<epam>

Networks Services

Computer Networks

TRAINING

CENTER

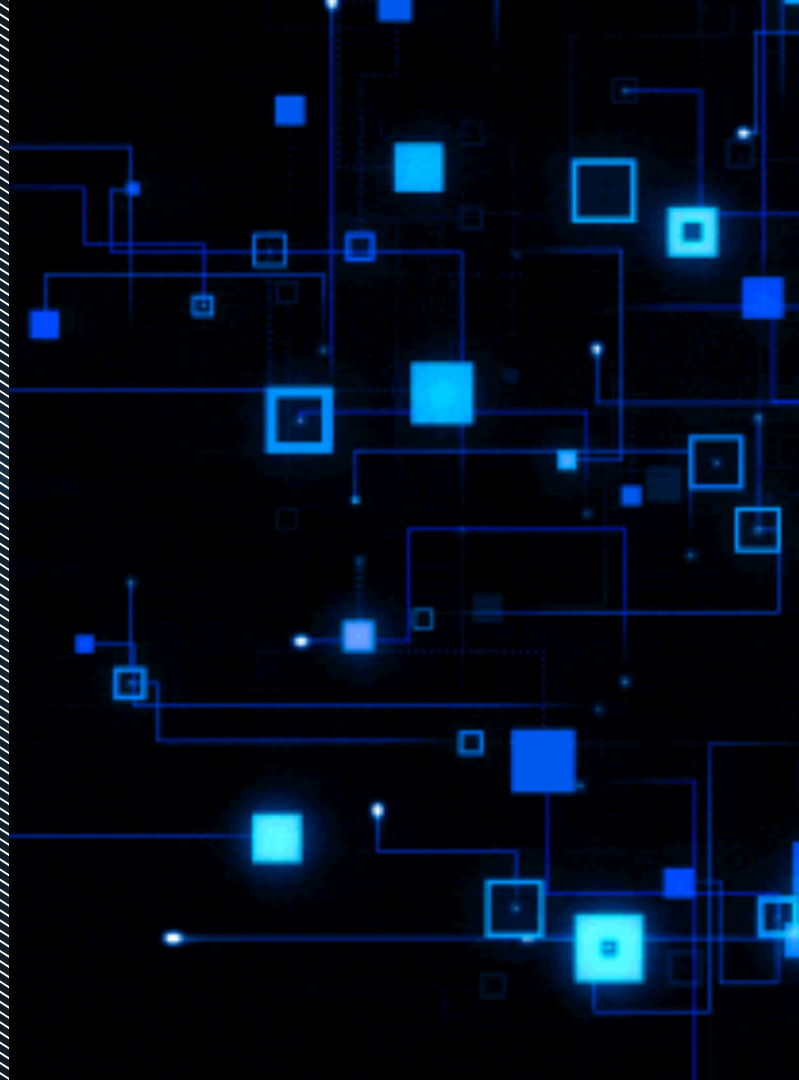
<epam>

Siarhei Kantarovich

<epam>

Agenda

- TELNET
- SSH
- RDP

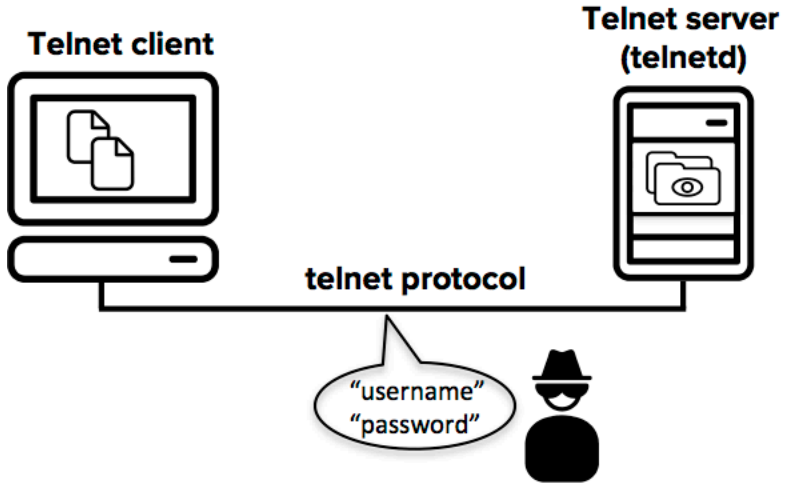


Remote connection

Telnet, developed in 1969, is a protocol that provides a command line interface for communication with a remote device or server, sometimes employed for remote management but also for initial device setup like network hardware. Telnet stands for Teletype Network, but it can also be used as a verb; 'to telnet' is to establish a connection using the Telnet protocol.

Because it was developed before the mainstream adaptation of the internet, Telnet on its own does not employ any form of encryption, making it outdated in terms of modern security. It has largely been overlapped by Secure Shell (SSH) protocol, at least on the public internet, but for instances where Telnet is still in use

Telnet provides users with a bidirectional interactive text-oriented communication system utilizing a virtual terminal connection over 8 byte. User data is interspersed in-band with telnet control information over the transmission control protocol (TCP). Often, Telnet was used on a terminal to execute functions remotely.



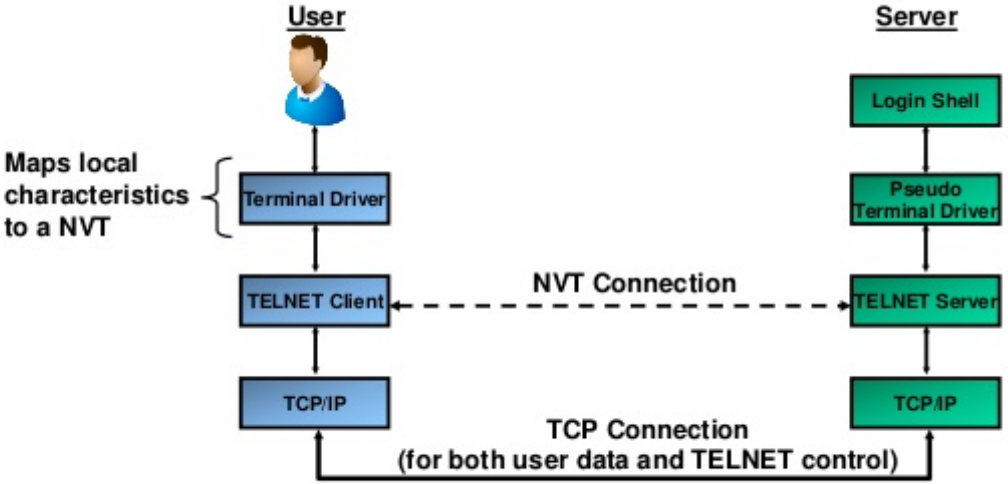
Remote connection

TELNET

indigoo.com

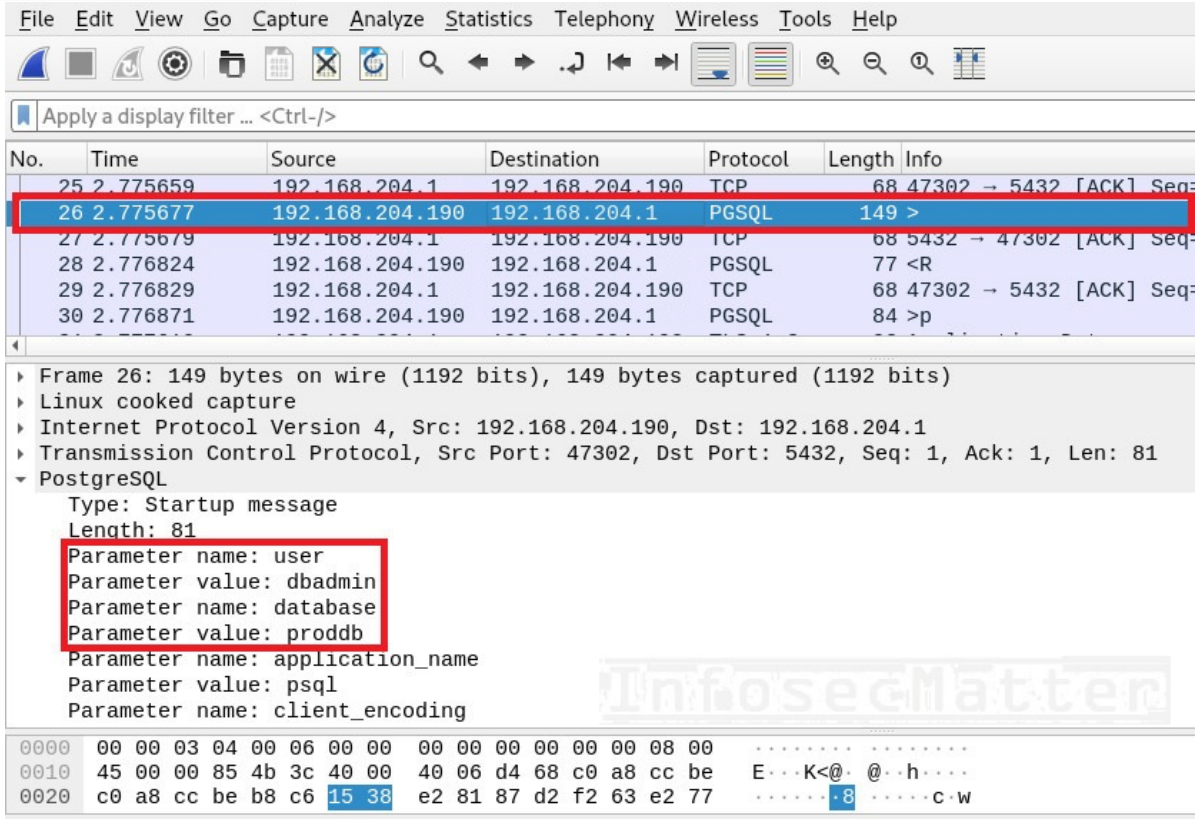
1. What is TELNET?

- TELNET (RFC854) is a protocol providing platform independent, bi-directional byte-oriented communication between hosts (unlike rlogin which is Unix based).
- Most often TELNET is used for remote login to hosts on the Internet.
- TELNET is basically a TCP connection with interspersed TELNET control information.
- TELNET may use option negotiation for providing additional services.



Remote connection

Telnet sessions between the client and the server are not encrypted without a workaround. So those with access to the TCP/IP packet flow between hosts can observe all of the traffic, listen in, and record potentially sensitive information like logins and passwords of users connecting to the Telnet server.

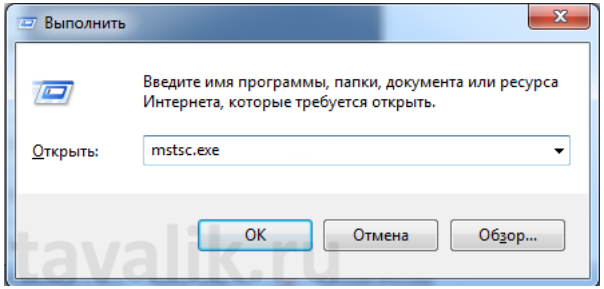


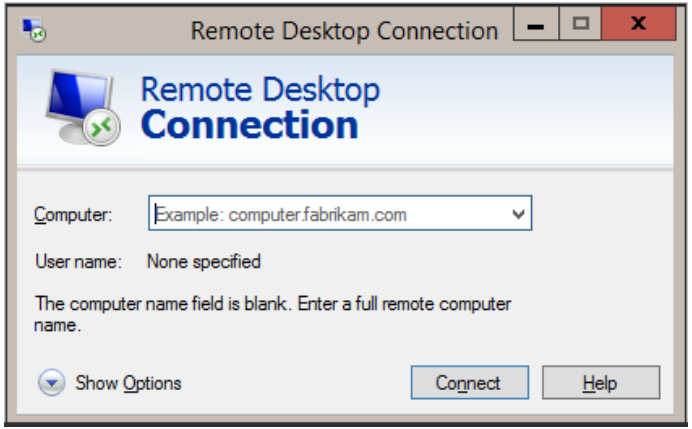
Remote connection

The **Windows Remote Desktop Connection** tool gives users the ability to connect to a remote Windows PC or server over the internet or on a local network, giving them full access to the tools and software installed on it. This is made possible by Microsoft's own **Remote Desktop Protocol** (or **RDP** for short).

For RDP connections to work, **you need two components—an RDP server and an RDP client**. A typical RDP server is the Windows PC or server you're connecting to and will control. The client is a PC or mobile device with an RDP client app installed, from which you control the server. Microsoft offers its own client for Windows, macOS, Android, and iOS, with various third-party options available for Linux and other platforms.

RDP is a Windows-only protocol, and you can only establish remote connections using RDP with Windows PCs and Windows Server installations that support it.



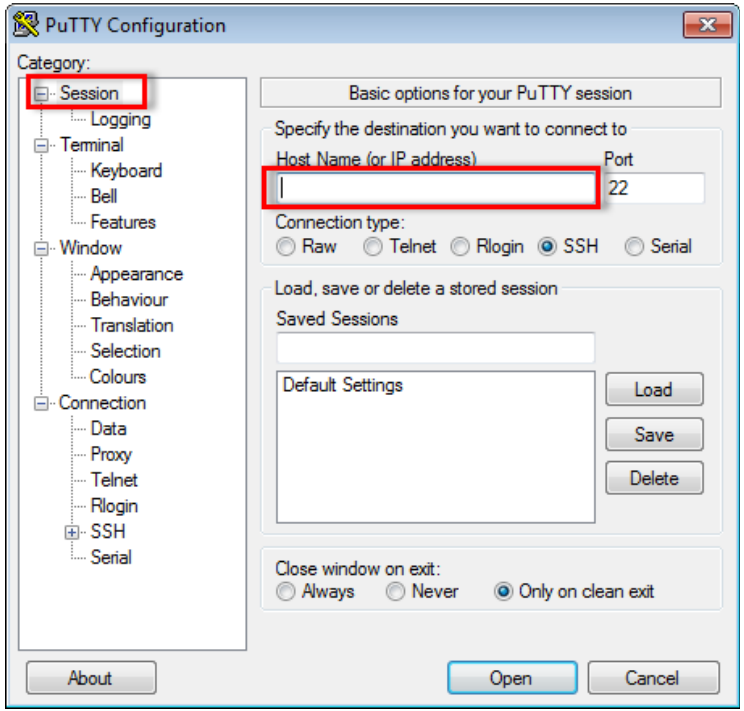


Remote connection

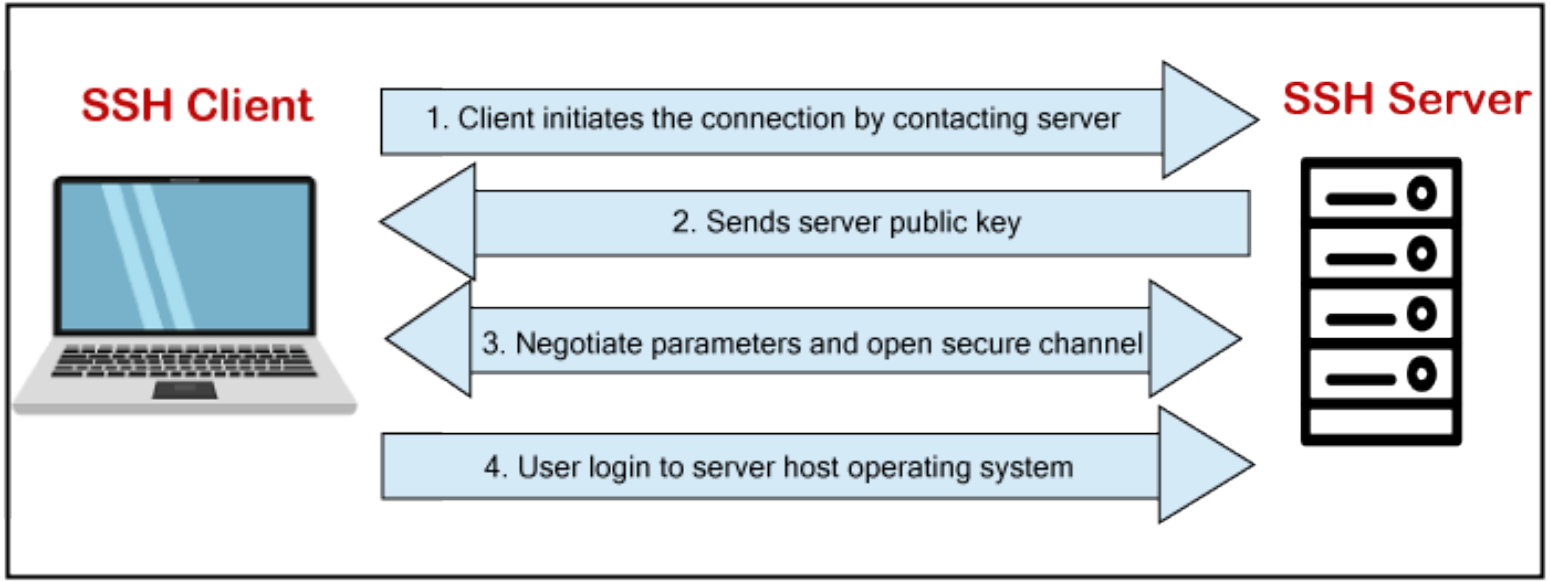
SSH, or Secure Shell, is a remote administration protocol that allows users to control and modify their remote servers over the Internet.
The service was created as a secure replacement for the unencrypted Telnet and uses cryptographic techniques to ensure that all communication to and from the remote server happens in an encrypted manner.

It provides a mechanism for authenticating a remote user, transferring inputs from the client to the host, and relaying the output back to the client.

Any Linux or macOS user can SSH into their remote server directly from the terminal window. Windows users can take advantage of SSH clients like Putty.
You can execute shell commands in the same manner as you would if you were physically operating the remote computer.



Remote connection

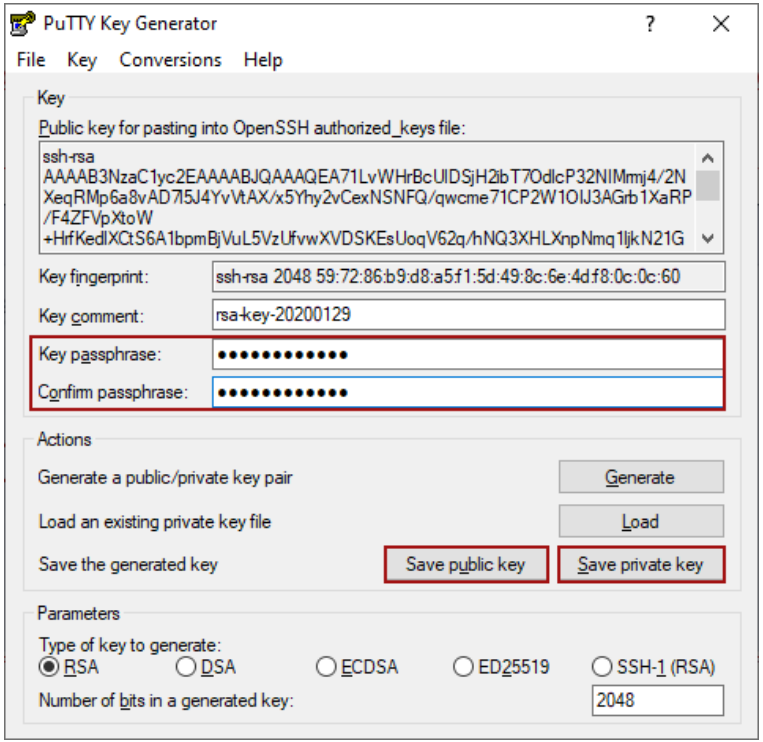


Remote connection

SSH operates on TCP port 22 by default (though this can be changed if needed). The host (server) listens on port 22 (or any other SSH assigned port) for incoming connections. It organizes the secure connection by authenticating the client and opening the correct shell environment if the verification is successful.

There are two stages to establishing a connection: first both the systems must agree upon encryption standards to protect future communications, and second, the user must authenticate themselves.

When a client tries to connect to the server via TCP, the server presents the encryption protocols and respective versions that it supports. If the client has a similar matching pair of protocol and version, an agreement is reached and the connection is started with the accepted protocol. The server also uses an asymmetric public key which the client can use to verify the authenticity of the host.

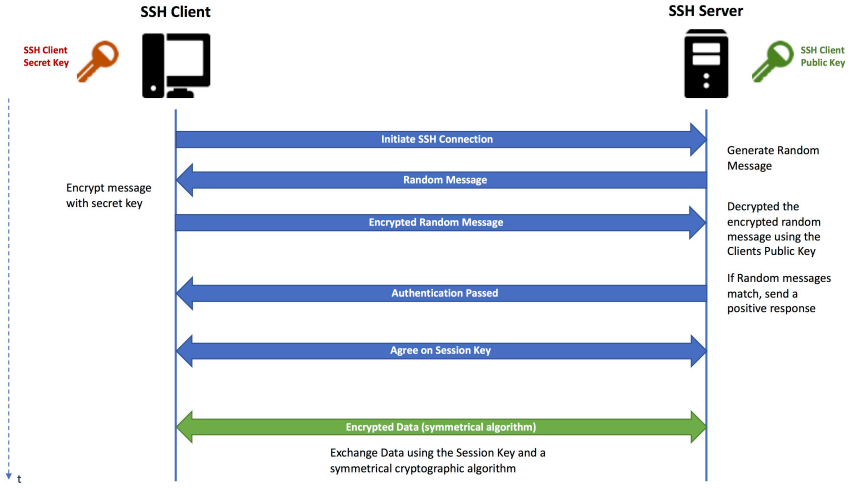


Remote connection

Once this is established, the two parties use what is known as a Diffie-Hellman Key Exchange Algorithm to create a symmetrical key.
This algorithm allows both the client and the server to arrive at a shared encryption key which will be used henceforth to encrypt the entire communication session.

Here is how the algorithm works at a very basic level:
Both the client and the server agree on a very large prime number, which of course does not have any factor in common.

This prime number value is also known as the **seed value**.
Next, the two parties agree on a common encryption mechanism to generate another set of values by manipulating the seed values in a specific algorithmic manner. These mechanisms, also known as encryption generators, perform large operations on the seed. An example of such a generator is AES (Advanced Encryption Standard).
be authenticated.

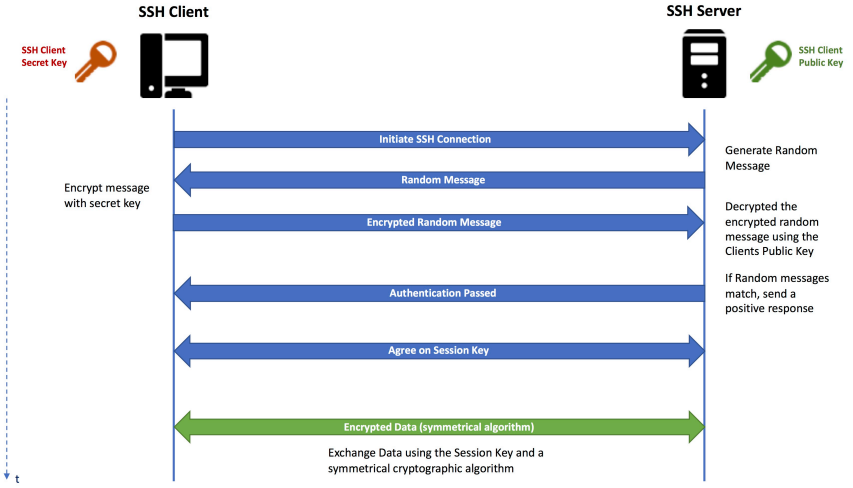


Remote connection

Both the parties independently generate another prime number. This is used as a secret private key for the interaction. This newly generated private key, with the shared number and encryption algorithm (e.g. AES), is used to compute a public key which is distributed to the other computer.

The parties then use their personal private key, the other machine's shared public key and the original prime number to create a final shared key. This key is independently computed by both computers but will create the same encryption key on both sides.

Now that both sides have a shared key, they can symmetrically encrypt the entire SSH session. The same key can be used to encrypt and decrypt messages (read: section on symmetrical encryption).
Now that the secured symmetrically encrypted session has been established, the user must



NAT – Network Address Translation

Basically, NAT allows a single device, such as a router, to act as an agent between the Internet (or public network) and a local network (or private network), which means that only a single unique IP address is required to represent an entire group of computers to anything outside their network.

Network Address Translation (NAT) is designed for IP address conservation. It enables private IP networks that use unregistered IP addresses to connect to the Internet. NAT operates on a router, usually connecting two networks together, and translates the private (not globally unique) addresses in the internal network into legal addresses, before packets are forwarded to another network.
As part of this capability, NAT can be configured to advertise only one address for the entire network to the outside world. This provides additional security by effectively hiding the entire internal network behind that address. NAT offers the dual functions of security and address conservation and is typically implemented in remote-access environments.