:** HTTP (Hypertext Transfer Protocol)
- **Port 443:** HTTPS (HTTP Secure)
- **Port 22:** SSH (Secure Shell)
- **Port 25:** SMTP (Simple Mail Transfer Protocol)
- **Port 53:** DNS (Domain Name System)
- **Port 3389:** RDP (Remote Desktop Protocol)
- **Port 21:** FTP (File Transfer Protocol)
- **Port 3306:** MySQL Database Server

[Default Gateway vs. Router: Understanding the Difference](#)

When it comes to computer networking, the terms "default gateway" and "router" are often used interchangeably, but they are not the same thing. Understanding the difference between these two terms is important for anyone who wants to set up a network or troubleshoot network issues.

What is a Default Gateway?

A default gateway is an IP address on a network that serves as a route for all traffic that is not destined for a local network. In other words, it's the address of the next hop that a device will send packets to if the destination IP address is not on the local network. The default gateway is usually the IP address of the router that connects the local network to the internet.

For example, suppose you have a computer on a local network with an IP address of 192.168.1.10. If the computer wants to send a packet to a website with an IP address of 192.168.2.10, it will send the packet to the default gateway (i.e., the router), which will then forward the packet to the correct destination.

What is a Router?

A router is a device that connects two or more networks and routes traffic between them. A router operates at the network layer (layer 3) of the OSI model and uses IP addresses to determine the next hop for each packet. Routers can be physical devices, such as the ones provided by internet service providers (ISPs), or they can be virtual devices, such as software routers.

A router can perform a variety of functions, such as:

- Routing traffic between networks.
- Providing network address translation (NAT) to allow multiple devices to share a single public IP address.
- Providing firewall and security features to protect the network.
- Allocating IP addresses to devices on the network.



What's the Difference?

The main difference between a default gateway and a router is their function. A default gateway is the IP address of the next hop for traffic that is not destined for a local network, while a router is a device that connects two or more networks and routes traffic between them. In other words, the default gateway is the address that a device uses to access the internet, while a router is the device that makes that connection possible.

Switch vs. Hub: Understanding the Differences

Switches and hubs are both network devices used to connect multiple devices to a network. While they may look similar and serve a similar purpose, there are some key differences between switches and hubs that can impact network performance and security.

What is a Hub?

A hub is a simple network device that works by broadcasting incoming data packets to all devices connected to it. This means that any data received by the hub is sent to all devices, regardless of whether the data is intended for them. This can result in network congestion and slower performance, as multiple devices are competing for the same data.

What is a Switch?

A switch is a more advanced network device that directs data packets only to the devices for which the data is intended. A switch creates a virtual circuit between the sender and the receiver, and data packets are transmitted only on that circuit. This results in faster and more efficient data transmission, as only the intended devices receive the data. Switches also provide additional security features, such as the ability to segment the network and restrict access to specific devices.



Differences Between Switches and Hubs

Broadcasting: Hubs broadcast incoming data packets to all devices, while switches direct data only to the intended devices. This can result in slower network performance with hubs, as multiple devices are competing for the same data.

- **Efficiency:** Switches are more efficient than hubs, as they create a direct connection between the sender and receiver, resulting in faster data transmission.
- **Security:** Switches provide better security features than hubs, as they can segment the network and restrict access to specific devices. Hubs provide no such security features.
- **Cost:** Hubs are generally less expensive than switches, as they are simpler devices with fewer features.

Choosing the Right Device for Your Network

When choosing between a switch and a hub, it's important to consider the size and needs of your network. If you have a small network with a limited number of devices and don't require advanced security features, a hub may be sufficient. However, if you have a larger network with multiple devices and require faster data transmission and better security, a switch is the better choice.

In summary, switches and hubs both serve a similar purpose of connecting multiple devices to a network, but switches are more advanced and efficient, providing faster data transmission and better security features. Hubs are simpler and less expensive, but can result in slower network performance and provide no security features.

Understanding the Fundamentals of TCP and UDP

In the world of computer networking, TCP and UDP are two important protocols used for communication between different devices. They are responsible for making sure that data is transmitted securely and efficiently across the internet and other networks.

What are TCP and UDP?

TCP (Transmission Control Protocol) and UDP (User Datagram Protocol) are both transport layer protocols that are used to transmit data between devices. TCP is a connection-oriented protocol, while UDP is a connectionless protocol.

In other words, when data is sent over TCP, a connection is established between the sender and receiver before data transmission can begin. This connection ensures that data is transmitted in the correct order and that no data is lost during transmission.

With UDP, there is no connection established between the sender and receiver before data transmission. Instead, data is sent as individual packets, with no guarantee that they will arrive in the correct order or that all packets will be received.

What are the key differences between TCP and UDP?

One of the biggest differences between TCP and UDP is the way in which they handle data transmission. TCP is a reliable protocol that guarantees that all data is transmitted and that it arrives in the correct order. UDP, on the other hand, is an unreliable protocol that does not guarantee that all data will be transmitted or that it will arrive in the correct order.

Another key difference between TCP and UDP is the way in which they handle congestion. TCP uses a congestion control mechanism that helps to ensure that the network is not overloaded with traffic. UDP, on the other hand, does not have any congestion control mechanism and can send data at a much faster rate.

What are the advantages of using TCP?

TCP is a reliable protocol that guarantees that all data is transmitted and that it arrives in the correct order. This makes it ideal for applications where data integrity is critical, such as online banking or e-commerce websites.

TCP is also able to handle congestion in a way that helps to prevent network overload. This is particularly important for applications that require a high degree of reliability, such as streaming video or online gaming.

What are the advantages of using UDP?

UDP is a faster protocol than TCP, as it does not have the overhead of establishing a connection before data transmission can begin. This makes it ideal for applications that require real-time data transmission, such as online voice and video chat.

UDP is also able to send data packets that are much larger than those sent by TCP, which can be useful for applications that require the transfer of large files or datasets.

Which protocol should I use?

The choice of whether to use TCP or UDP depends on the specific requirements of your application. If data integrity and reliability are critical, then TCP is the way to go. If speed and real-time data transmission are more important, then UDP is the better choice.

In general, TCP is used for applications that require a high degree of reliability, such as e-commerce websites and online banking. UDP is used for applications that require real-time data transmission, such as online voice and video chat, and for applications that require the transfer of large files or datasets.

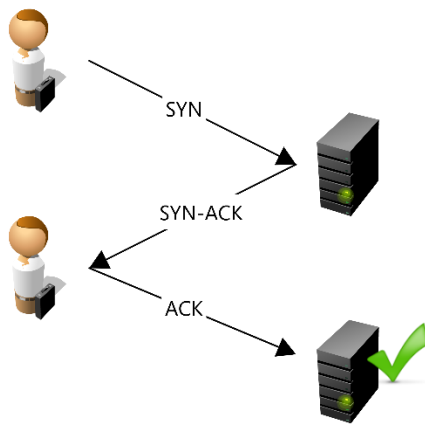
3-Way Handshake

The three-way handshake is a process used by computer networking protocols like TCP (Transmission Control Protocol) to establish a connection between two devices before they start exchanging information. The handshake involves three steps, which is why it's called the "three-way" handshake.

Here's how it works:

1. **SYN (synchronize) packet:** The process begins when one device (let's call it Device A) sends a SYN packet to the other device (Device B). This packet is essentially Device A's way of saying "Hey, I want to talk to you. Can we establish a connection?"
2. **SYN-ACK (synchronize-acknowledge) packet:** When Device B receives the SYN packet, it responds with a SYN-ACK packet. This packet lets Device A know that Device B has received the request and is willing to establish a connection. The SYN-ACK packet contains a random sequence number that Device B generates, as well as an acknowledgement number that lets Device A know which sequence number it's expecting to receive next.
3. **ACK (acknowledge) packet:** Finally, Device A sends an ACK packet back to Device B, letting it know that it received the SYN-ACK packet and is ready to start exchanging information. The

ACK packet contains the sequence number generated by Device B in the previous step, plus one, and another acknowledgement number that lets Device B know which sequence number it's expecting to receive next.



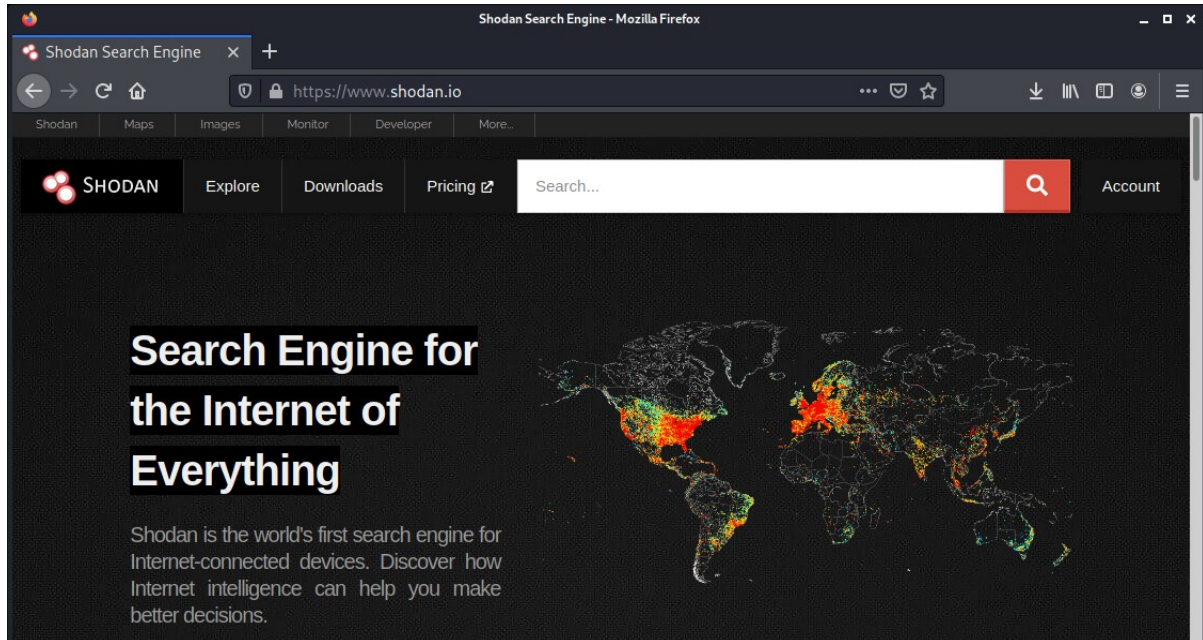
4. Once the three-way handshake is complete, the two devices have established a connection, and they can start exchanging information.

Think of it like a conversation between two people who have never met before. Before they can start talking, they need to introduce themselves and make sure the other person is ready to talk. The three-way handshake is like this introduction process for devices on a network.

In summary, the three-way handshake is a crucial process that allows devices to establish a connection with each other before exchanging information. It's like the "hello" before a conversation, and it ensures that both devices are ready to start communicating.

Shodan: The Search Engine for Hackers

Shodan, often referred to as the "search engine for the Internet of Things (IoT)," is a powerful tool for discovering and analyzing connected devices worldwide. Shodan collects data from devices connected to the internet, providing information on their location, services, operating systems, and potential vulnerabilities.



What is Shodan Used For?

Shodan serves a variety of purposes for different users, including:

- **Security Research:** Professionals use Shodan to identify vulnerable devices and networks, helping them develop and implement security measures to protect against cyber threats.
- **Penetration Testing:** Shodan assists ethical hackers in discovering potential targets and assessing their security posture during penetration testing engagements.
- **Internet Infrastructure Analysis:** Researchers and analysts employ Shodan to study global internet trends, such as the distribution of specific devices or the prevalence of certain vulnerabilities.
- **Asset Discovery and Management:** Organizations utilize Shodan to inventory their internet-facing devices and ensure proper security configurations.
- **Market Research:** Businesses leverage Shodan to analyze competitors' technology deployments and gain insights into industry trends.

How Shodan Works

Shodan operates by scanning the entire IPv4 address space and selected portions of the IPv6 address space, indexing metadata about the devices it encounters. It uses banners, which are metadata that devices transmit when communicating with other devices, to identify the type of device, its operating system, and the services it provides. Shodan indexes this information and makes it searchable for users.

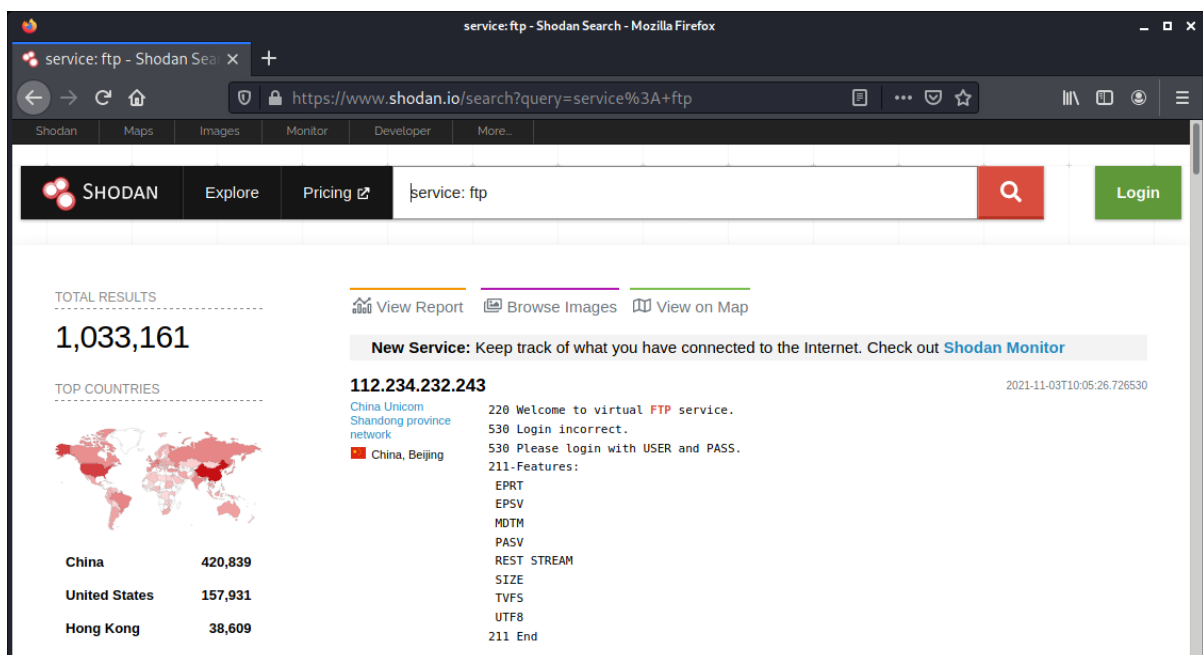
The process involves the following steps:

- **Scanning:** Shodan continuously scans the internet for open ports and protocols, probing devices to collect banners.
- **Indexing:** It then indexes the collected information, organizing it into a searchable database.
- **Searching:** Users can search Shodan's database using various filters and query syntax to identify devices, networks, or vulnerabilities of interest.
- **Analyzing:** Shodan provides additional tools to visualize and analyze the search results, enabling users to gain insights and make informed decisions.

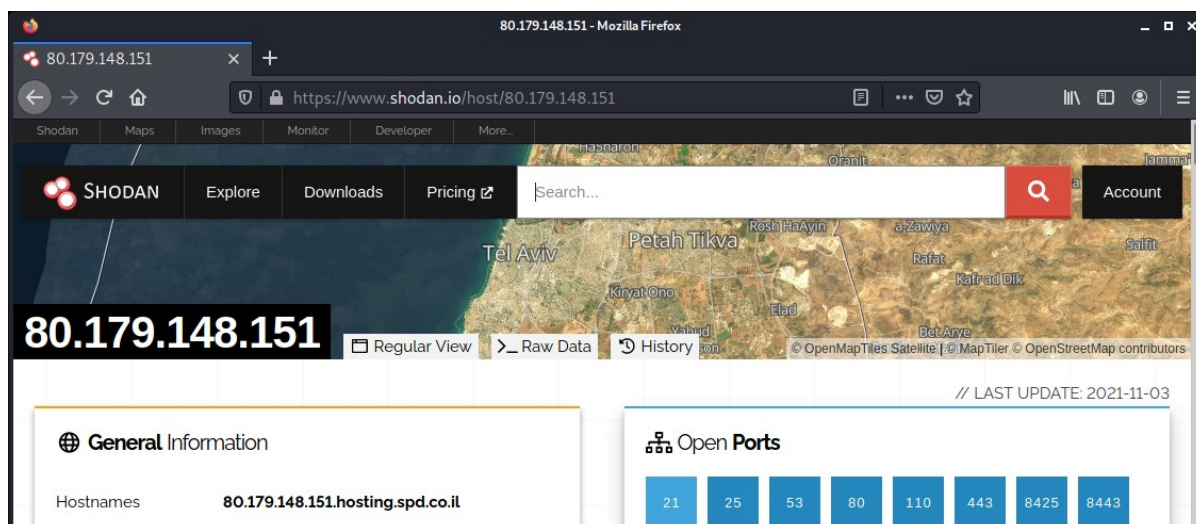
Crafting Shodan Queries and Examples

Shodan's search capabilities allow users to filter results based on different criteria. Some common filters include:

- IP Address (ip)
- Hostname (hostname)
- Operating System (os)
- Port (port)
- Geo-location (geo)
- Organization (org)



The screenshot shows the Shodan search results page for the query 'service: ftp'. The browser address bar shows the URL 'https://www.shodan.io/search?query=service%3A+ftp'. The page displays a search bar with the query 'service: ftp' and a 'Login' button. Below the search bar, there are navigation links for 'View Report', 'Browse Images', and 'View on Map'. The total number of results is 1,033,161. A 'New Service' alert is displayed, indicating a new service found at IP address 112.234.232.243. The alert includes details such as the location (China, Beijing), the organization (China Unicom Shandong province network), and the banner text: '220 Welcome to virtual FTP service. 530 Login incorrect. 530 Please login with USER and PASS. 211-Features: EPRT, EPSV, MDTM, PASV, REST STREAM, SIZE, TVFS, UTF8, 211 End'. A world map shows the top countries with the following counts: China (420,839), United States (157,931), and Hong Kong (38,609).



Here are some example queries to demonstrate Shodan's search capabilities:

Find web servers running on port 80:

`port:80`

Identify devices using the Apache web server:

`product:Apache`

Search for devices running the Windows operating system:

`os:windows`

Discover devices with a specific vulnerability, e.g., Heartbleed:

`vuln:heartbleed`

Locate devices within a particular organization:

`org:"Example Organization"`

Find devices in a specific country, e.g., United States:

`country:US`

As demonstrated, Shodan's search capabilities provide users with powerful insights into the connected devices that make up the internet. Understanding how to craft queries and interpret examples is crucial for researchers, security professionals, and businesses to harness Shodan's full potential.

Working with WHOIS Queries

A WHOIS query is a protocol used to retrieve information about domain names, IP addresses, and other network-related information. The protocol allows users to search a database of registered domain names and IP addresses to obtain details about the owners, registration date, expiration date, and other technical details associated with a particular domain or IP address range.

The WHOIS protocol was first introduced in the early 1980s to help system administrators identify and resolve technical problems related to domain names and IP addresses. However, over time, the protocol became widely used for other purposes, including identifying the owners of domain names and IP addresses for legal, business, and security purposes.

To perform a WHOIS query, a user typically accesses a WHOIS service or database, either through a web-based interface or by using a command-line tool. The user then enters the domain name or IP address of interest, and the WHOIS database returns a set of information related to that domain or IP address.

The information returned by a WHOIS query can vary depending on the type of query and the WHOIS service being used. However, some common pieces of information that may be returned include:

- The domain name or IP address of the website
- The registrar or organization responsible for registering the domain or IP address
- The date the domain or IP address was created and when it will expire
- The name and contact information of the domain or IP address owner
- The name servers associated with the domain or IP address
- The status of the domain or IP address, such as whether it is active or inactive
- The organization or person responsible for maintaining the information in the WHOIS database.

The screenshot shows a web browser window displaying the DomainTools WHOIS lookup page for the IP address 8.8.8.8. The page title is "IP Information for 8.8.8.8". The main content area is titled "Quick Stats" and lists the following information:

IP Location	United States Glenmont Google
ASN	AS15169 GOOGLE, US (registered Mar 30, 2000)
Resolve Host	dns.google
Whois Server	whois.arin.net
IP Address	8.8.8.8
Reverse IP	13,418 websites use this address.

Below the "Quick Stats" section, there is a "NetRange" section with the following details:

NetRange:	8.0.0.0 - 8.127.255.255
CIDR:	8.0.0.0/9
NetName:	LVL-ORG-8-8
NetHandle:	NET-8-0-0-0-1
Parent:	NET8 (NET-8-0-0-0-0)

On the right side of the page, there is a "DomainTools Iris" advertisement and a "Tools" section with the following options:

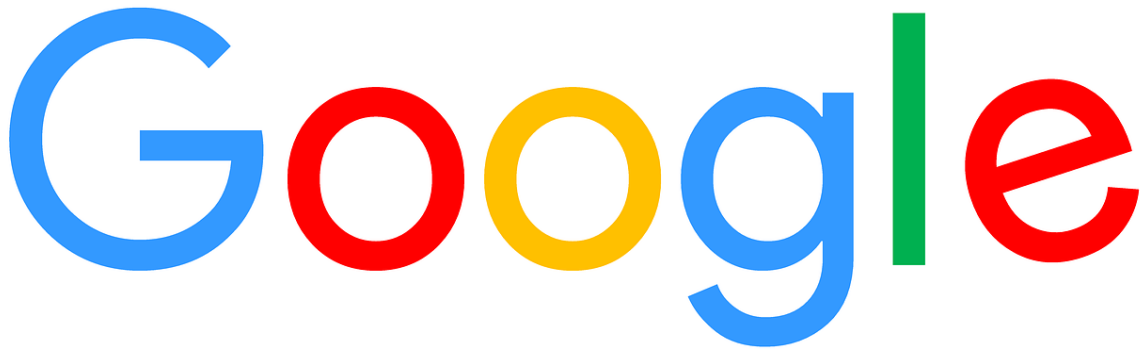
- Monitor Domain Properties
- Reverse IP Address Lookup
- Network Tools

WHOIS queries are commonly used by a variety of stakeholders, including domain name registrars, law enforcement agencies, intellectual property lawyers, and individuals seeking to protect their online privacy. For example, law enforcement agencies may use WHOIS queries to identify the owners of

domain names associated with criminal activity, while intellectual property lawyers may use WHOIS queries to identify the owners of domain names that infringe on their client's trademarks or copyrights.

In recent years, there have been concerns about the accuracy and privacy implications of WHOIS queries, which has led to the development of alternative approaches to WHOIS, such as RDAP (Registration Data Access Protocol) and GDPR-compliant WHOIS. Nonetheless, WHOIS queries remain an important tool for many individuals and organizations seeking to obtain information about domain names and IP addresses.

Unlocking the Power of Google Dorks



Introduction

Google Dorks, also known as Google Hacking or Google Operators, are advanced search techniques that enable users to fine-tune their queries and retrieve more precise results. By leveraging Google Dorks, users can uncover specific information, discover vulnerabilities, or even access sensitive data inadvertently exposed on the web.

Understanding Google Dorks

At its core, Google Dorks take advantage of advanced search operators to refine search queries, filtering out irrelevant results and focusing on the information users genuinely seek. Although the term "Google Hacking" may imply malicious intent, Google Dorks are not inherently harmful. They can be used for legitimate purposes, such as market research, competitive analysis, or enhancing the efficiency of online investigations. However, they can also be employed by cybercriminals to identify potential targets or find sensitive information that should not be publicly available.

Basic Google Search Operators

Google search operators are special characters or commands that modify a search query to achieve more specific results. Some basic operators include:

Quotation Marks (""): Using quotation marks around a phrase searches for the exact match, excluding results with only partial matches.

Example: **"apple pie"**

Minus Sign (-): The minus sign excludes results containing the specified term.

Example: **apple -pie**

Site (site:): The site operator restricts the search to a specific website or domain.

Example: **site:example.com**

Filetype (filetype:): This operator searches for specific file types, such as PDF, DOC, or XLS.

Example: **filetype:pdf**

Inurl (inurl:): The inurl operator searches for results with a specific word or phrase in the URL.

Example: **inurl:login**

Intitle (intitle:): This operator searches for results with a specific word or phrase in the page title.

Example: **intitle:"index of"**

Combining Google Search Operators

The true power of Google Dorks comes from combining multiple search operators to craft more precise queries. Some examples include:

Find PDF files on a specific website:

site:example.com filetype:pdf

Search for login pages within a specific domain:

site:example.com inurl:login

Locate documents with sensitive information:

intitle:"confidential" filetype:doc site:example.com

Identify potentially exposed directories:

intitle:"index of" inurl:wp-content/uploads

Ethical Considerations and Best Practices

While Google Dorks can be a powerful tool, it's essential to use them ethically and responsibly. Users should avoid searching for sensitive information that they have no legitimate reason to access. Additionally, organizations should be aware of the potential risks associated with Google Dorks and take necessary precautions to safeguard their data, such as:

Restricting Access: Limit access to sensitive information and use proper access controls, such as password protection or IP whitelisting.

Robots.txt: Use the robots.txt file to instruct search engines not to index specific directories or files.

Monitoring: Regularly monitor search engine indexes and logs to identify and remediate potential exposures.

Awareness: Educate employees on the risks associated with Google Dorks and the importance of properly securing sensitive information.

Google Dorks offer a powerful means to refine search queries and access targeted information with greater precision. By understanding the various operators and their combinations, users can unlock the full potential of Google Dorks for legitimate purposes while remaining mindful of the ethical considerations and potential risks associated with their use.

Understanding Hex, Base64, and Hashing

Hexadecimal

Hexadecimal, or simply hex, is a number system that represents numbers using a base-16 notation. Unlike the familiar decimal system (base-10), which uses digits 0 through 9, the hexadecimal system uses sixteen distinct symbols: 0-9 to represent the values 0 to 9, and A-F to represent values 10 to 15. Hexadecimal notation is widely used in computing and programming due to its concise representation of binary data and ease of conversion between binary and hexadecimal.

Hexadecimal Basics

A single hexadecimal digit can represent four binary digits, or bits, making it a compact way to display binary values. For example, the binary value 1101 can be represented as D in hexadecimal. To convert between hexadecimal and decimal or binary, follow these steps:

1. **Write the Hexadecimal Number:** for example – 2A3.
2. **Reverse the Hexadecimal Number:**
Starting from the right, label each position: 0, 1, 2, and so on. So, for 2A3, it would be:

Hexadecimal	3	A	2
Position	0	1	2

3. **Convert & Calculate:**
For each position, convert the symbol to its decimal value and multiply by 16^{position} . If the symbol is A-F, use its decimal value equivalent (A=10, B=11, C=12, D=13, E=14, F=15).

Hexadecimal	3	A	2
Position	0	1	2
Calculation	3×16^0	10×16^1	2×16^2
Result	3	160	512

4. **Add the Results:**
 $3 + 160 + 512 = 675$

So, 2A3 in hexadecimal is 675 in decimal.

To make it easier, you can also use online tools such as: <https://www.rapidtables.com/convert/number/hex-to-decimal.html>

The screenshot shows the RapidTables website interface for converting hexadecimal to decimal. The 'From' dropdown is set to 'Hexadecimal' and the 'To' dropdown is set to 'Decimal'. The input field 'Enter hex number' contains '2A3'. Below the input field are buttons for '= Convert', 'x Reset', and '↕ Swap'. The output field 'Decimal number (3 digits)' displays '675'.

Hexadecimal Encoding and Decoding Examples

Hexadecimal encoding is often used to represent binary data, such as RGB color codes, memory addresses, or checksums. Examples include:

- **RGB Color Codes:**
#FF0000 represents red, where FF is red, 00 is green, and 00 is blue.
- **Memory Addresses:**
0x7FFDE000 is a hexadecimal representation of a memory address.
- **Checksums:**
2A3B5C is an example of a hexadecimal checksum value.

Base64 Encoding and Decoding

Base64 is a binary-to-text encoding scheme that represents binary data in an ASCII format. It is commonly used for transmitting binary data over protocols that only support ASCII characters, such as email and HTTP. Base64 converts binary data into a series of 64 printable characters, which can be transmitted over any text-based protocol.

To understand Base64, imagine you want to send a photo to a friend via email. The photo is a binary file, which means it is made up of 1s and 0s. However, email can only transmit text-based data, such as letters and symbols. This is where Base64 comes in. You can convert the binary photo into a Base64-encoded string of text, which can then be sent over email.

Here's a simple example to illustrate how Base64 works.

1. Let's say we want to encode the binary sequence 11010111 01101000 00110001 00110011 into Base64. To do this, we first split the binary sequence into groups of six bits, like this:

110101 110110 001100 110011

2. Then, we convert each group of six bits into its corresponding Base64 symbol. The Base64 symbols are:

ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/-

3. So, the first group of six bits (110101) corresponds to the Base64 symbol "1", the second group (110110) corresponds to the Base64 symbol "m", and so on. After converting all four groups of six bits, we get the Base64-encoded string "1mww".
4. To decode a Base64-encoded string back into its original binary format, the process is reversed. Each Base64 symbol is converted back into its corresponding six bits, and the resulting binary sequence is combined to recreate the original binary data.

Decoding a Base64-encoded string:

To decode the example above, a Base64 decoding tool or programming library would be used to convert the string back into its original binary form.

Cryptographic Hash Functions: Exploring MD4, MD5, and SHA Algorithms

Cryptographic hash functions are mathematical algorithms that take an input and produce a fixed-size output, typically called a hash or digest. Hash functions are designed to be deterministic, meaning the same input will always produce the same output. They are also designed to be one-way functions, meaning it should be computationally infeasible to reverse-engineer the input from the output. Common hash functions include MD4, MD5, and the SHA family of algorithms.

Hash functions are mathematical algorithms that take input data of any size and convert it into fixed-size output, commonly known as a hash value, digest, or fingerprint. The logic behind hash functions is to create a unique identifier for any given input data that can be used for various purposes, such as data verification, indexing, or secure storage.

The basic principle behind hash functions is to map an input data set of arbitrary length to a fixed-size output, which is typically a sequence of bits or bytes. The output is calculated based on a set of rules defined by the hash function, which may include combining and manipulating the input data in various ways, such as addition, multiplication, shifting, or bitwise operations.

Hash functions have several key properties that make them useful for a wide range of applications, such as:

- **Determinism:** Hash functions produce the same output for a given input every time, regardless of when or where it is computed. This property enables hash functions to be used for data verification, where the hash value of a file or message can be compared against the expected hash value to ensure that the data has not been tampered with.
- **Uniqueness:** Hash functions aim to produce a unique hash value for each input data set. Although it is theoretically possible for two different input data sets to produce the same hash value, the probability of this occurring is very low for most widely-used hash functions. This property makes hash functions useful for indexing and searching, where a hash value can be used to quickly locate or compare data sets.
- **Non-reversibility:** Hash functions are designed to be one-way functions, meaning that it is computationally infeasible to generate the original input data from the hash value alone. This property makes hash functions useful for secure storage and password encryption, where the hash value of a password or other sensitive data can be stored instead of the original data.
- **Sensitivity to input changes:** Hash functions aim to produce different hash values for even slightly different input data sets. This property ensures that even small changes to the input data will result in a completely different hash value, making it difficult to predict the output of the hash function.

Overall, hash functions are a powerful and versatile tool for data manipulation and security, and are widely used in many different applications, such as cryptography, data indexing, and digital signatures. By applying a set of mathematical rules to input data, hash functions can generate unique and reliable hash values that can be used for a variety of purposes, from data verification to secure storage.

MD4 and MD5

MD4 (Message Digest Algorithm 4) and MD5 (Message Digest Algorithm 5) are two hash functions created by Ronald Rivest in the early 1990s. MD4 is a 128-bit hash function, while MD5 produces a 128-bit output as well. Though they were widely used in the past, both MD4 and MD5 are now considered insecure due to their susceptibility to various types of attacks, including collision attacks and preimage attacks.

For example, MD5 hash of the string "hello":

```
5d41402abc4b2a76b9719d911017c592
```

SHA (Secure Hash Algorithm) Family

The SHA family of algorithms was developed by the National Security Agency (NSA) and includes multiple variations, such as SHA-0, SHA-1, SHA-2, and SHA-3. Among these, SHA-1 and SHA-2 are the most widely used. SHA-1 produces a 160-bit hash, while SHA-2 has several versions (SHA-224, SHA-256, SHA-384, SHA-512) with different output sizes. Due to security concerns, SHA-1 is no longer considered secure, and the use of SHA-2 or SHA-3 is recommended.

For example, SHA-256 hash of the string "hello":

```
2cf24dba5fb0a30e26e83b2ac5b9e29e1b161e5c1fa7425e73043362938b9824
```

Practical Uses of Hash Functions

Cryptographic hash functions have several practical applications, including:

- 1. File Integrity Verification**

By comparing the hash of a downloaded file to the original hash provided by the author, users can verify that the file has not been tampered with or corrupted.

- 2. Password Storage**

Storing hashes of user passwords instead of plain text helps protect sensitive information in case of a data breach.

- 3. Digital Signatures**

Hash functions are used in digital signatures to ensure the integrity and authenticity of messages or documents.

- 4. Blockchain Technology**

Cryptographic hash functions play a crucial role in securing blockchain networks, such as Bitcoin, by verifying and linking blocks of transactions.

The Differences

Encoding, hashing, and encryption are all techniques used to protect data or information, but they serve different purposes and offer varying levels of security. Here's a brief overview of each technique and their differences:

Encoding

Encoding is the process of converting data from one format to another format. It's often used to make data more suitable for transmission or storage. For example, when sending an email, special characters or non-ASCII characters may be encoded in order to ensure that the recipient can properly read the

email. Encoding does not provide any security, as the encoded data can be easily decoded by anyone who knows the encoding method used.

Hashing

Hashing is a technique used to ensure data integrity, meaning that the data has not been modified or tampered with. Hashing takes a piece of data and generates a fixed-length string of characters, called a hash value or message digest, which is unique to that data. If the data is modified, the hash value will change. Hashing is a one-way process, meaning that it's impossible to generate the original data from the hash value. Hashing is often used to store passwords securely, as the password is hashed and stored instead of the plaintext password itself.

Encryption

Encryption is the process of converting plaintext (unencrypted) data into ciphertext (encrypted) data, so that it can only be read by someone who has the decryption key. Encryption is often used to protect sensitive information, such as credit card numbers, personal information, or confidential business data. Encryption can be either symmetric, where the same key is used to encrypt and decrypt the data, or asymmetric, where two different keys are used. Asymmetric encryption is typically more secure than symmetric encryption, but it's also slower and more resource-intensive.

An Introduction to Windows Command-Line (CMD)

The Windows Command Prompt, also known as cmd.exe or simply CMD, is a command-line interface (CLI) included with Microsoft Windows. The Command Prompt allows users to execute commands, navigate the file system, and manage system settings without the need for a graphical user interface (GUI).

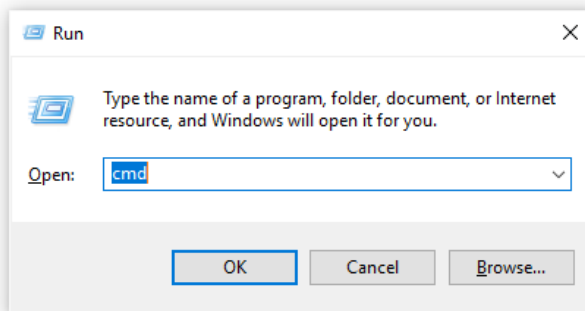
Understanding of Windows Command Prompt

The Windows Command Prompt provides a text-based interface for interacting with the operating system, running commands, and managing files and directories. While modern Windows versions include graphical interfaces, the Command Prompt remains a powerful tool for system administration, troubleshooting, and automation tasks.

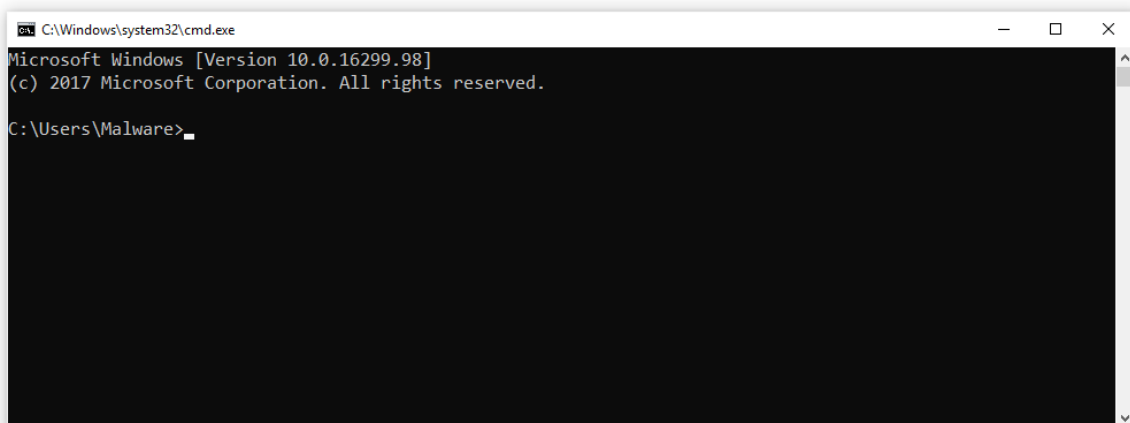
Basic Usage of Windows Command Prompt

To open the Windows Command Prompt, follow these steps:

1. Press Winkey + R.
2. Type "cmd" in the search bar.



3. Click on the "Command Prompt" result or press Enter.



Once the Command Prompt is open, you can begin executing commands. Some basic commands include:

- dir** Lists the contents of the current directory.
- cd** Changes the current directory. For example, cd Documents will change the current directory to the Documents folder.

mkdir	Creates a new directory. For example, mkdir NewFolder will create a new folder named "NewFolder" in the current directory.
rmdir	Deletes an empty directory. For example, rmdir NewFolder will delete the "NewFolder" directory.
del	Deletes a file. For example, del file.txt will delete the "file.txt" file.
copy	Copies a file from one location to another. For example, copy source.txt destination.txt will copy "source.txt" to "destination.txt."
move	Moves a file from one location to another. For example, move source.txt destination.txt will move "source.txt" to "destination.txt."
ren	Renames a file or directory. For example, ren oldname.txt newname.txt will rename "oldname.txt" to "newname.txt."

[Internet Commands in Windows Command Prompt](#)

The Windows Command Prompt includes several built-in commands for network diagnostics and internet-related tasks. Some of these commands include:

ping	Tests the network connectivity between the local computer and a remote host or IP address. For example, ping www.example.com or ping 8.8.8.8.
tracert	Traces the route that packets take to reach a remote host or IP address. This command can help identify network issues or slow connections. For example, tracert www.example.com or tracert 8.8.8.8.
nslookup	Queries the Domain Name System (DNS) to obtain domain name or IP address information. For example, nslookup www.example.com will display the IP address for "www.example.com."
netstat	Displays active network connections, listening ports, and network statistics. For example, netstat -a will display all active connections and listening ports.
ipconfig	Displays the current network configuration, including IP addresses, subnet masks, and default gateway. For example, ipconfig /all will display detailed network configuration information.

To view detailed information about all network adapters on your computer, including IP addresses, DNS servers, and more, type **ipconfig /all** and press Enter. This will display a long list of information about each adapter.

To release the IP address assigned to a network adapter, type **ipconfig /release** followed by the adapter name in quotes. For example, to release the IP address assigned to the Ethernet adapter, you would type ipconfig /release "Ethernet" and press Enter.

To renew the IP address assigned to a network adapter, type `ipconfig /renew` followed by the adapter name in quotes. For example, to renew the IP address assigned to the Ethernet adapter, you would type `ipconfig /renew "Ethernet"` and press Enter.

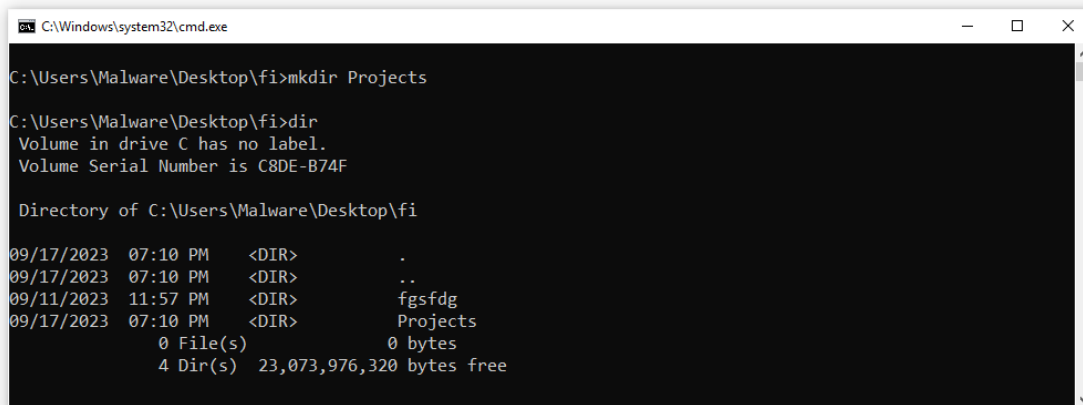
To flush the DNS resolver cache, which can sometimes help resolve network connectivity issues, type `ipconfig /flushdns` and press Enter.

In summary, the `ipconfig /all` command is a powerful tool for viewing detailed information about network adapters on your computer. It can be used to troubleshoot network connectivity issues, release and renew IP addresses, and flush the DNS resolver cache.

Here are some simple examples of using the Windows Command Prompt for everyday tasks:

Example 1: Creating and navigating directories

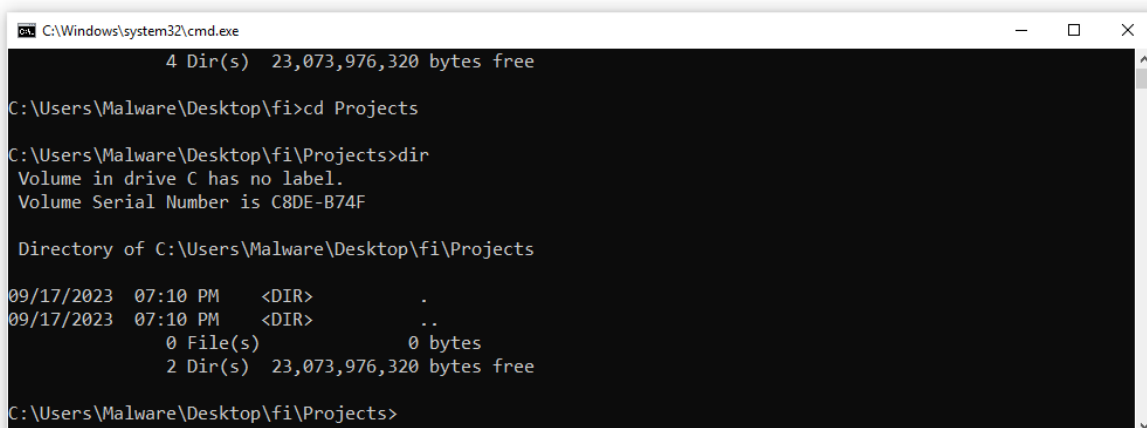
Create a new directory called "Projects" with the following command: `mkdir Projects`



```
C:\Windows\system32\cmd.exe
C:\Users\Malware\Desktop\fi>mkdir Projects
C:\Users\Malware\Desktop\fi>dir
Volume in drive C has no label.
Volume Serial Number is C8DE-B74F

Directory of C:\Users\Malware\Desktop\fi
09/17/2023  07:10 PM    <DIR>          .
09/17/2023  07:10 PM    <DIR>          ..
09/11/2023  11:57 PM    <DIR>          fgsfdg
09/17/2023  07:10 PM    <DIR>          Projects
               0 File(s)                0 bytes
               4 Dir(s)  23,073,976,320 bytes free
```

Change the current directory to the newly created "Projects" directory: `cd Projects`

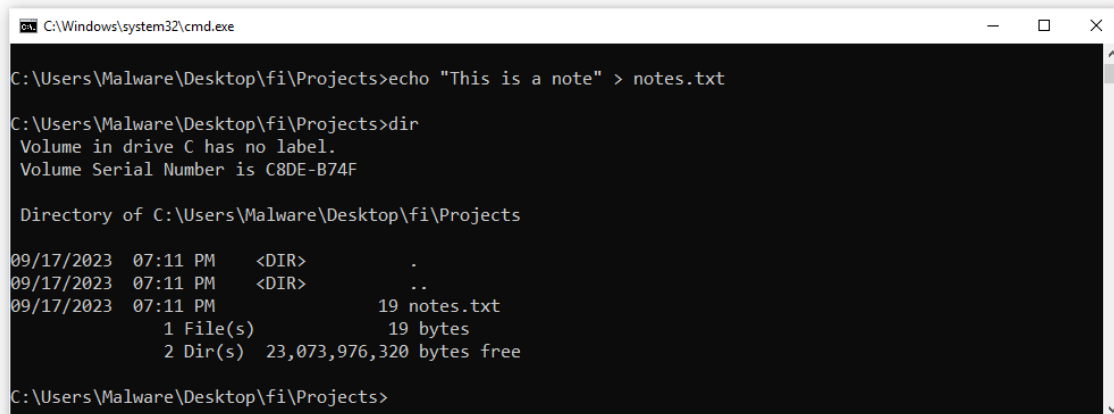


```
C:\Windows\system32\cmd.exe
4 Dir(s) 23,073,976,320 bytes free
C:\Users\Malware\Desktop\fi>cd Projects
C:\Users\Malware\Desktop\fi\Projects>dir
Volume in drive C has no label.
Volume Serial Number is C8DE-B74F

Directory of C:\Users\Malware\Desktop\fi\Projects
09/17/2023  07:10 PM    <DIR>          .
09/17/2023  07:10 PM    <DIR>          ..
               0 File(s)                0 bytes
               2 Dir(s)  23,073,976,320 bytes free
C:\Users\Malware\Desktop\fi\Projects>
```

Example 2: Creating a text file and viewing its contents

Create a new text file called "notes.txt" with the following command: **echo "This is a note" > notes.txt**

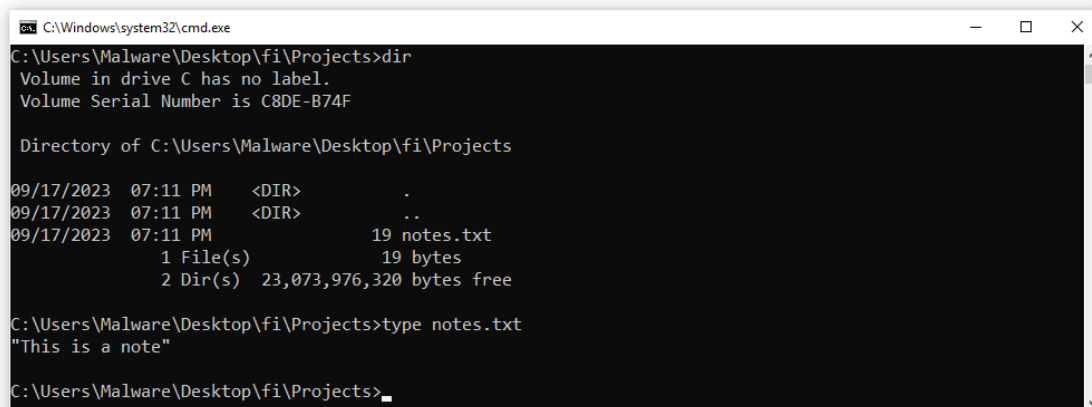


```
C:\Windows\system32\cmd.exe
C:\Users\Malware\Desktop\fi\Projects>echo "This is a note" > notes.txt
C:\Users\Malware\Desktop\fi\Projects>dir
Volume in drive C has no label.
Volume Serial Number is C8DE-B74F

Directory of C:\Users\Malware\Desktop\fi\Projects
09/17/2023  07:11 PM  <DIR>          .
09/17/2023  07:11 PM  <DIR>          ..
09/17/2023  07:11 PM                19 notes.txt
               1 File(s)                19 bytes
               2 Dir(s)  23,073,976,320 bytes free

C:\Users\Malware\Desktop\fi\Projects>
```

View the contents of "notes.txt" with the following command: **type notes.txt**



```
C:\Windows\system32\cmd.exe
C:\Users\Malware\Desktop\fi\Projects>dir
Volume in drive C has no label.
Volume Serial Number is C8DE-B74F

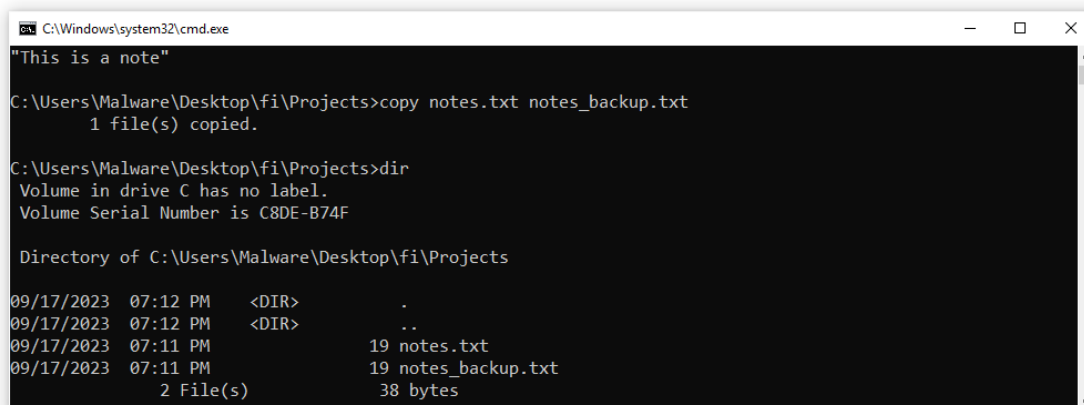
Directory of C:\Users\Malware\Desktop\fi\Projects
09/17/2023  07:11 PM  <DIR>          .
09/17/2023  07:11 PM  <DIR>          ..
09/17/2023  07:11 PM                19 notes.txt
               1 File(s)                19 bytes
               2 Dir(s)  23,073,976,320 bytes free

C:\Users\Malware\Desktop\fi\Projects>type notes.txt
"This is a note"

C:\Users\Malware\Desktop\fi\Projects>
```

Example 3: Copying, renaming, and deleting files

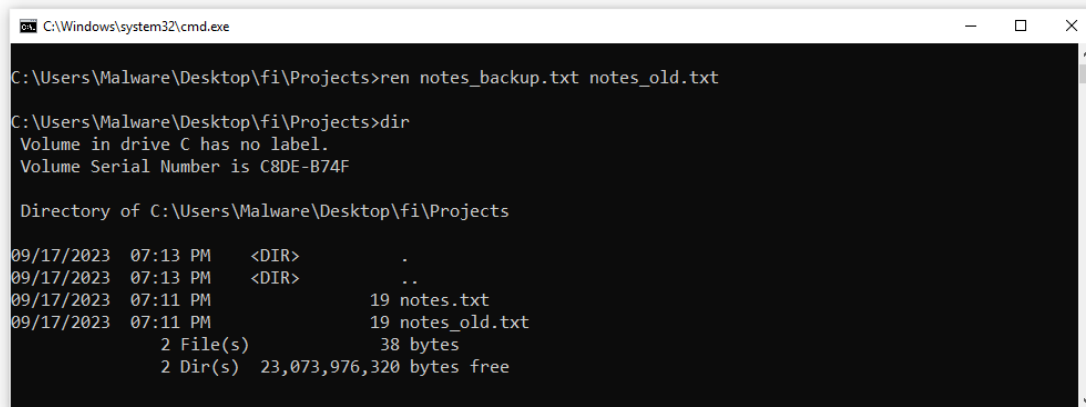
Create a copy of "notes.txt" called "notes_backup.txt" with the following command: **copy notes.txt notes_backup.txt**



```
C:\Windows\system32\cmd.exe
"This is a note"
C:\Users\Malware\Desktop\fi\Projects>copy notes.txt notes_backup.txt
1 file(s) copied.
C:\Users\Malware\Desktop\fi\Projects>dir
Volume in drive C has no label.
Volume Serial Number is C8DE-B74F

Directory of C:\Users\Malware\Desktop\fi\Projects
09/17/2023  07:12 PM  <DIR>          .
09/17/2023  07:12 PM  <DIR>          ..
09/17/2023  07:11 PM                19 notes.txt
09/17/2023  07:11 PM                19 notes_backup.txt
               2 File(s)                38 bytes
```

Rename "notes_backup.txt" to "notes_old.txt" with the following command: ***ren notes_backup.txt notes_old.txt***



```
C:\Windows\system32\cmd.exe

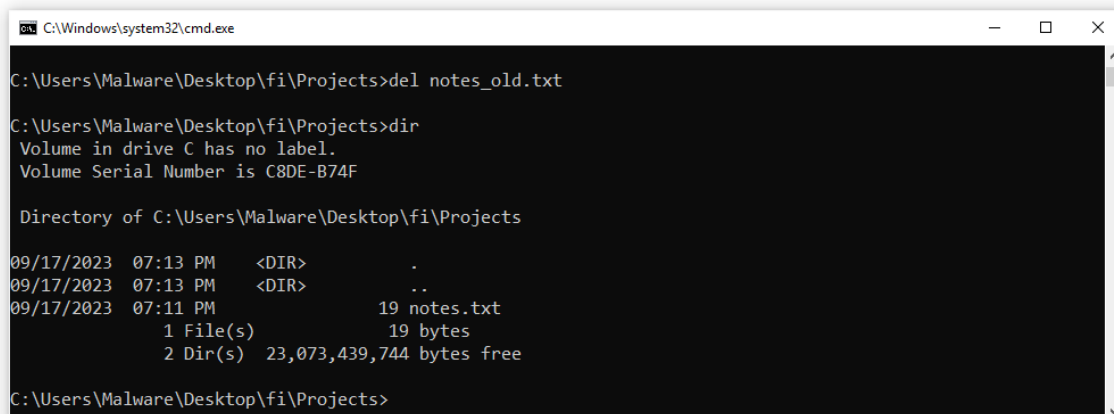
C:\Users\Malware\Desktop\fi\Projects>ren notes_backup.txt notes_old.txt

C:\Users\Malware\Desktop\fi\Projects>dir
Volume in drive C has no label.
Volume Serial Number is C8DE-B74F

Directory of C:\Users\Malware\Desktop\fi\Projects

09/17/2023  07:13 PM  <DIR>          .
09/17/2023  07:13 PM  <DIR>          ..
09/17/2023  07:11 PM                19 notes.txt
09/17/2023  07:11 PM                19 notes_old.txt
           2 File(s)                38 bytes
           2 Dir(s) 23,073,976,320 bytes free
```

Delete "notes_old.txt" with the following command: ***del notes_old.txt***



```
C:\Windows\system32\cmd.exe

C:\Users\Malware\Desktop\fi\Projects>del notes_old.txt

C:\Users\Malware\Desktop\fi\Projects>dir
Volume in drive C has no label.
Volume Serial Number is C8DE-B74F

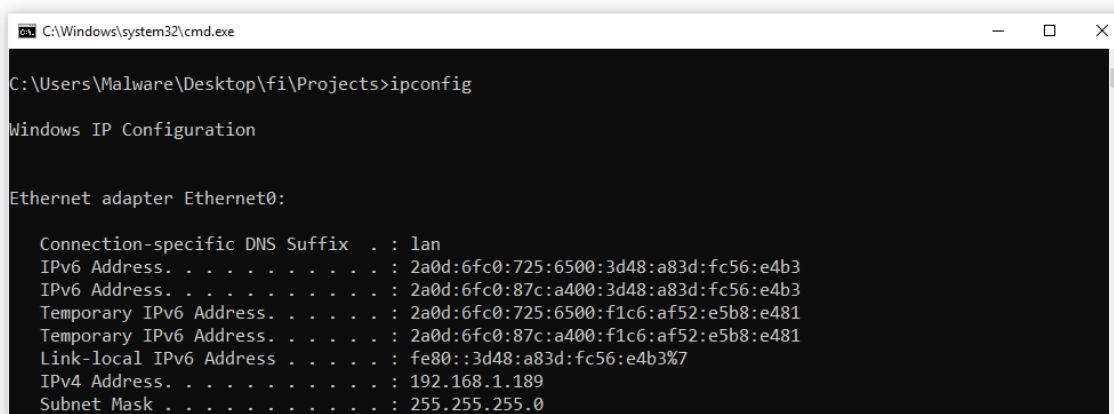
Directory of C:\Users\Malware\Desktop\fi\Projects

09/17/2023  07:13 PM  <DIR>          .
09/17/2023  07:13 PM  <DIR>          ..
09/17/2023  07:11 PM                19 notes.txt
           1 File(s)                19 bytes
           2 Dir(s) 23,073,439,744 bytes free

C:\Users\Malware\Desktop\fi\Projects>
```

Example 4: Displaying IP configuration and network status

Display the current IP configuration with the following command: ***ipconfig***



```
C:\Windows\system32\cmd.exe

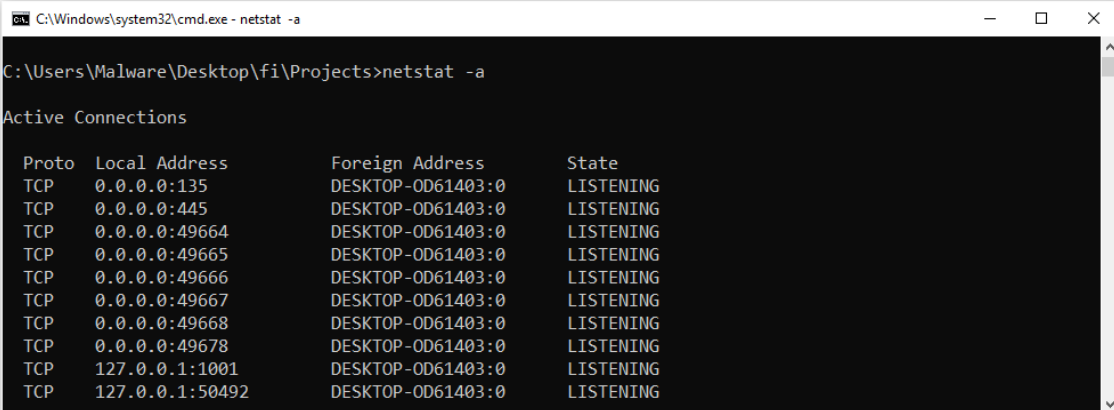
C:\Users\Malware\Desktop\fi\Projects>ipconfig

Windows IP Configuration

Ethernet adapter Ethernet0:

   Connection-specific DNS Suffix  . : lan
   IPv6 Address. . . . . : 2a0d:6fc0:725:6500:3d48:a83d:fc56:e4b3
   IPv6 Address. . . . . : 2a0d:6fc0:87c:a400:3d48:a83d:fc56:e4b3
   Temporary IPv6 Address. . . . . : 2a0d:6fc0:725:6500:f1c6:af52:e5b8:e481
   Temporary IPv6 Address. . . . . : 2a0d:6fc0:87c:a400:f1c6:af52:e5b8:e481
   Link-local IPv6 Address . . . . . : fe80::3d48:a83d:fc56:e4b3%7
   IPv4 Address. . . . . : 192.168.1.189
   Subnet Mask . . . . . : 255.255.255.0
```

Display active network connections and listening ports with the following command: **netstat -a**



```
C:\Windows\system32\cmd.exe - netstat -a
C:\Users\Malware\Desktop\fi\Projects>netstat -a
Active Connections

Proto Local Address           Foreign Address         State
TCP   0.0.0.0:135             DESKTOP-0D61403:0      LISTENING
TCP   0.0.0.0:445             DESKTOP-0D61403:0      LISTENING
TCP   0.0.0.0:49664           DESKTOP-0D61403:0      LISTENING
TCP   0.0.0.0:49665           DESKTOP-0D61403:0      LISTENING
TCP   0.0.0.0:49666           DESKTOP-0D61403:0      LISTENING
TCP   0.0.0.0:49667           DESKTOP-0D61403:0      LISTENING
TCP   0.0.0.0:49668           DESKTOP-0D61403:0      LISTENING
TCP   0.0.0.0:49678           DESKTOP-0D61403:0      LISTENING
TCP   127.0.0.1:1001         DESKTOP-0D61403:0      LISTENING
TCP   127.0.0.1:50492        DESKTOP-0D61403:0      LISTENING
```

These simple examples demonstrate the ease of performing everyday tasks using the Windows Command Prompt. By practicing these commands, users can become more proficient in using the command-line interface to manage files, directories, and network configurations.

Common File Extensions

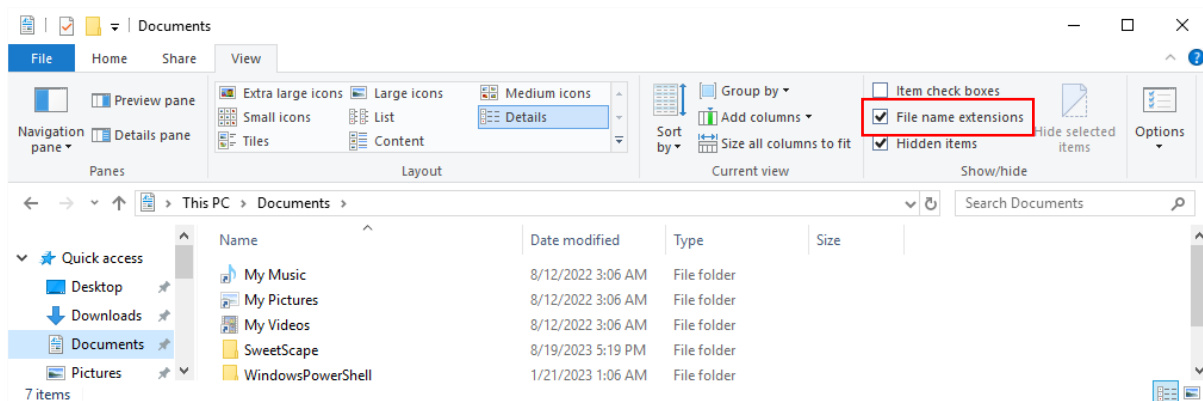
Working with file extensions is relatively simple. In most cases, you don't need to worry about changing the file extension, as the file will work with the software program it was designed for. In Windows operating systems, common file extensions (like .txt for text files, .jpg for JPEG images, .exe for executables, etc.) are hidden by default for several reasons:

1. **Simplicity for Users:** For many everyday users, seeing file extensions can be confusing, especially if they're unfamiliar with what these extensions mean. By hiding them, Microsoft aimed to present a simpler and cleaner look to the user interface.
2. **Reduced Accidental Changes:** If file extensions are shown and users accidentally change or delete them while renaming a file, the file may become unusable or unrecognizable by the system. By hiding the extensions, it reduces the chance of accidental modification.
3. **Aesthetic Reasons:** Displaying file names without their extensions provides a more streamlined look in the file explorer.
4. **File Icons:** Windows uses unique icons for different file types, which provides a visual cue to users about the file's type or the program associated with it. For example, a Word document has its own distinct icon different from an Excel spreadsheet.
5. **Security:** Some malicious entities use file extensions as a way to deceive users. For instance, a file named "picture.jpg.exe" might look like a harmless picture file at first glance, but it's an executable that could contain malware. If extensions are shown, inexperienced users might just see the ".jpg" and assume it's safe. Hiding known extensions can potentially mitigate this threat, though it's worth noting that it's not a foolproof security measure.

common file extensions and what they indicate:

- .doc/.docx** - Microsoft Word document
- .pdf** - Adobe PDF document
- .jpg/.jpeg** - JPEG image file
- .png** - Portable Network Graphics image file
- .mp3** - MP3 audio file
- .mp4** - MP4 video file
- .xlsx** - Microsoft Excel spreadsheet

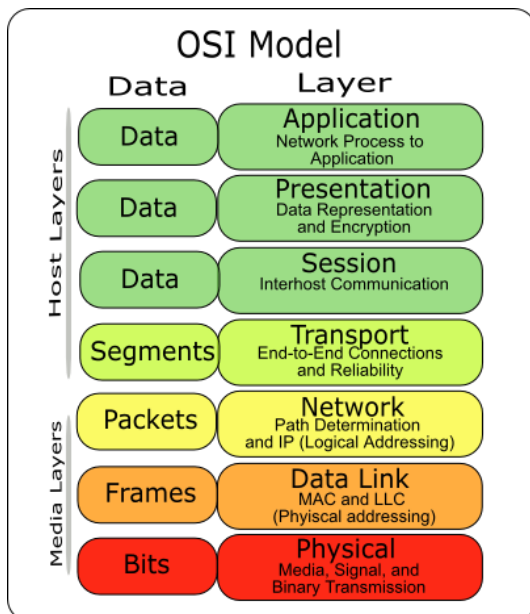
However, advanced users and professionals often prefer to see the extensions to manage their files better. If you wish to see file extensions in Windows, you can easily change this in the Folder Options or View settings of File Explorer.



Understanding the OSI Model

The OSI (Open Systems Interconnection) model is a conceptual framework that describes the communication process between two devices over a network. It was developed in the 1980s by the International Organization for Standardization (ISO) as a way to standardize network communication protocols and improve interoperability between different vendors' network equipment.

The OSI model is organized into seven layers, each of which describes a specific aspect of the communication process. Each layer builds upon the layer below it, creating a stack of layers that work together to transmit data between devices.



Layer 1: Physical Layer

The Physical Layer is the lowest layer of the OSI model, responsible for the physical connection between devices. This layer deals with the transmission of raw data in the form of electrical, optical, or radio signals. It defines the types of cables, connectors, and network interface cards (NICs) used to establish these connections. In simple terms, the Physical Layer ensures that data is transmitted from one device to another over a physical medium, like an Ethernet cable or Wi-Fi signal.

Layer 2: Data Link Layer

The Data Link Layer establishes a reliable link between two directly connected devices. This layer ensures the correct transmission of data packets by organizing them into frames and adding error detection and correction information. The Data Link Layer is also responsible for Media Access Control (MAC) addressing, which uniquely identifies devices on a local network. In summary, the Data Link Layer helps maintain a stable connection between devices, manages data flow, and enables devices to be uniquely identified within a local network.

Layer 3: Network Layer

The Network Layer is responsible for routing data packets between different networks. This layer uses logical addresses, such as IP addresses, to identify devices and determine the best path to transmit data between them. The Network Layer also manages traffic congestion and deals with packet fragmentation when necessary. In essence, the Network Layer ensures that data reaches its destination even when it needs to travel across multiple networks.

Layer 4: Transport Layer

The Transport Layer provides end-to-end communication services between devices. This layer is responsible for ensuring that data is transmitted reliably, in the correct sequence, and without errors. The Transport Layer uses protocols like Transmission Control Protocol (TCP) for reliable, connection-oriented communication or User Datagram Protocol (UDP) for connectionless, faster communication. In simple terms, the Transport Layer ensures that data is delivered accurately and efficiently from one device to another over a network.

Layer 5: Session Layer

The Session Layer manages the establishment, maintenance, and termination of connections (sessions) between devices. This layer enables multiple applications on different devices to communicate simultaneously by creating, managing, and synchronizing sessions. In essence, the Session Layer helps devices maintain an organized and secure connection during communication.

Layer 6: Presentation Layer

The Presentation Layer is responsible for data formatting, translation, and encryption. This layer ensures that the data sent by one device is understandable by the receiving device by converting it into a common format. The Presentation Layer can also handle data compression and encryption to ensure data is transmitted efficiently and securely. In simple terms, the Presentation Layer translates data into a format that both the sender and receiver can understand and ensures the data's confidentiality and integrity.

Layer 7: Application Layer

The Application Layer is the topmost layer of the OSI model, and it interfaces directly with user applications. This layer provides the protocols and services necessary for communication between applications, such as web browsers, email clients, and file sharing programs. The Application Layer is responsible for identifying communication partners, establishing resource availability, and ensuring data is sent and received correctly. In essence, the Application Layer is the bridge between the network and the user, enabling seamless communication between applications on different devices.

In summary, the OSI model is a framework that describes the communication process between two devices over a network. Each layer of the model performs a specific function, building upon the layer below it to transmit data between devices. By understanding the OSI model, network engineers and administrators can troubleshoot network issues, design and implement network protocols, and ensure that network communication is reliable and efficient.

Inspecting Network Traffic using Wireshark

Wireshark is a free and open-source packet analyzer that allows you to see what's happening on your network at a microscopic level. With Wireshark, you can capture network traffic in real-time and analyze it to troubleshoot network issues, detect security vulnerabilities, and optimize network performance.

What is Wireshark?

Wireshark is a network protocol analyzer that captures packets and displays their details in a user-friendly interface. It supports a wide range of protocols, including TCP, UDP, HTTP, DNS, and many more. Wireshark is available for Windows, macOS, and Linux, and it can capture traffic from both wired and wireless networks.

How does Wireshark work?

Wireshark captures packets by putting the network interface card (NIC) into promiscuous mode. In this mode, the NIC captures all traffic on the network, regardless of its intended recipient. Wireshark then decodes and analyzes the packets, displaying their details in a hierarchical tree-like structure. You can drill down into each packet to see its various fields, such as source and destination IP addresses, protocol type, packet length, and much more.

What can you do with Wireshark?

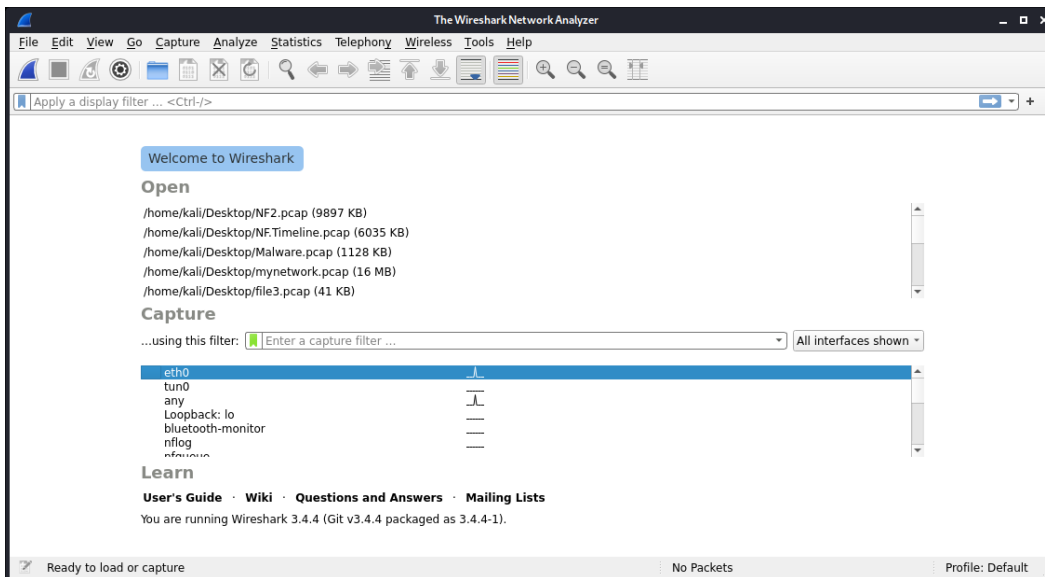
Wireshark can be used for a variety of network-related tasks, such as:

- **Troubleshooting network issues:** Wireshark can help you identify issues such as slow network performance, high packet loss, and network outages. By analyzing the captured packets, you can pinpoint the source of the problem and take corrective action.
- **Detecting security vulnerabilities:** Wireshark can help you detect security vulnerabilities such as data breaches, malware infections, and phishing attacks. By analyzing the packets, you can identify suspicious traffic and take steps to protect your network.
- **Optimizing network performance:** Wireshark can help you optimize network performance by analyzing network traffic patterns and identifying areas where improvements can be made. For example, you can use Wireshark to identify bottlenecks in the network and optimize network configurations for better performance.
- **Learning about network protocols:** Wireshark is a great tool for learning about network protocols and how they work. By analyzing packets captured on your network, you can see how different protocols are used and get a better understanding of how your network functions.

How to use Wireshark

Using Wireshark is relatively straightforward. Here are the basic steps:

Install Wireshark: You can download Wireshark from the official website and install it on your computer.



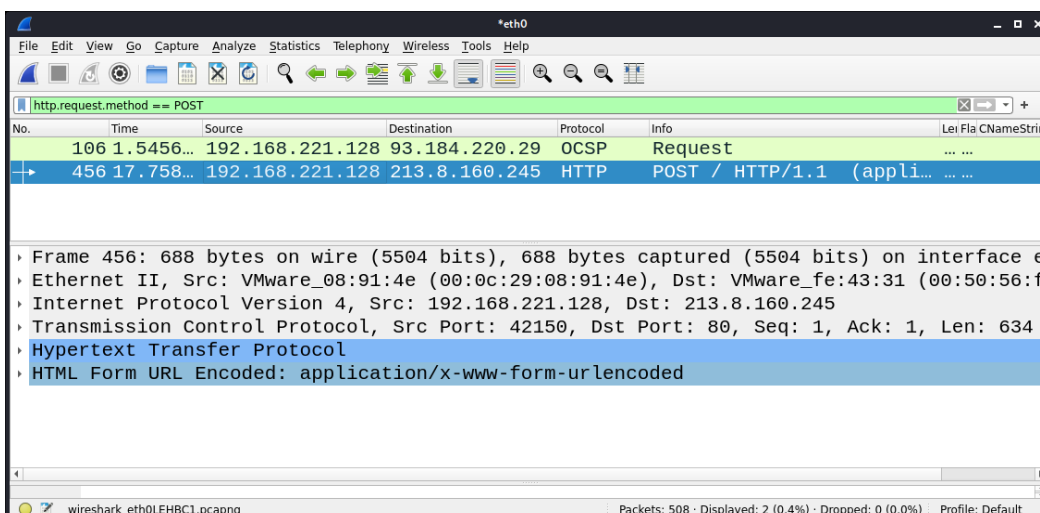
Capture packets: Start Wireshark and select the network interface you want to capture packets from. Then click the "Start" button to begin capturing packets.

Analyze packets: Once packets are captured, you can analyze them by selecting a packet in the packet list and examining its details in the packet details pane.

Filter packets: Wireshark allows you to filter packets based on various criteria such as IP address, protocol type, and packet length. Filtering can help you focus on specific packets and make analysis easier.

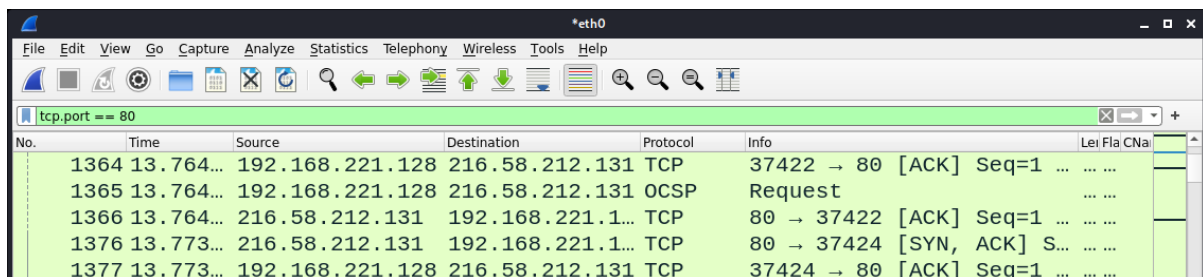
Packets contain the following:

- No Several packages from the start of the capture.
- Time Time passed from the initial capture.
- Source IP address, the source of the packet (sender).
- Destination Packet destination, IP address.
- Protocol The protocol is used to transfer the packet.
- Length Length of the packet, size in bytes.
- Info Details of the packet.



Basic Filters to Know

ip.addr	Filters packets based on source or destination IP address.
tcp.port	Filters packets based on the source or destination TCP port number.
udp.port	Filters packets based on the source or destination UDP port number.
http	Filters packets that contain HTTP traffic.
dns	Filters packets that contain DNS traffic.
arp	Filters packets that contain ARP traffic.
icmp	Filters packets that contain ICMP traffic.
ip.proto	Filters packets based on the protocol type, such as TCP, UDP, or ICMP.
frame.len	Filters packets based on the length of the packet.
eth.addr	Filters packets based on the source or destination MAC address.



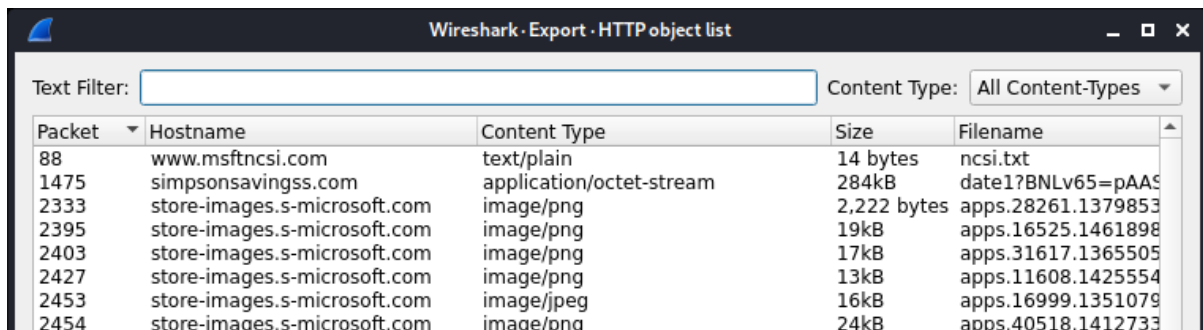
No.	Time	Source	Destination	Protocol	Info	Len	Fla	CNa
1364	13.764...	192.168.221.128	216.58.212.131	TCP	37422 → 80 [ACK] Seq=1 ...			
1365	13.764...	192.168.221.128	216.58.212.131	OCSP	Request			
1366	13.764...	216.58.212.131	192.168.221.1...	TCP	80 → 37422 [ACK] Seq=1 ...			
1376	13.773...	216.58.212.131	192.168.221.1...	TCP	80 → 37424 [SYN, ACK] S...			
1377	13.773...	192.168.221.128	216.58.212.131	TCP	37424 → 80 [ACK] Seq=1 ...			

Extracting Objects and Files from Pcap Files

We can extract all the files transferred in that part of captured network traffic with the export objects option.

File > Export Objects > HTTP

Then we get this screen - that allows us to save all the files to our computer for further analysis.



Packet	Hostname	Content Type	Size	Filename
88	www.msftncsi.com	text/plain	14 bytes	ncsi.txt
1475	simpsonsavingss.com	application/octet-stream	284kB	date1?BNLv65=pAAS
2333	store-images.s-microsoft.com	image/png	2,222 bytes	apps.28261.1379853
2395	store-images.s-microsoft.com	image/png	19kB	apps.16525.1461898
2403	store-images.s-microsoft.com	image/png	17kB	apps.31617.1365505
2427	store-images.s-microsoft.com	image/png	13kB	apps.11608.1425554
2453	store-images.s-microsoft.com	image/jpeg	16kB	apps.16999.1351079
2454	store-images.s-microsoft.com	image/png	24kB	apps.40518.1412733

These filters can be combined using logical operators such as "and" and "or" to create more complex filters. For example, you could use the filter "ip.addr == 192.168.1.1 and tcp.port == 80" to only show packets that have a source or destination IP address of 192.168.1.1 and a TCP port of 80 (indicating HTTP traffic).

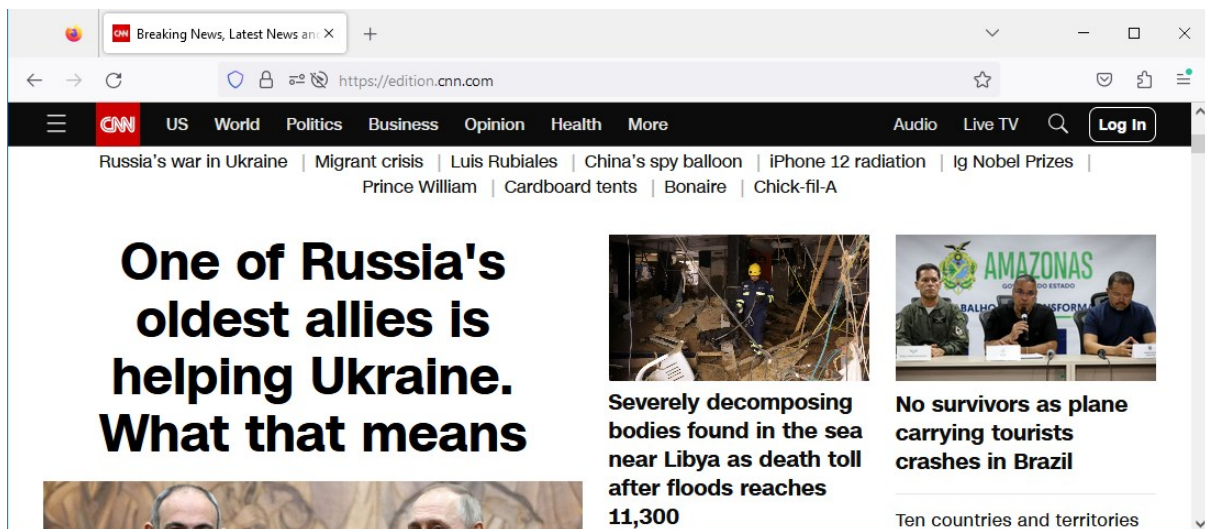
Remote Services

Introduction

The digital age has heralded a new era of connectivity, allowing professionals to remotely connect to servers or systems from the comfort of their homes or halfway across the globe. Three primary services enable this seamless connectivity: HTTP, FTP, RDP, and SSH.

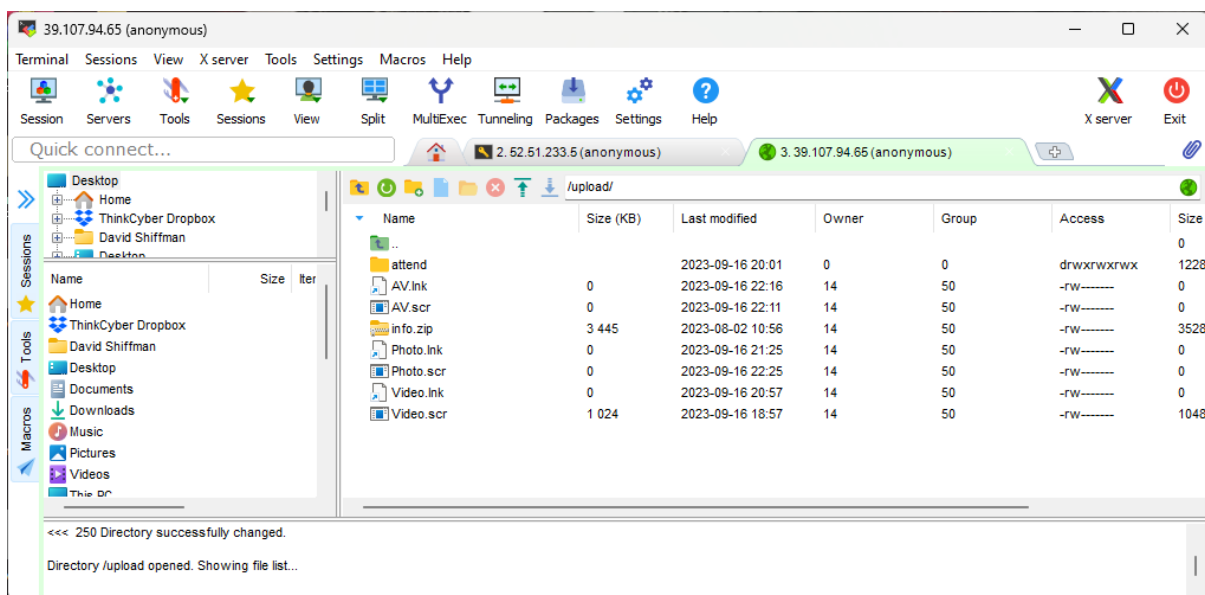
HTTP (HyperText Transfer Protocol)

HTTP is the foundation of any data exchange on the web, used primarily for transferring web pages on the internet. When you visit a website, you're essentially connecting to a remote service using HTTP.



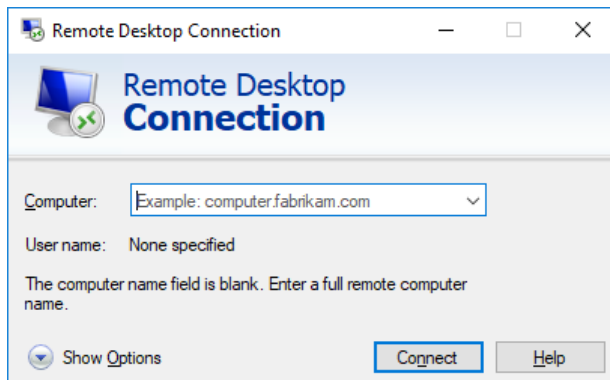
FTP (File Transfer Protocol)

FTP is a standard network protocol used to transfer files from one host to another over a TCP-based network, such as the internet. It's primarily used for uploading or downloading files to and from a server.



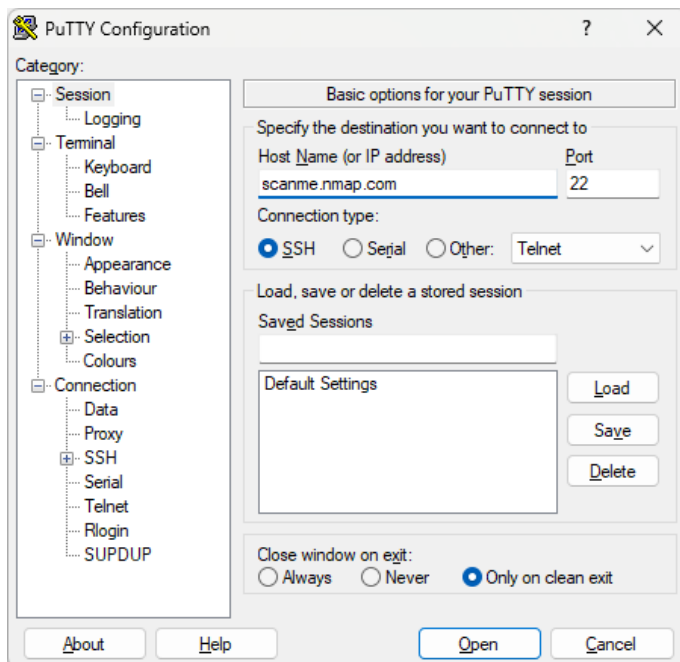
RDP (Remote Desktop Protocol)

RDP is a proprietary protocol developed by Microsoft that allows a user to connect to another computer over a network connection in a graphical interface. It's used predominantly for remote desktop sessions to Windows machines.



SSH (Secure Shell)

SSH is a cryptographic network protocol used for secure data communication, remote shell services, and other secure network services between two networked computers. It's widely used to securely access and manage systems.



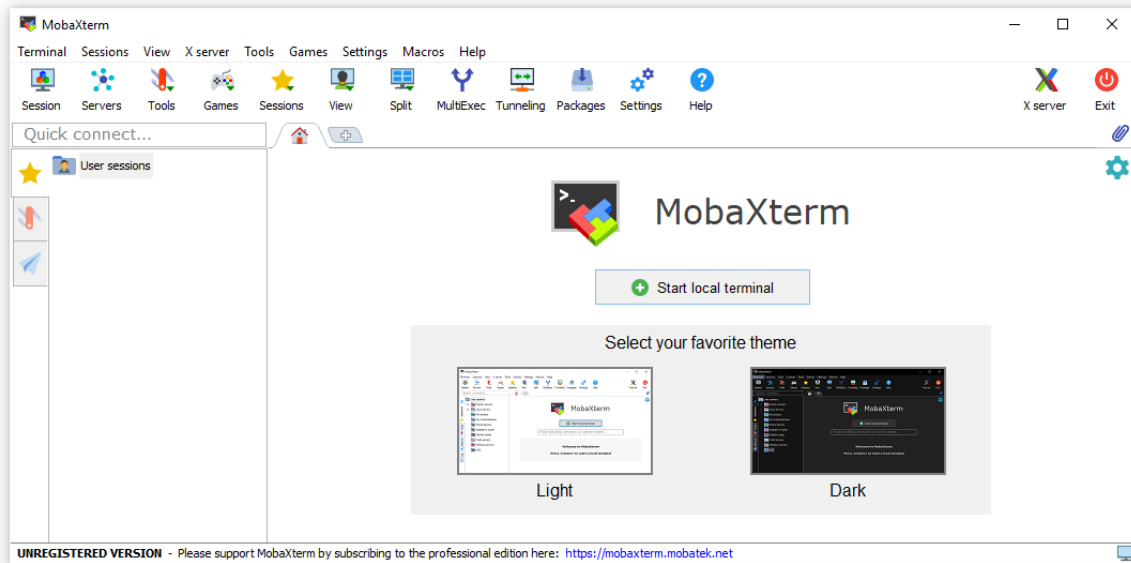
How to Connect to Remote Services

Connecting to FTP

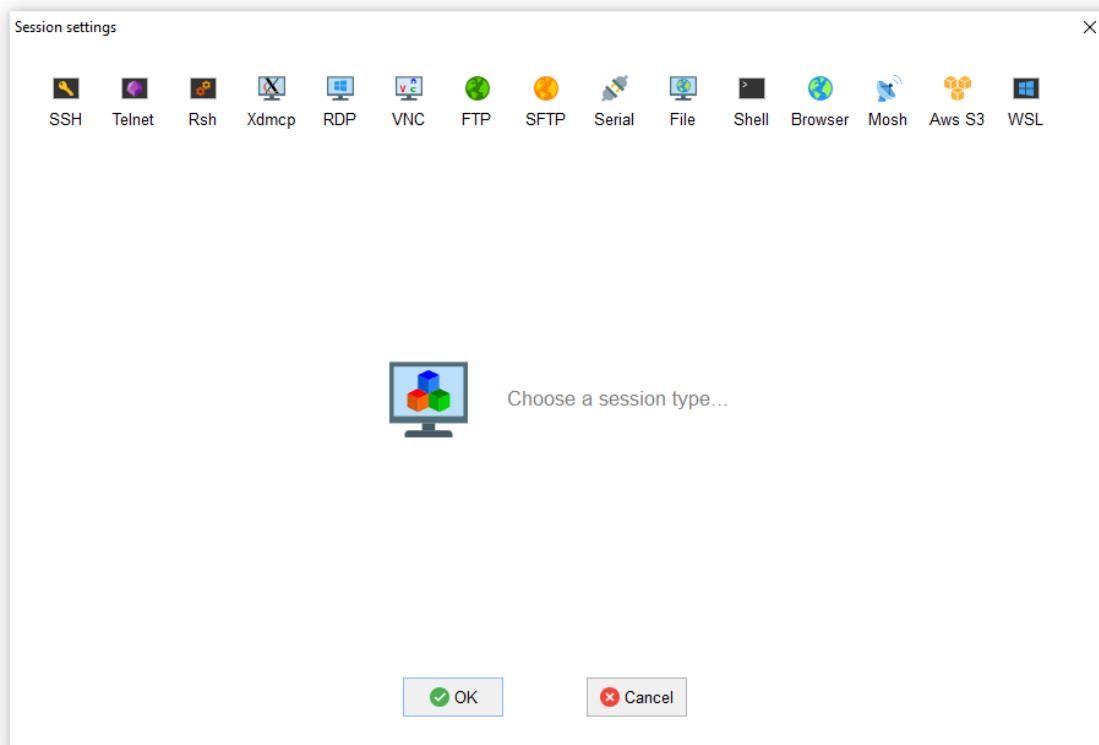
Via MobaXterm

MobaXterm is a versatile tool that offers both FTP and SFTP capabilities.

1. Open MobaXterm.

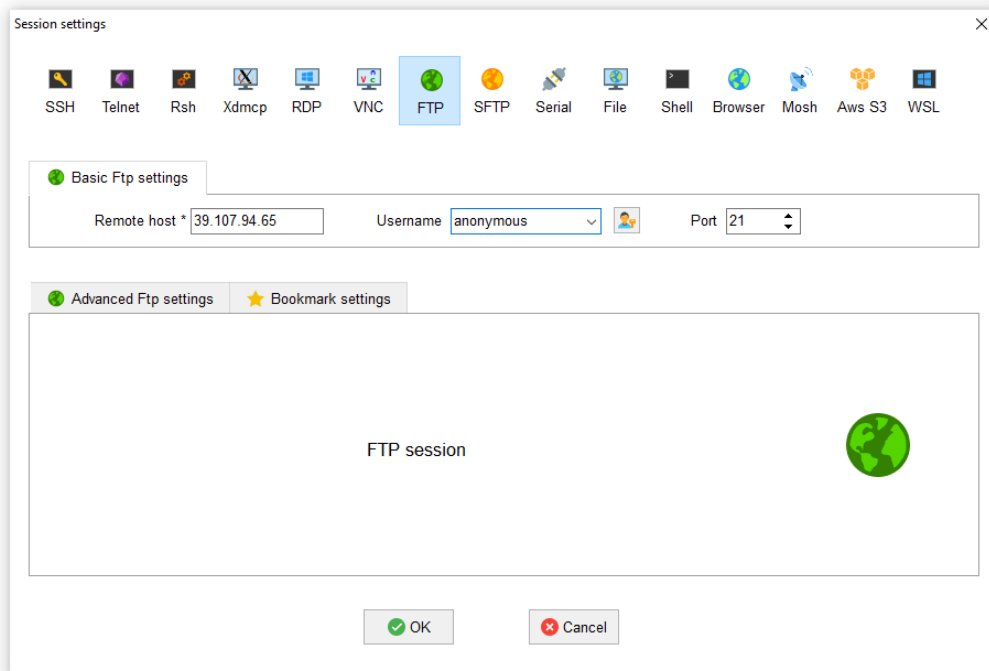


2. Click on the "Session" button.



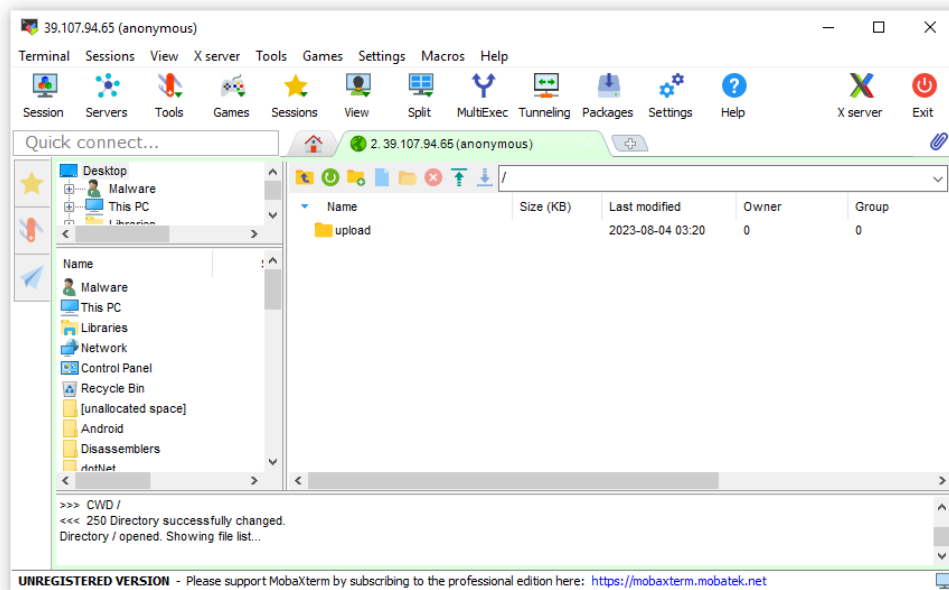
3. Choose "FTP" from the session type list and fill the details:

- Remote host
- Username
- Password
- Port (usually 21 for FTP)



4. Click "OK" to initiate the connection.

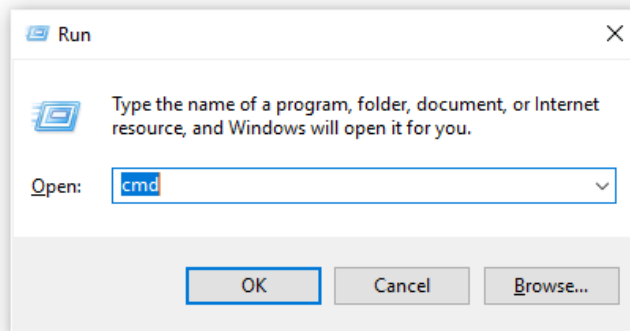
Once connected, you can use the graphical interface to navigate and manage files.



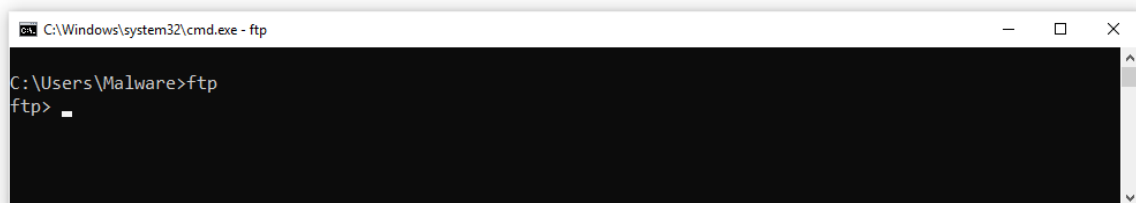
Via Command Line (Windows Command Prompt)

The Windows Command Prompt provides native support for FTP.

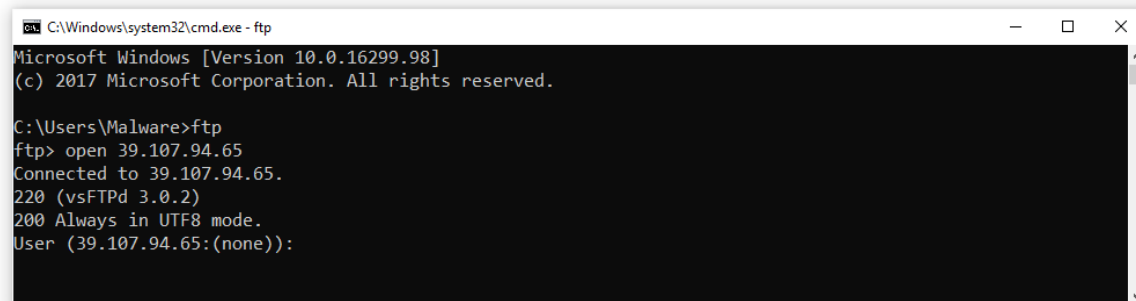
1. Open the Command Prompt.



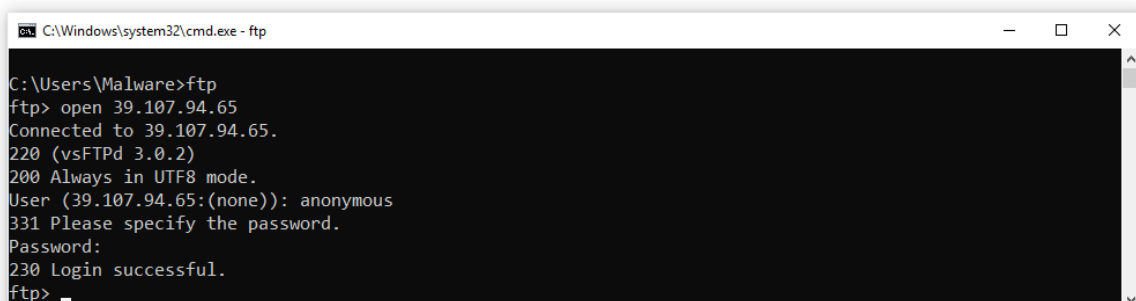
2. Type *ftp* and press Enter.



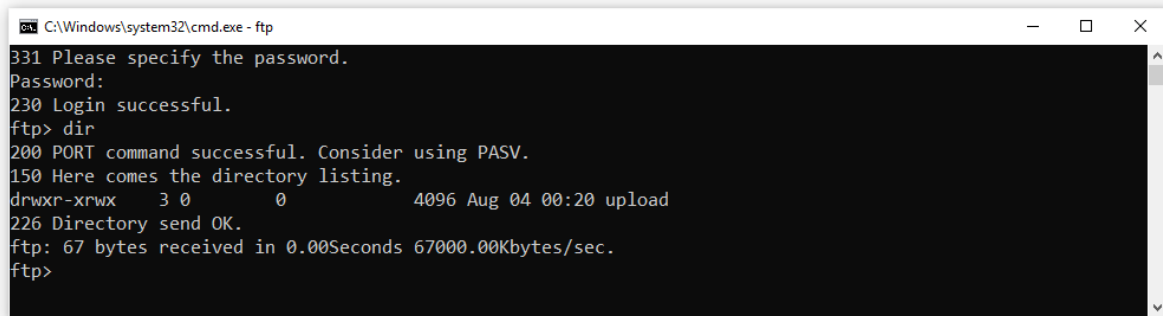
3. Connect to the server.



4. Enter your username and password when prompted.

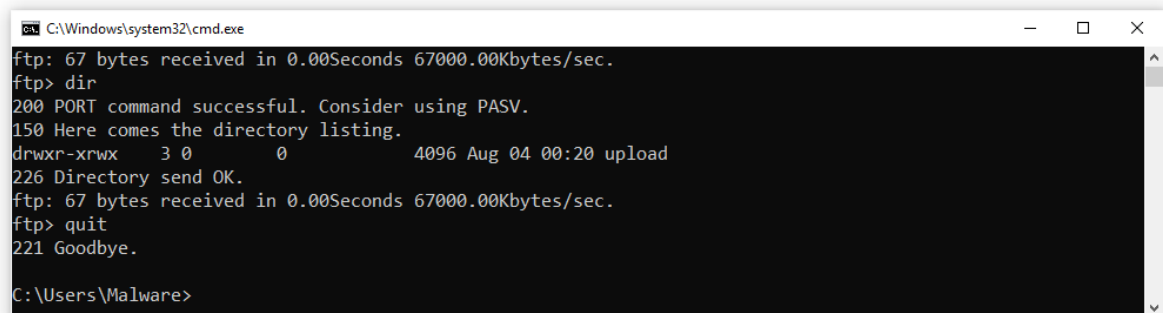


5. You can now use FTP commands like **dir**, **cd**, **get**, **put**, etc., to navigate and manage files.



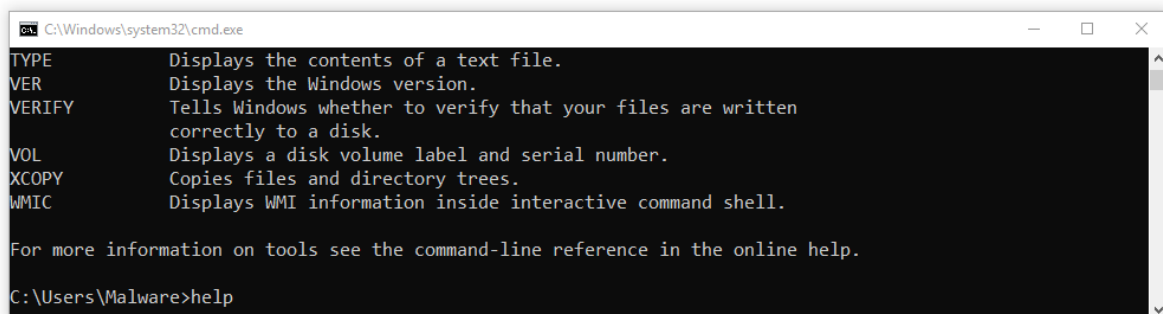
```
C:\Windows\system32\cmd.exe - ftp
331 Please specify the password.
Password:
230 Login successful.
ftp> dir
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
drwxr-xrwx   30      0      4096 Aug 04 00:20 upload
226 Directory send OK.
ftp: 67 bytes received in 0.00Seconds 67000.00Kbytes/sec.
ftp>
```

6. To exit, type quit and press Enter.



```
C:\Windows\system32\cmd.exe
ftp: 67 bytes received in 0.00Seconds 67000.00Kbytes/sec.
ftp> dir
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
drwxr-xrwx   30      0      4096 Aug 04 00:20 upload
226 Directory send OK.
ftp: 67 bytes received in 0.00Seconds 67000.00Kbytes/sec.
ftp> quit
221 Goodbye.
C:\Users\Malware>
```

For a list of available commands, type help after establishing an FTP connection.



```
C:\Windows\system32\cmd.exe
TYPE      Displays the contents of a text file.
VER       Displays the Windows version.
VERIFY    Tells Windows whether to verify that your files are written
          correctly to a disk.
VOL       Displays a disk volume label and serial number.
XCOPY     Copies files and directory trees.
WMIC      Displays WMI information inside interactive command shell.

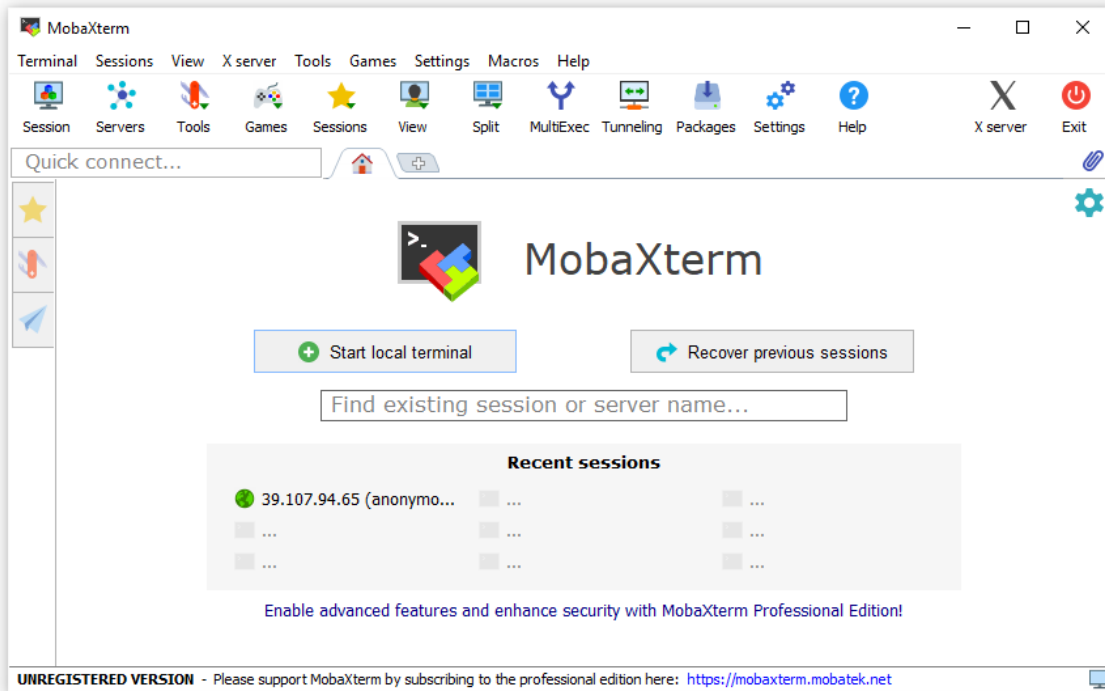
For more information on tools see the command-line reference in the online help.
C:\Users\Malware>help
```

Connecting to RDP

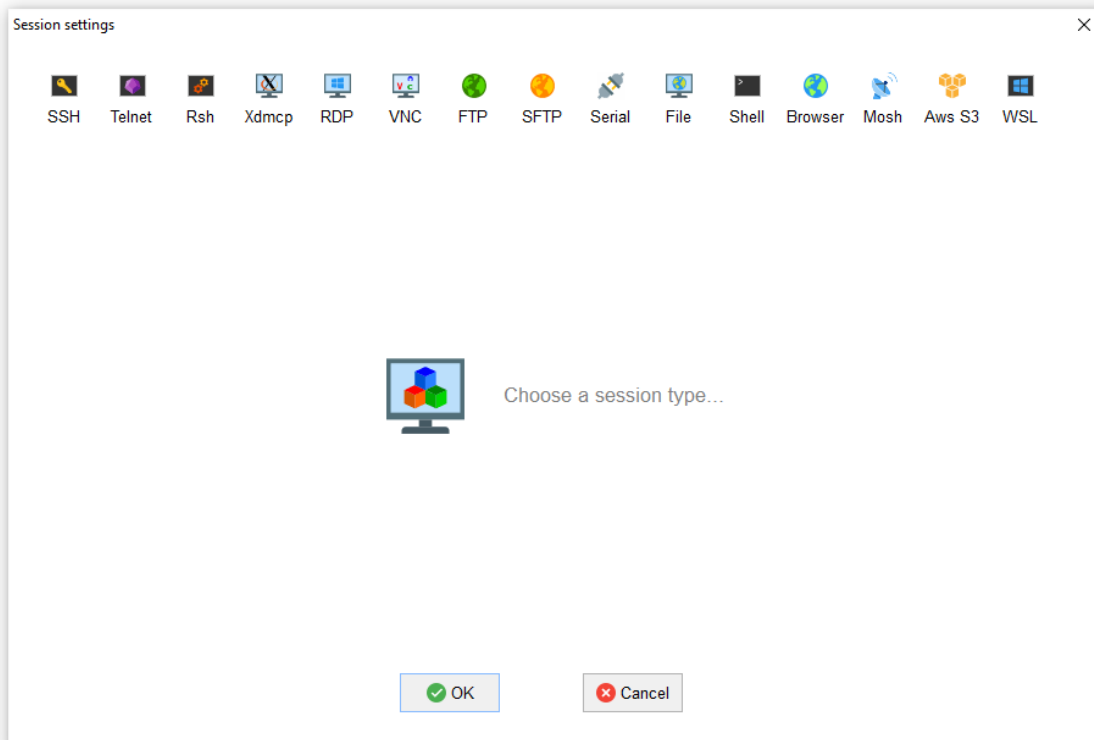
Via MobaXterm

MobaXterm is a versatile tool that offers a wide range of network tools, including RDP.

1. Launch MobaXterm: Start the MobaXterm application on your machine.



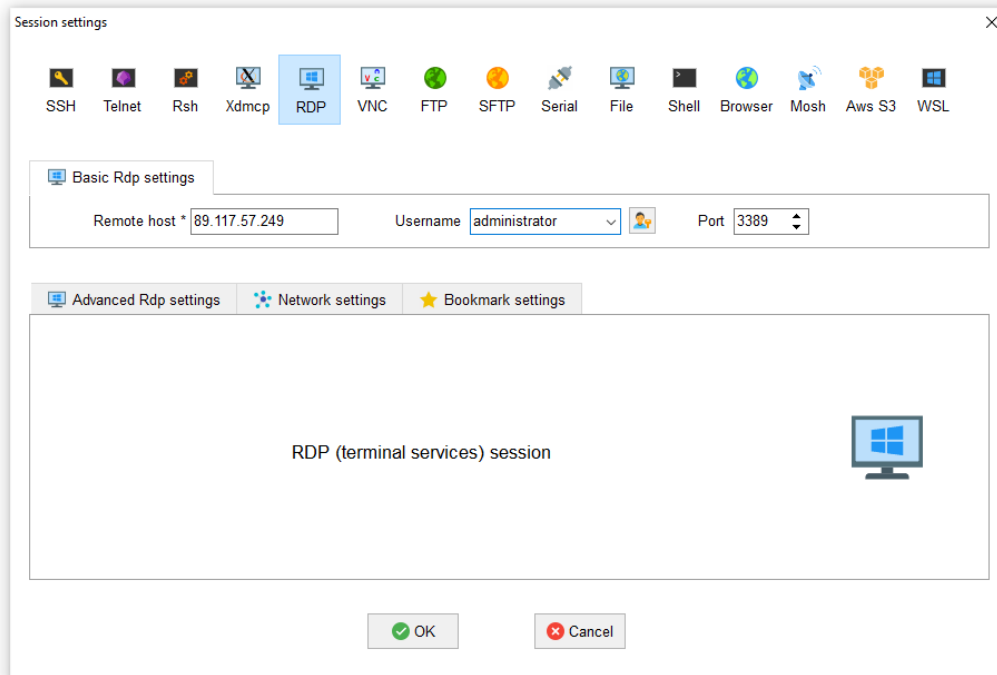
2. Initiate a New Session: Click on the "Session" button located at the top left corner.



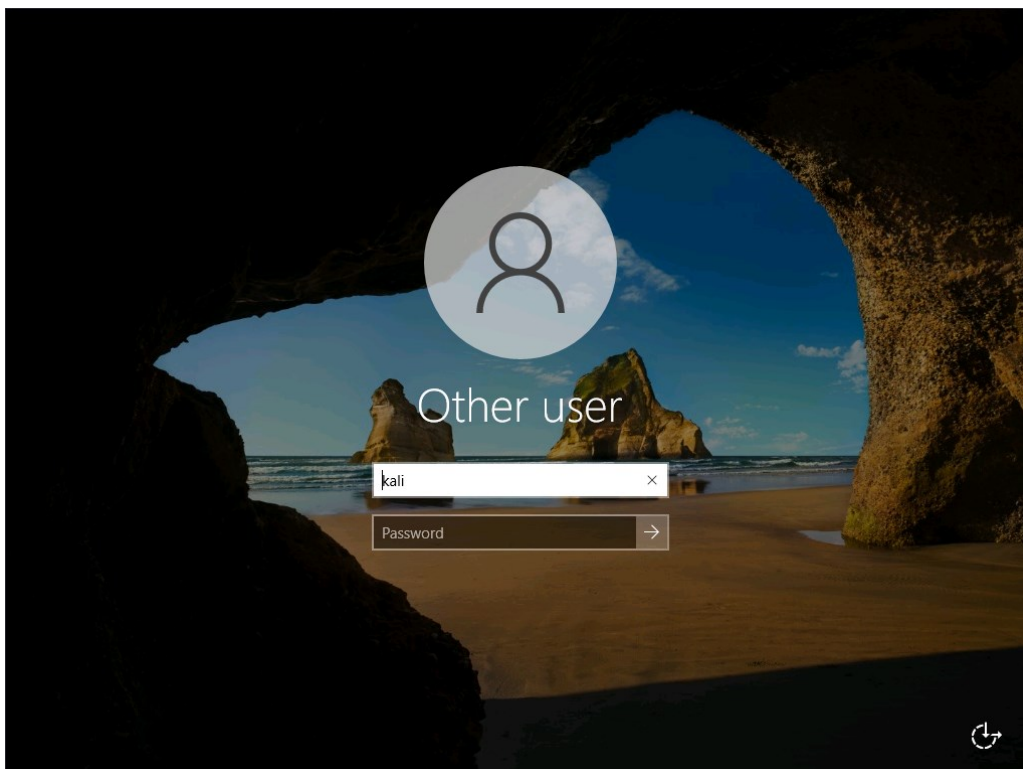
3. Choose RDP Protocol: From the session type list, select "RDP".

Configuration:

- Remote host: Enter the address of the machine you want to connect to.
- Username: Provide the username if known (can be left blank to enter during the connection).
- Port: Typically, this is set to 3389 for RDP, but adjust if necessary.



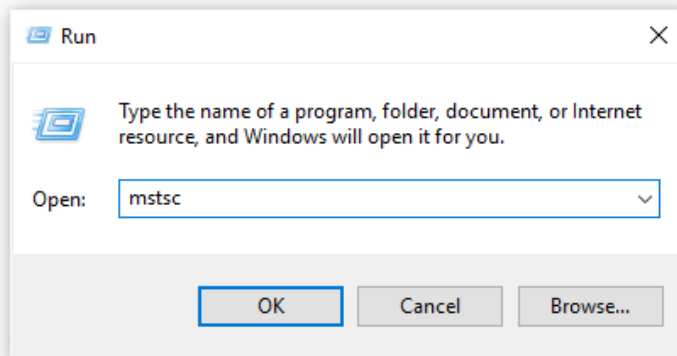
4. Initiate Connection: Click "OK" to initiate the RDP connection.



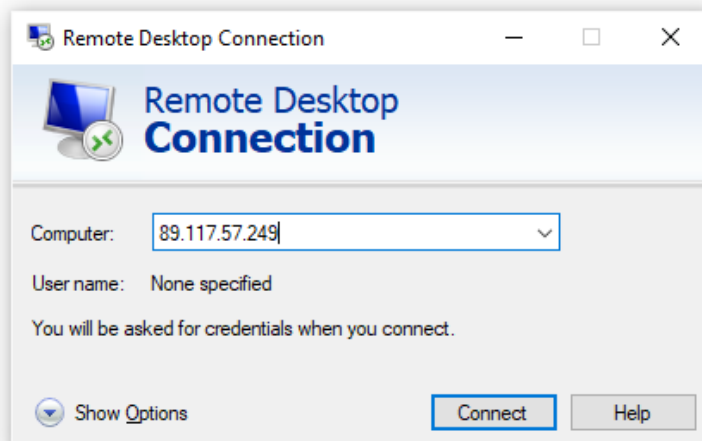
Via mstsc (Microsoft Terminal Services Client)

mstsc is the built-in Remote Desktop client for Windows, allowing you to connect to other Windows machines using RDP.

1. Launch mstsc: Press Win + R on your keyboard to open the "Run" dialog. Type mstsc and hit Enter.



Enter Credentials: If not previously provided, you'll be prompted for the password for the provided username. Input the necessary credentials.

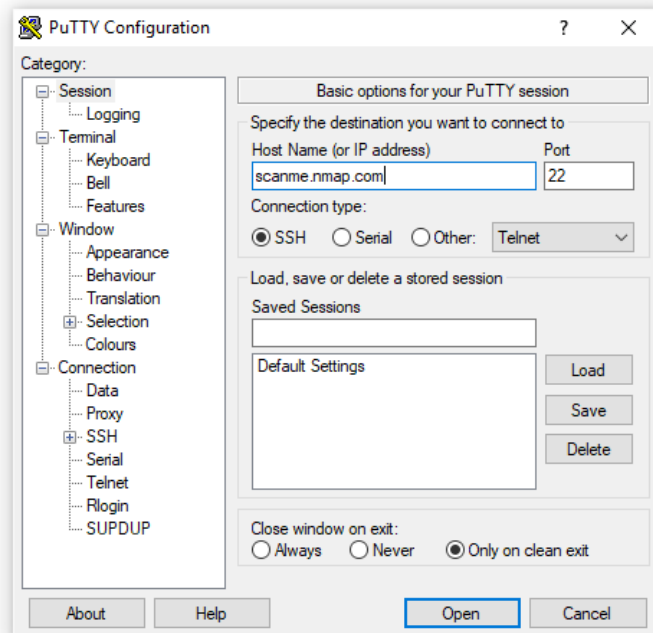


Connecting to SSH

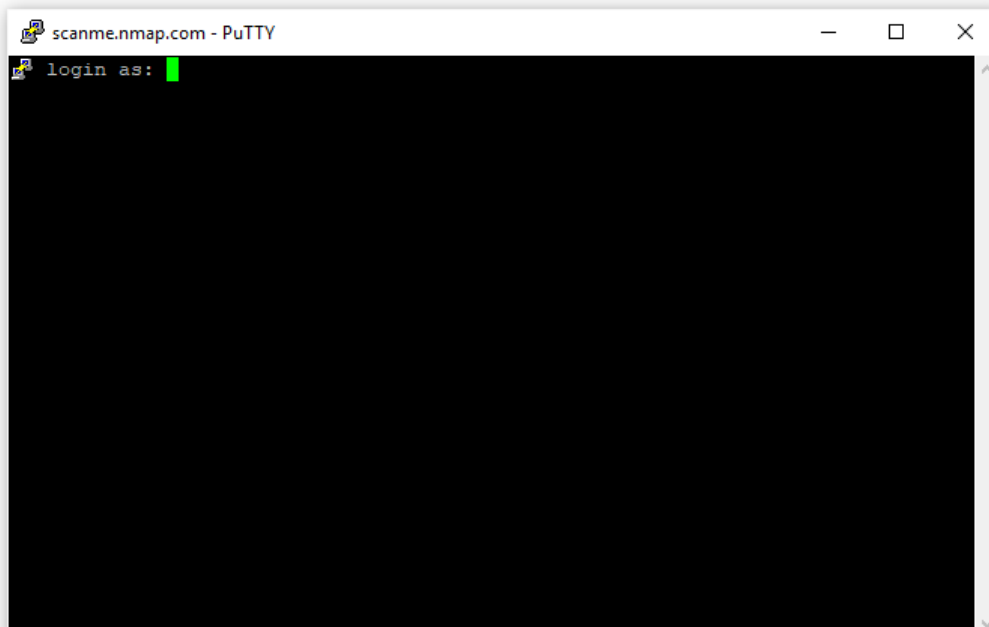
Via Putty

Putty is one of the most popular SSH clients for Windows.

1. **Launch Putty:** Start the Putty application on your machine.



2. **Initiate Connection:** Click "Open" to initiate the SSH connection.

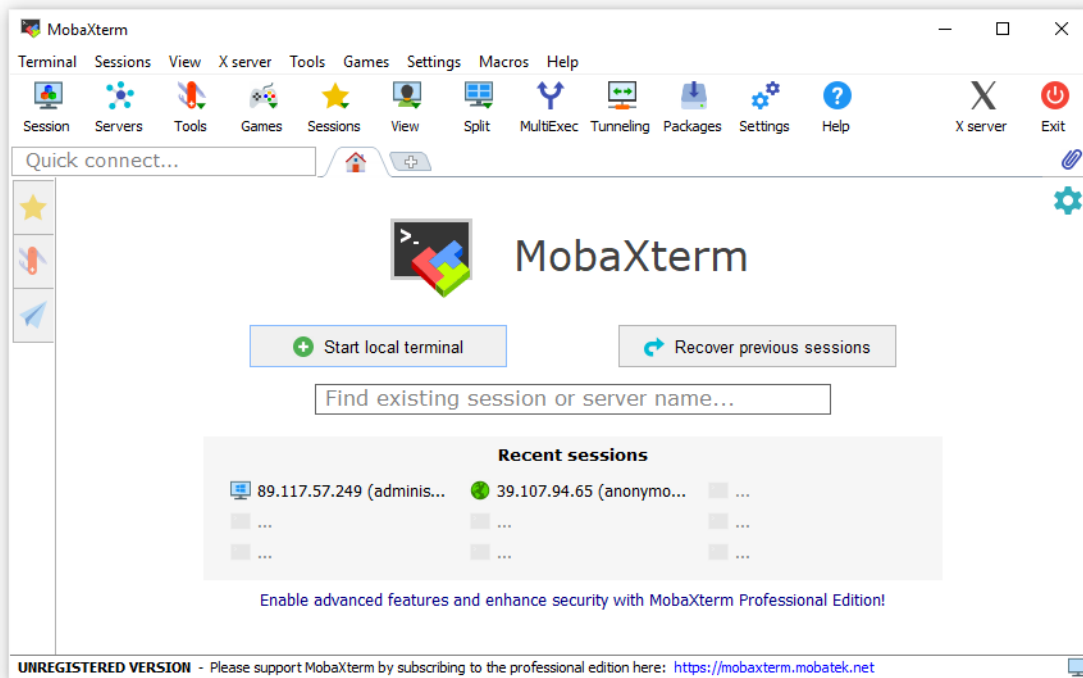


Enter Credentials: You'll be prompted in the terminal window for the username and password for the remote machine. Input the necessary credentials.

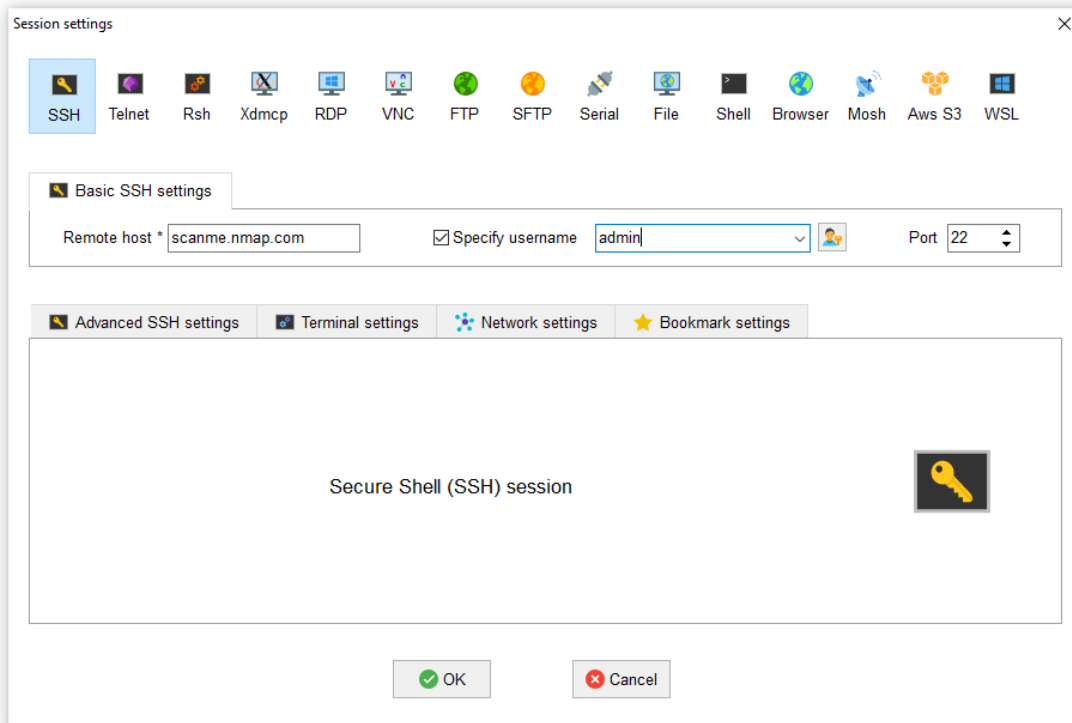
Via MobaXterm

MobaXterm provides advanced terminal functionality and SSH integration.

1. Launch MobaXterm: Start the application.



2. Initiate a New Session: Click on the "Session" button and choose SSH.



3. Initiate Connection: Click "OK" to start the SSH connection.