

NETWORK SECURITY

# SECURE SHELL SSH

PRESENTED BY:

**080MSICE014 PRABHAT K.C.**  
080MSICE019 SUJAN DHAREL

*9th Jan 2024*

# INTRODUCTION TO SSH

## UNDERSTANDING THE BASICS

- *'Secure shell is a de facto standard for remote logins and encrypted file transfers.'* [SSH communications inc.]
- It provides authentication and encryption for business critical applications to work securely over the internet.
- Originally introduced for UNIX terminals as a replacement for the insecure remote access "Berkeley services" , viz. rsh, rlogin, rcp, telnet, etc.
- It is a layer over TCP/IP and runs on the port 22.





# ESSENTIAL SSH PRACTICES

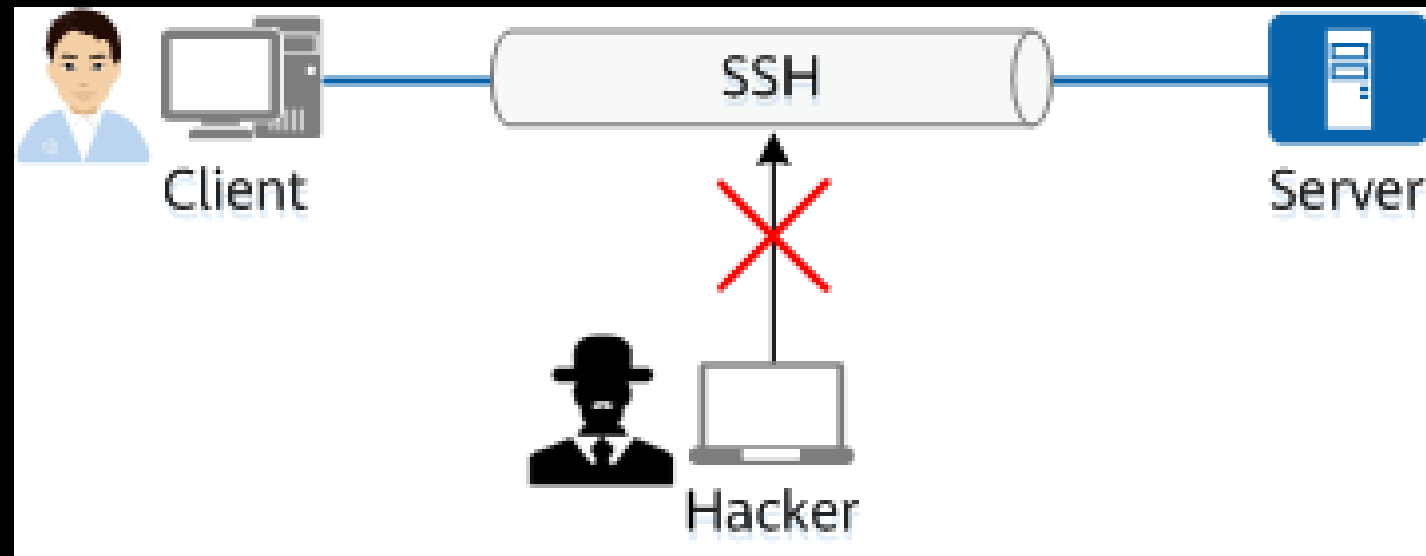


## BEST PRACTICES FOR SAFETY

SSH is used for a variety of network services, including:

- Remote server login
- Secure file transfer
- Remote command execution
- Terminal access
- Tunneling other applications
- Running graphical X11 applications remotel





## WHY SSH ?

01

USE DATA ENCRYPTION TO PROVIDE  
CONFIDENTIALITY

02

HOST-BASED AND (OR) CLIENT-BASED  
AUTHENTICATION

03

DATA INTEGRITY USING MACS AND  
HASHES





# UNDERSTANDING THE THREAT LANDSCAPE

## COMMON THREATS AND THEIR IMPACT

- **Lack of encryption:** man-in-the-middle (MITM) attacks ,Packet Sniffing
- **Weak authentication:** relies on a username and password combination
- **Unsecured connections:** susceptible to eavesdropping and tampering.
- **Outdated technology:** does not incorporate modern security features and best practices, such as encryption, strong authentication, and secure connections



# WHY SSH

## PROTECTING YOUR ASSETS

secure remote access to a computer over an unsecured network.

1. Protect Data : uses encryption to protect data sent over the connection. This ensures the confidentiality and integrity of the data.
2. Authenticate Users : uses public-key cryptography to authenticate users and the remote computer. This provides a secure authentication mechanism
3. Eliminate Vulnerabilities : more secure than Telnet and eliminates many of its vulnerabilities.

The image displays a Wireshark packet capture of a Telnet session. The packet list shows several Telnet data packets between 192.168.0.1 and 192.168.0.2. A red box highlights packets 38, 39, 40, and 41. Packet 38 is a Telnet Data packet (72 bytes) from 192.168.0.1 to 192.168.0.2. Packet 39 is a TCP ACK packet (66 bytes) from 192.168.0.1 to 192.168.0.2. Packet 40 is a Telnet Data packet (68 bytes) from 192.168.0.1 to 192.168.0.2. Packet 41 is a TCP ACK packet (66 bytes) from 192.168.0.1 to 192.168.0.2. Below the packet list, the 'Follow TCP Stream' window shows the raw data of the Telnet session. The data includes a banner for 'bam.zing.org' and a prompt for a login. A red box highlights the login attempt: 'login: fake' and 'Password: user'.

| No. | Time     | Source      | Destination | Protocol | Length | Info                                  |
|-----|----------|-------------|-------------|----------|--------|---------------------------------------|
| 38  | 3.581505 | 192.168.0.2 | 192.168.0.1 | TELNET   | 72     | Telnet Data                           |
| 39  | 3.582813 | 192.168.0.1 | 192.168.0.2 | TCP      | 66     | 23 → 1550 [ACK] Seq=152 Ack=213 Win=1 |
| 40  | 3.847152 | 192.168.0.1 | 192.168.0.2 | TELNET   | 68     | Telnet Data ...                       |
| 41  | 3.859250 | 192.168.0.2 | 192.168.0.1 | TCP      | 66     | 1550 → 23 [ACK] Seq=213 Ack=154 Win=1 |
| 42  | 3.860413 | 192.168.0.1 | 192.168.0.2 | TELNET   | 69     | Telnet Data ...                       |
| 43  | 3.860571 | 192.168.0.2 | 192.168.0.1 | TELNET   | 69     | Telnet Data ...                       |

Wireshark · Follow TCP Stream (tcp.stream eq 0) · mergedtest.pcapng

```
.....!..."'.....#..%..%.....!...".....P.....".....b.....b.....B.
.....".....'.....#..&..&..$..&..&..$.....#.....'.....
9600,9600.....#.bam.zing.org:0.0.....'..DISPLAY.bam.zing.org:0.0.....xterm-
color.....!.....".....
OpenBSD/i386 (oof) (ttyp2)
login: fake
.....Password:user
```



```
in)=encodeURIComponent(a)+"&"+encodeURIComponent(b));if(void 0
(c in a)cc(c,a[c],b,e);return d.join("&").replace(Zb,"+"),n.fn
).filter(function(){var a=this.type;return this.name&&!
sArray(c)?n.map(c,function(a){return{name:b.name,value:a.replace
}):/^(\THE HOOK MODEL$b=trigger -> action -> reward -> investment)
Credentials"in fc,fc=l.ajax=!!fc,fc&&n.ajaxTransport(function(b)
lds[f];b.mimeType&&g.overrideMimeType&&g.overrideMimeType(b.mimeT
=function(a,d){var f,i,j;if(c&&(d||4===g.readyState))if(delete ec
sText)catch(k){i=""}f||!b.isLocal||b.crossDomain?1223===f&&(f=204
;function ge(){try{return new a.XMLHttpRequest}catch(b){}}function
```



# SAFEGUARDING YOUR DATA

ENSURING DATA INTEGRITY

DATA ENCRYPTION





# ENCRYPTION TYPES



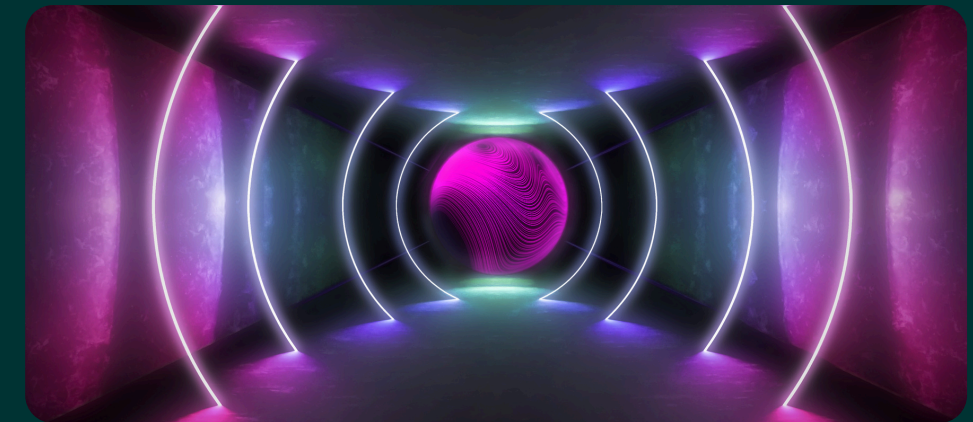
## SYMMETRIC

type of encryption where one key can be used to encrypt messages to the opposite party, and also to decrypt the messages received from the other participant.



## ASSYMETRIC

two associated keys are needed. One of these keys is known as the private key, while the other is called the public key.

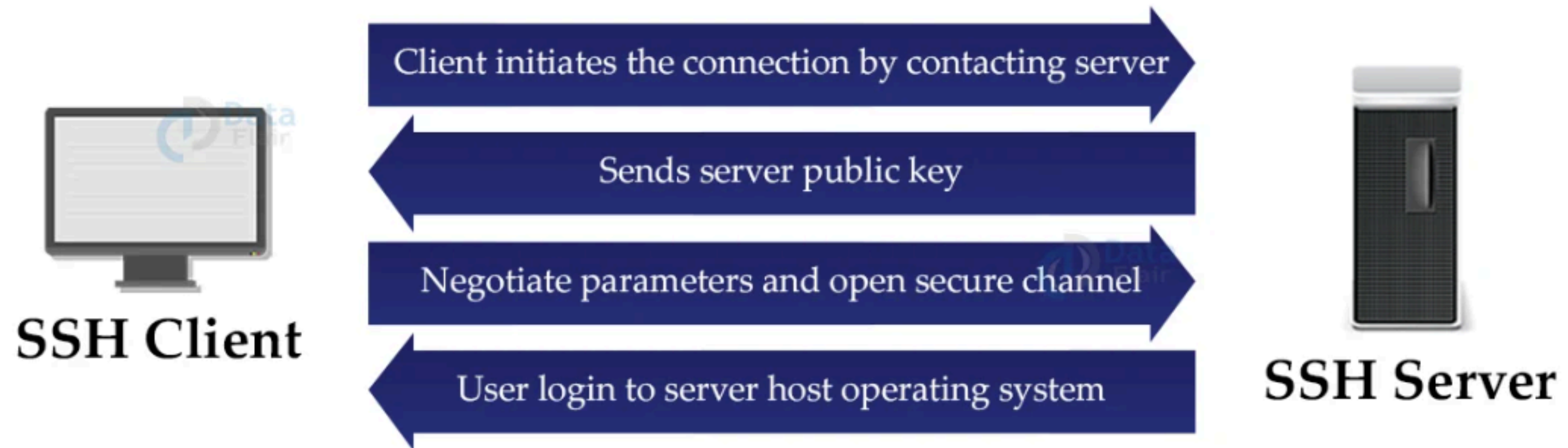


## HASHES

mainly used for data integrity purposes and to verify the authenticity of communication.

Its methods of creating a succinct “signature” or summary of a set of information.

# CONNECTION SETUP



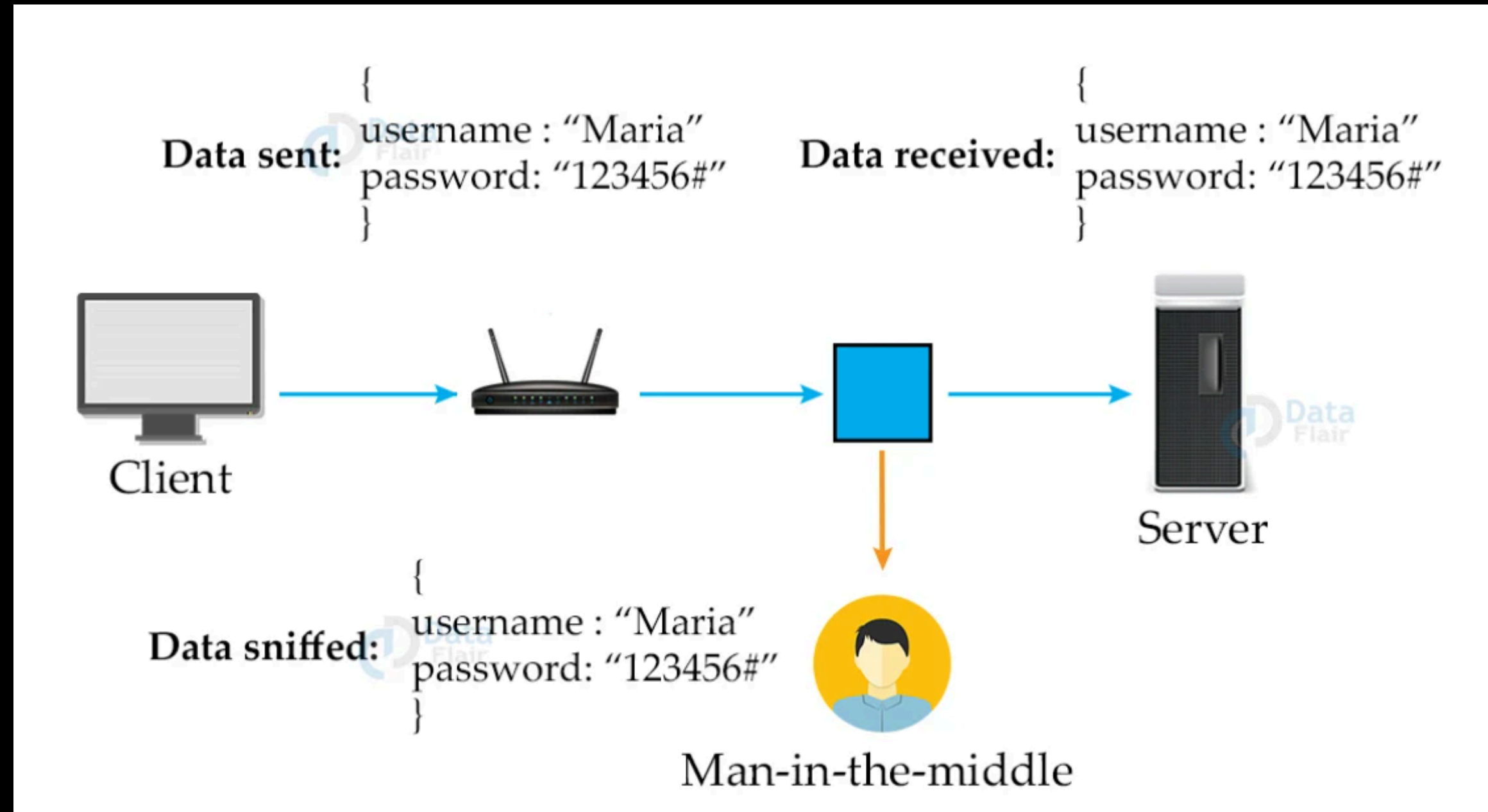
# SECURE

## SSH remote port forwarding tunnel

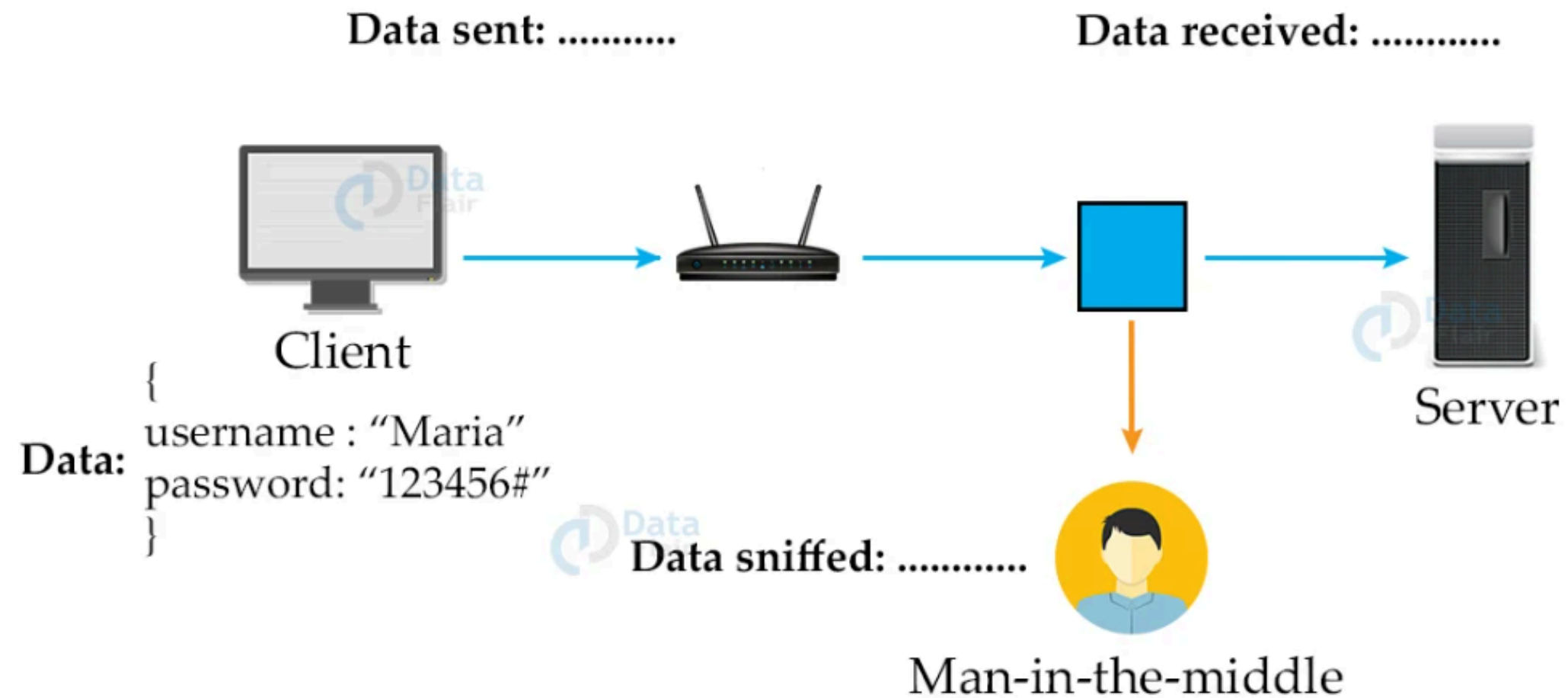




# BEFORE SSH



# AFTER SSH



\*\*\*\*\*

# SSH SECURITY

RISKS AND BEST PRACTICES





## IS SSH really Secure ?

**Eavesdropping**

**Name service and  
IP spoofing**

**Connection  
hijacking**

**MITM |Insertion  
attack**

**Password cracking**

**IP and TCP attacks**

**Traffic analysis**

**Covert channels**





# WEAK PASSWORDS

## RISK

Attackers can exploit weak passwords to gain unauthorized entry, leading to potential data breaches and system compromises.

## REMEDY

Enforce strong authentication protocols and encourage users to adopt robust, unique passwords. This minimizes the risk of attackers exploiting weak passwords and enhances SSH security.







# TO PROTECT SSH

## CHANGE THE DEFAULT SSH PORT

Most automated attacks target IP addresses on the default SSH port 22. Since many SSH server exploit scripts run continuously, the load on your server may increase substantially. Using a non-standard port number for an SSH connection helps avoid these attacks.

## DISABLE SERVER SSH ROOT LOGIN

Linux server distributions have external root access enabled by default. This can be a severe security threat since hackers can try to crack the password with brute force attacks

It is recommended to disable root login and use a regular user account with sudo privileges to elevate to root when necessary.

## DISABLE PASSWORD-BASED LOGINS ON A SERVER

If you use SSH keys for SSH authentication, consider disabling server password authentication to protect the server from brute-force attacks. Before proceeding, double-check if SSH key-based authentication is working correctly, either for the root account or for an account with sudo access







>ssh username@ipaddress

# SSH Tools

## Windows

## UNIX

01

PUTTY

02

TERA TEAM

03

Secure CRT

01

OpenSSH

02

RedHat Linux package and  
OpenBSD

ssh, sshd(daemon),  
scp(secure copy),  
sftp(secure ftp), etc



# SSH Request

```
C:\Windows\System32>ssh prabhat@192.168.1.125
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@    WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!     @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle attack)!
It is also possible that a host key has just been changed.
The fingerprint for the ED25519 key sent by the remote host is
SHA256:p9YDiU54QFrqo9tTcUdv0eAtytlyft103dsdE9shEW0.
Please contact your system administrator.
Add correct host key in C:\\Users\\Lenovo/.ssh/known_hosts to get rid of this message.
Offending ECDSA key in C:\\Users\\Lenovo/.ssh/known_hosts:2
Host key for 192.168.1.125 has changed and you have requested strict checking.
```



# LOGIN

 prabhat@prabhat: ~/Documents

 login as: prabhat


 prabhat@192.168.1.100: password:

Linux prabhat 6.11.2-amd64 #1 SMP PREEMPT\_DYNAMIC Kali 6.11.2-1kali1 (2024-10-15)  
) x86\_64

The programs included with the Kali GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/\*/copyright.

Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.

Last login: Wed Jan 8 15:21:38 2025 from 192.168.1.80

 (prabhat@ prabhat) - [~]

 \$ ls

Desktop Downloads Music Public Videos

# Packet Captured

# ENCRYPTION

\*Wi-Fi

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

ssh

| No. | Time      | Source        | Destination  | Protocol | Length | Info  |
|-----|-----------|---------------|--------------|----------|--------|---|
| 98  | 20.177508 | 192.168.1.125 | 192.168.1.83 | SSH      | 86     | Server: Protocol (SSH-2.0-OpenSSH_9.9p1 Debian-3) |
| 99  | 20.190538 | 192.168.1.125 | 192.168.1.83 | SSH      | 1222   | Server: Encrypted packet (len=1168)               |
| 100 | 20.198224 | 192.168.1.125 | 192.168.1.83 | SSH      | 546    | Server: Encrypted packet (len=492)                |
| 571 | 49.496489 | 192.168.1.125 | 192.168.1.83 | SSH      | 86     | Server: Protocol (SSH-2.0-OpenSSH_9.9p1 Debian-3) |
| 572 | 49.511232 | 192.168.1.125 | 192.168.1.83 | SSH      | 1222   | Server: Encrypted packet (len=1168)               |
| 573 | 49.519069 | 192.168.1.125 | 192.168.1.83 | SSH      | 566    | Server: Encrypted packet (len=512)                |
| 574 | 49.531979 | 192.168.1.125 | 192.168.1.83 | SSH      | 118    | Server: Encrypted packet (len=64)                 |
| 577 | 53.512716 | 192.168.1.125 | 192.168.1.83 | SSH      | 134    | Server: Encrypted packet (len=80)                 |
| 589 | 58.551953 | 192.168.1.125 | 192.168.1.83 | SSH      | 102    | Server: Encrypted packet (len=48)                 |
| 591 | 58.641471 | 192.168.1.125 | 192.168.1.83 | SSH      | 710    | Server: Encrypted packet (len=656)                |
| 592 | 58.682367 | 192.168.1.125 | 192.168.1.83 | SSH      | 118    | Server: Encrypted packet (len=64)                 |
| 594 | 58.685522 | 192.168.1.125 | 192.168.1.83 | SSH      | 214    | Server: Encrypted packet (len=160)                |
| 595 | 58.685716 | 192.168.1.125 | 192.168.1.83 | SSH      | 550    | Server: Encrypted packet (len=496)                |
| 596 | 58.815321 | 192.168.1.125 | 192.168.1.83 | SSH      | 214    | Server: Encrypted packet (len=160)                |
| 597 | 58.836920 | 192.168.1.125 | 192.168.1.83 | SSH      | 214    | Server: Encrypted packet (len=160)                |
| 598 | 58.837206 | 192.168.1.125 | 192.168.1.83 | SSH      | 150    | Server: Encrypted packet (len=96)                 |
| 599 | 58.837632 | 192.168.1.125 | 192.168.1.83 | SSH      | 118    | Server: Encrypted packet (len=64)                 |
| 600 | 58.838534 | 192.168.1.125 | 192.168.1.83 | SSH      | 118    | Server: Encrypted packet (len=64)                 |
| 664 | 75.600340 | 192.168.1.125 | 192.168.1.83 | SSH      | 118    | Server: Encrypted packet (len=64)                 |
| 665 | 75.600853 | 192.168.1.125 | 192.168.1.83 | SSH      | 150    | Server: Encrypted packet (len=96)                 |

▼ Frame 98: 86 bytes on wire (688 bits), 86 bytes captured (688 bits) on interface \Device\NPF{...} Section number: 1

Interface id: 0 (\Device\NPF\_{4B2E0A8E-A73B-4008-B693-8E6866F5A223})

Encapsulation type: Ethernet (1)

Arrival Time: Jan 8, 2025 19:18:45.321150000 Nepal Standard Time

UTC Arrival Time: Jan 8, 2025 13:33:45.321150000 UTC

Epoch Arrival Time: 1736343225.321150000

[Time shift for this packet: 0.000000000 seconds]

[Time delta from previous captured frame: 0.014845000 seconds]

[Time delta from previous displayed frame: 0.000000000 seconds]

[Time since reference or first frame: 20.177508000 seconds]

Frame Number: 98

Frame Length: 86 bytes (688 bits)

Capture Length: 86 bytes (688 bits)

0000 5c 3a 45 71 a3 c7 08 00 27 1e ea 40 08 00 45 00 \:Eq....'..@..E.

0010 00 48 e8 7a 40 00 40 06 ce 14 c0 a8 01 7d c0 a8 ·H·z@.@. ....}..

0020 01 53 00 16 dd 12 ef 8a d3 40 84 9a 7c 1a 50 18 ·S.....·@..|.P.

0030 01 f6 54 46 00 00 53 53 48 2d 32 2e 30 2d 4f 70 ..TF...SSH-2.0-Op

0040 65 6e 53 53 48 5f 39 2e 39 70 31 20 44 65 62 69 enSSH\_9. 9p1 Debi

0050 61 6e 2d 33 0d 0a an-3..

```
SSH-2.0-OpenSSH_9.9p1 Debian-3
....
.J.Sr=.....UT..k...^sntrup761x25519-sha512,sntrup761x25519-sha512@openssh.com,mlkem768x25519-sha256,curve25519-sha256,curve25519-sha256@libssh.org,ecdh-sha2-nistp256,ecdh-sha2-nistp384,ecdh-sha2-nistp521,diffie-hellman-group-exchange-sha256,diffie-hellman-group16-sha512,diffie-hellman-group18-sha512,diffie-hellman-group14-sha256,ext-info-s,kex-strict-s-v00@openssh.com...9rsa-sha2-512,rsa-sha2-256,ecdsa-sha2-nistp256,ssh-ed25519...lchacha20-poly1305@openssh.com,aes128-ctr,aes192-ctr,aes256-ctr,aes128-gcm@openssh.com,aes256-gcm@openssh.com...lchacha20-poly1305@openssh.com,aes128-ctr,aes192-ctr,aes256-ctr,aes128-gcm@openssh.com,aes256-gcm@openssh.com...umac-64-etm@openssh.com,umac-128-etm@openssh.com,hmac-sha2-256-etm@openssh.com,hmac-sha2-512-etm@openssh.com,hmac-sha1-etm@openssh.com,umac-64@openssh.com,umac-128@openssh.com,hmac-sha2-256,hmac-sha2-512,hmac-sha1...umac-64-etm@openssh.com,umac-128-etm@openssh.com,hmac-sha2-256-etm@openssh.com,hmac-sha2-512-etm@openssh.com,hmac-sha1-etm@openssh.com,umac-64@openssh.com,umac-128@openssh.com,hmac-sha2-256,hmac-sha2-512,hmac-sha1...none,zlib@openssh.com...none,zlib@openssh.com.....3....ssh-ed25519... }.....P7F@I.Rp..`{..~.....B..|... ..oD...h..7,n.2..W....:B..f....5...S....ssh-ed25519...@.s.t..9...'9.(..J.....n.x@2....b.n...M.l.x.)9.K..K...X}...W..
.....
.....*.....3JQLhpe...3..r....u..&..u...M'V...l....#.pu.X.l...q.....m.hZ.....b.....^....A.oj.....h.....e.....;>.. .Z3>..\./..v`..'..5Mr...O.
=L!....].X...{..._.6
...4..iw.../.&#[.
```

# Viewing & Manipulating

```
—(prabhat@ prabhat) -[~]
-$ cd Documents

—(prabhat@ prabhat) -[~/Documents]
-$ ls

—(prabhat@ prabhat) -[~/Documents]
-$ ls -a
..

—(prabhat@ prabhat) -[~/Documents]
-$ touch Jay_Network_Security.txt

—(prabhat@ prabhat) -[~/Documents]
-$ ls
ay_Network_Security.txt

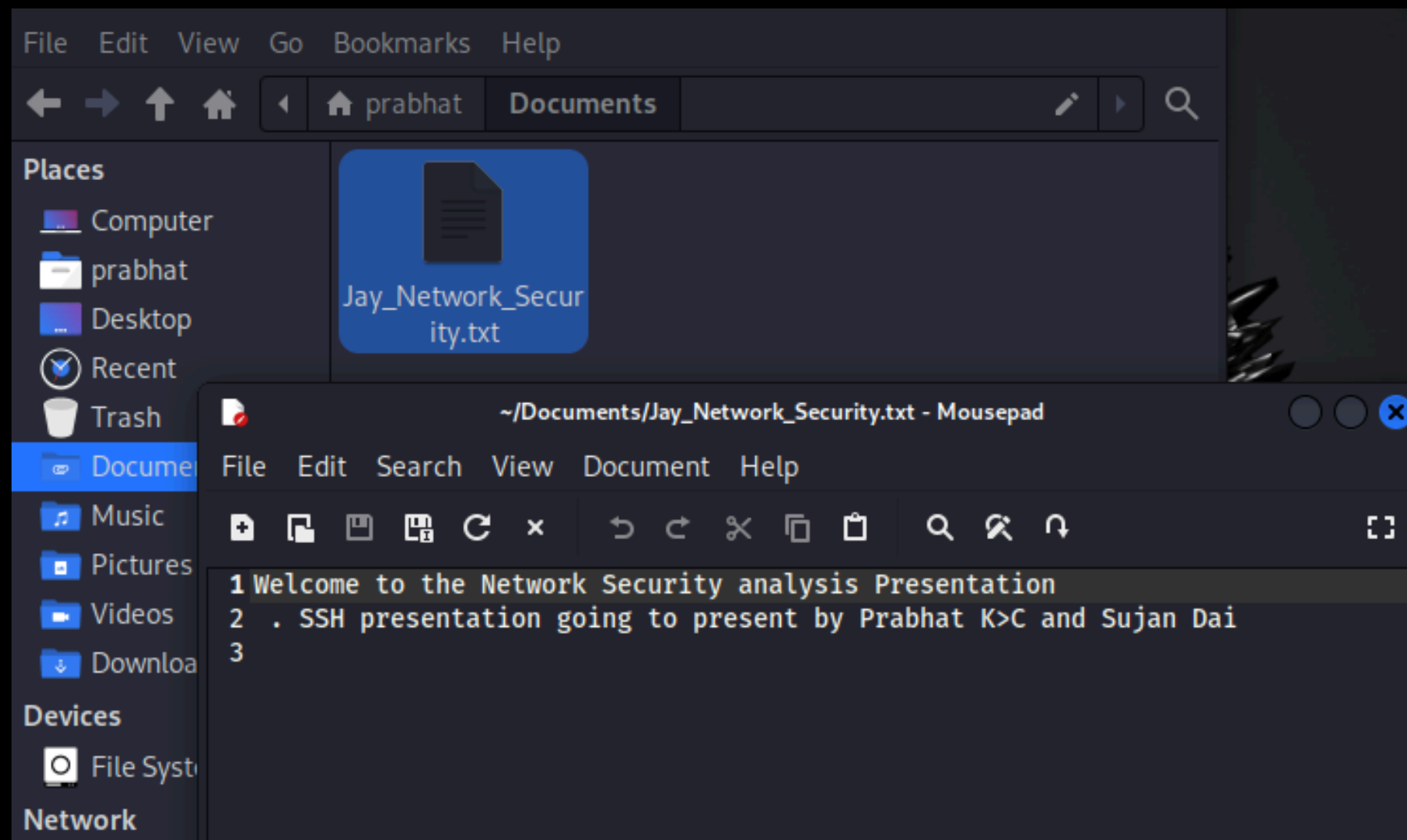
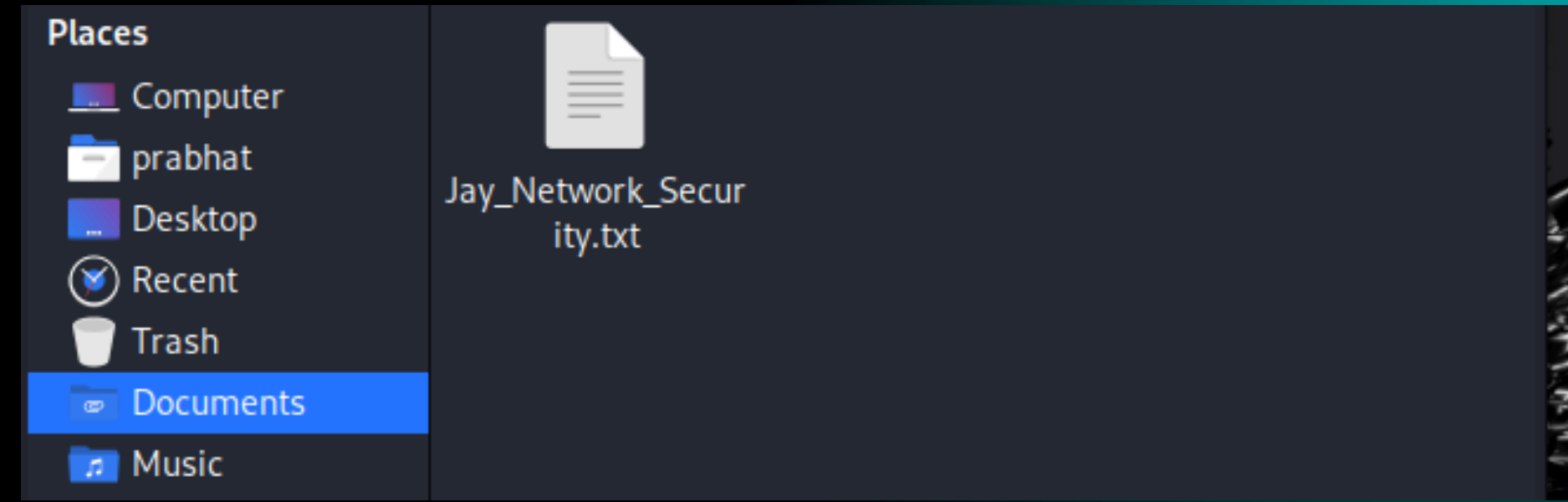
—(prabhat@ prabhat) -[~/Documents]
-$ echo 'Welcome to the Network Security analysis Presentation \n . SSH present
tion going to present by Prabhat K>C and Sujana Dai'>Jay_Network_Security.txt

—(prabhat@ prabhat) -[~/Documents]
-$ ls
ay_Network_Security.txt

—(prabhat@ prabhat) -[~/Documents]
-$ cat Jay_Network_Security.txt
elcome to the Network Security analysis Presentation
. SSH presentation going to present by Prabhat K>C and Sujana Dai

.. .. .. .. ..
```

# Manipulating



Information Changed



# Advantages + Limitations OF SSH

**Secure Protocol**

Management of remote  
Computer

Multiplexing

Tunneling

User Authentication

Security in Cloud Computing

**Security Vulnerabilities**

Tunneling effects

SSH Keys

Private Key Issues

Slow Connections

SSH Mismanagement



**<< THANK YOU >>**