# FTP Security with Sockets

Dr. Kiramat ullah, Ayesha Salar, Muhammad Zunair, Muhammad Umar, Waleed Akhtar

**Abstract:** File Transfer Protocol (FTP), is a standard network protocol used to transfer files from one host to another host over a Transmission Control Protocol (TCP), based network such as internet. This protocol can be used to upload or download files form one host to another. This protocol is based on client-server architecture.

This article is about the security issues in FTP i.e., bounce attack, brute force, spoof attack and port stealing etc. Port Stealing is an issue in FTP that enables cyber criminals to steal data in transit. In this document we will be presenting a novel way to solve this problem by using Sockets and will secure information using encryption techniques to encrypt the data in transit.

***Key Words*— FTP (File Transfer Protocol), ARP (Address Resolution Protocol)**

## I. Introduction

**Protocol:** Protocols exist at several levels in a telecommunication connection. For example, there are protocols for the data interchange at the hardware device level and protocols for data interchange at the application program level. In the standard model known as Open Systems Interconnection (OSI), there are one or more protocols at each layer in the telecommunication exchange that both ends of the exchange must recognize and observe. Protocols are often described in an industry or international standard. The TCP Internet protocols is a common example of this.

**FTP Protocol:** FTP is a standard network protocol used for the transfer of computer files between a client and server on a computer network. It allows you to have ownership and access restrictions. It hides the details of individual computer system. FTP is a client-server protocol that relies on two communications channels between client and server: a command channel for controlling the conversation and a data channel for transmitting file content. Clients initiate conversations with servers by requesting to download a file. Using FTP, a client can upload, download, delete, and rename, move and copy files on a server. A user typically needs to log on to the FTP server, although some servers make some or all of their content available without login, also known as anonymous FTP. FTP sessions work in passive or active modes.

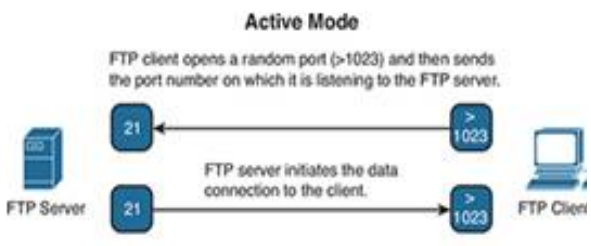


Figure 1 Active Mode

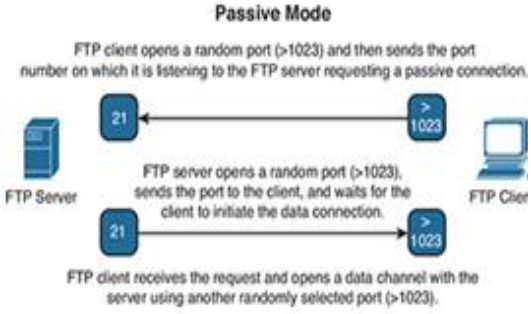


Figure 2 Passive Mode

In **active mode**, after a client initiates a session via a command channel request, the server initiates a data connection back to the client and begins transferring data. In **passive mode**, the server instead uses the command channel to send the client the information it needs to open a data channel. Because passive mode has the client initiating all connections, it works well across firewalls and Network Address Translation **(NAT)** gateways.
In next section we will discuss about the common issues that lie in FTP
.

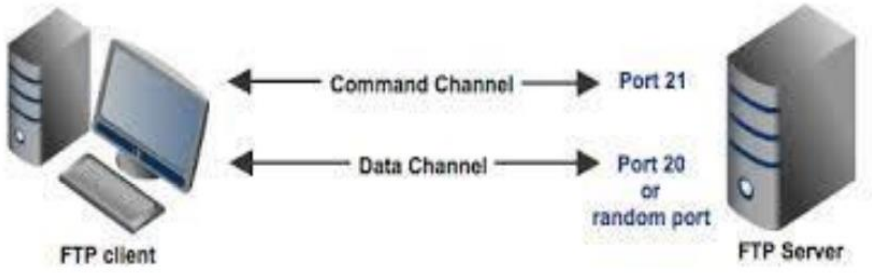


Figure 3 FTP Client-Server Communication

# II. Issues in FTP

A. **FTP bounce attack** is an exploit of the FTP protocol whereby an attacker is able to use the PORT command to request access to ports indirectly with the victim machine as an intermediary for the request.
B. **FTP brute force attack** is a way of cracking passwords by guessing. However, these "guesses", delivered one after another, are done very rapidly. They are spewed forth by hacking tools that reference a long list of possible passwords, often called a wordlist.
C. **Spoof Attack** is all about we restrict access to FTP servers based on the network address, it is possible that a cyber-criminal can use an external computer and assume the host address of a computer on the enterprise network, and download files during data transfer.
D. **Port Stealing** is all about when operating systems assign dynamic port numbers in a particular order or pattern, an attacker easily decodes the pattern and identify the next port number, which will be used. By illegally gaining access to a port number, the legitimate client trying to access the file will be denied and the hacker can steal files, or even insert a forged file or malicious file into the data stream, which will be accessed by other legitimate users in the organization.

In this article, we will be focusing on port stealing from the above ones. So next, we will be discussing about port stealing.

# III. Port Stealing Background

Many operating systems assign dynamic port numbers in increasing order. By making a legitimate transfer, an attacker can observe the current port number allocated by the server and "guess" the next one that will be used. The attacker can make a connection to this port, thus denying another legitimate client the ability to make a transfer. Alternatively, the attacker can steal a file meant for a legitimate user. In addition, an attacker can insert a forged file into a data stream thought to come from an authenticated

client. This problem can be mitigated by either making FTP clients and servers use random local port numbers for data connections, by requesting random ports from the operating system or using system dependent mechanisms.

# IV. Port Stealing

Port Stealing is a technique, which cyber criminals use to sniff in a switched environment when Address Resolution Protocol (ARP) [1] poisoning is not effective (for example, where static mapped ARPs are used). It floods the LAN with ARP packets. The destination Media Access Control (MAC) address of each "stealing" packet is the same as the attacker's one (other Network Interface Card (NIC) will not see these packets); the source MAC address will be one of the MACs of the victims. This process "steals" the switch's port of each victim. The attacker, winning the race condition with the real port owner, will receive using low delays, packets destined to "stolen" MAC addresses. When the attacker receives packets for "stolen" hosts, it stops the flooding process and performs an ARP request for the real destination of the packet. When the attacker receives the ARP, reply it is sure that the victim has "taken back" his port, so attacker can re-send the packet to the destination as is. The attacker will not keep the stolen port for himself; he will proceed in the following loop:

1. Steal the port,
2. Receive some data,
3. Give the port back,
4. Forward the data to the real destination,
5. Go back in step 1 by stealing the port again.

Let us say an attacker (evil0, behind switch port 1) wants to steal pc2 (the victim) port on the switch (port 2). SW1 has to be "tricked" into thinking that pc2 is behind the same switch port as evil0 (port1) To do that we evil0 has to send an Ethernet packet with bb:00:00:00:00:02 as source MAC address We say that evil0 has to "spoof" the victim's MAC address, or in other words to "forge an Ethernet packet with spoofed source MAC address" evil0 has to send "whatever" packet (ARP, raw Internet Protocol (IP), Internet Control Message (ICMP) [2], empty User Datagram Protocol (UDP)/TCP, Domain Name Service (DNS), etc..) with spoofed source MAC address and the switch will update the Forwarding Database (FDB) [3] properly.

---

1. Address Resolution Protocol poisoning (ARP poisoning) is a form of attack in which an attacker changes the Media Access Control (MAC) address and attacks an Ethernet LAN by changing the target computer's ARP cache with a forged ARP request and reply packets.
2. The Internet Control Message Protocol (**ICMP**) is a supporting protocol in the Internet protocol suite. Network devices, including routers, to send error messages and operational information

indicating, for example, that a requested service is not available or that a host or router could not be reached, use it.

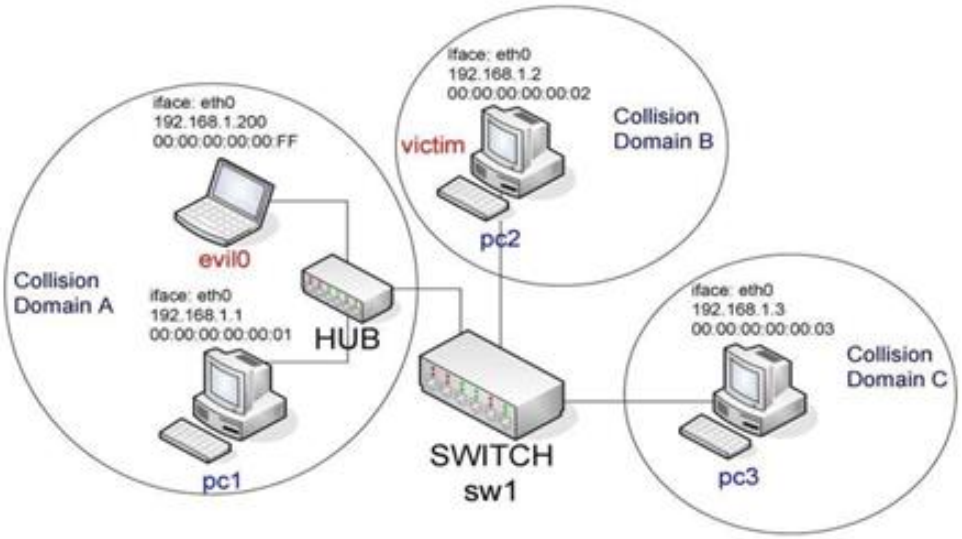


**Figure 4 Port Stealing Attack Scenario**

### A. How can Port Stealing be used as a MITM attack?

One can "steal" a Port by sending Ethernet Frames faking the victims source MAC-Address with the goal of confusing the Switch to a point where the CAM-Table (Content Addressable Memory Table) associates the attackers Port with the victims MAC Address that for real is behind another Port. This will result in the attacker receiving all the Packets, which are destined for the victim. However, at this point the attacker is not able to forward the packets to the victim because the switch still thinks the victim is behind the attacker's port.

### B. How to avoid Port Stealing:

We can avoid port stealing by making our FTP Protocol more reliable and more secure. We can secure our FTP Protocol by means of encryption and by making login password lengthy and making, the use of special symbols compulsory in the password.

# V. Sockets

Sockets allow communication between two different processes on the same or different machines. To be more precise, it is a way to talk to other computers using standard UNIX file descriptors. In UNIX, every I/O action is done by writing or reading a file descriptor. A file descriptor is just an integer associated with an open file and it can be a network connection, a text file, a terminal, or something else. To a programmer, a socket looks and behaves much like a low-level file descriptor. This is because commands such as read () and write () work with sockets in the same way they do with files and pipes.
Sockets were first introduced in 2.1BSD (Berkeley Software Distribution) and subsequently refined into their current form with 4.2BSD. The sockets feature is now available with most current UNIX system releases.

### A. Where is Socket Used?

A Unix Socket is used in a client-server application framework. A server is a process that performs some functions on request from a client. Most of the application-level protocols like

FTP, (Simple Mail Transfer Protocol (SMTP), and POP3 make use of sockets to establish connection between client and server and then for exchanging data.

**B. Socket Types:**

There are four types of sockets available to the users. The first two are most commonly used and the last two are rarely used. Processes are presumed to communicate only between sockets of the same type but there is no restriction that prevents communication between sockets of different types.

1) **Stream Sockets -** Delivery in a networked environment is guaranteed. If you send through the stream socket three items "A, B, C", they will arrive in the same order - "A, B, C". These sockets use TCP for data transmission. If delivery is impossible, the sender receives an error indicator. Data records do not have any boundaries.
2) **Datagram Sockets** - Delivery in a networked environment is not guaranteed. They are connectionless because you do not need to have an open connection as in Stream Sockets - you build a packet with the destination information and send it out. They use UDP.
3) **Raw Sockets -** These provide users access to the underlying communication protocols, which support socket abstractions. These sockets are normally datagram oriented, though their exact characteristics are dependent on the interface provided by the protocol. Raw sockets are not intended for the general user; they have been provided mainly for those interested in developing new communication protocols, or for gaining access to some of the more cryptic facilities of an existing protocol.
4) **Sequenced Packet Sockets** - They are similar to a stream socket, with the exception that record boundaries are preserved. This interface is provided only as a part of the Network Systems (NS) socket abstraction, and is very important in most serious NS applications. Sequenced-packet sockets allow the user to manipulate the Sequence Packet Protocol (SPP) or Internet Datagram Protocol (IDP) headers on a packet or a group of packets, either by writing a prototype header along with whatever data is to be sent, or by specifying a default header to be used with all outgoing data, and allows the user to receive the headers on incoming packets.

# VI. Solution

The best possible way is to use one of the above sockets schemes to secure data and have safe transit according to the requirements. We know this is not a generic solution but this is the best solution available in concerns with reliability, cost, speed and accuracy. However, we can do one thing and that is to provide multiple socket schemes in one protocol to make it more generic. Of course, we will not put all in one protocol because it just ruined the speed and accuracy factor. Rather than we will use different combos according to requirements. We can use this scheme to make is more generic that to divide this combos into different groups i.e., business, local, public etc.

# VII. Conclusion

This article was all about how to secure data transit-using FTP. We discussed about different issues in FTP and talked about port stealing. In addition, in port stealing we picked sockets for the solution. This is one of the best reliable methods available nowadays. For further work we have thoughts of exploring the data encryption cost and methods.

# VIII. References

## References


[1] Z. Haider, M. Saleem, and T. Jamal, "Analysis of Interference in Wireless", in Proc. of ArXiv, arXiv:1810.13164 [cs.NI], Oct. 2018.

[2] T. Jamal and P. Mendes, “Relay Selection Approaches for Wireless Cooperative Networks”, in Proc. of IEEE WiMob, Niagara Falls, Canada, Oct. 2010.

[3] T. Jamal, P. Mendes, and A. Zúquete, “Opportunistic Relay Selection for Wireless Cooperative Network”, in Proc. of IEEE IFIP NTMS, Istanbul Turkey, May 2012.

[4] T. Jamal and P. Mendes, “Cooperative relaying in user-centric networking under interference conditions”, in Proc. of IEEE Communications Magazine, vol. 52, no. 12, pp. 18–24, Dec 2014.

[5] P. Mendes, W. Moreira, T. Jamal, and Huiling Zhu, “Cooperative Networking in User-Centric Wireless Networks”, In: Aldini A., Bogliolo A. (eds) User-Centric Networking. Lecture Notes in Social Networks. Springer, Cham, ISBN 978-3-319- 05217-5, May 2014.

[6] T. Jamal, P. Mendes, and A. Zúquete, “Relayspot: A Framework for Opportunistic Cooperative Relaying”, in Proc. of IARIA ACCESS, Luxembourg, June 2011.

[7] T. Jamal, and SA Butt, "Malicious Node Analysis in MANETS", in Proc. of International Journal of Information Technology, PP. 1-9, Springer Publisher, Apr. 2018.

[8] T. Jamal, and P. Mendes, “COOPERATIVE RELAYING FOR DYNAMIC NETWORKS”, EU PATENT, (EP13182366.8), Aug. 2013.

[9] T. Jamal, and P. Mendes, “802.11 Medium Access Control In MiXiM”, in Proc. of Tech Rep. SITILabs-TR-13- 02, University Lusófona, Lisbon Portugal, Mar. 2013.

[10] T Jamal, M Alam, and MM Umair, "Detection and Prevention Against RTS Attacks in Wireless LANs", in Proc. of IEEE C-CODE, Islamabad Pakistan, Mar. 2017.

[11] L. Lopes, T. Jamal, and P. Mendes, "Towards Implementing Cooperative Relaying", In Proc. of Technical Report COPE-TR-13-06, CopeLabs University Lusofona Portugal, Jan 2013.

[12] T. Jamal, P. Mendes, and A. Zúquete, “Interference-Aware Opportunistic Relay Selection”, In Proc. of ACM CoNEXT student workshop, Tokyo, Japan, Dec. 2011.

[13] T. Jamal, "Cooperative MAC for Wireless Network", In Proc. of 1st MAP Tele Workshop, Porto, Portugal, 2010.

[14] T. Jamal and P. Mendes, “Analysis of Hybrid Relaying in Cooperative WLAN”, In Proc. of IEEE IFIP Wireless Days (WD), Valencia, Spain, November 2013.

[15] T. Jamal, and P. Mendes, “Cooperative Relaying for Wireless Local Area Networks”, In: Ganchev I., Curado M., Kassler A. (eds) Wireless Networking for Moving Objects. Lecture Notes in Computer Science, vol 8611. Springer, Cham, (WiNeMo), Aug. 2014.

[16] T. Jamal, and SA Butt, “Cooperative Cloudlet for Pervasive Networks”, in Proc. of Asia Pacific Journal of Multidisciplinary Research, Vol. 5, No. 3, PP. 42-26, Aug 2017.

[17] T. Jamal, P. Mendes, and A. Zúquete, “Wireless Cooperative Relaying Based on Opportunistic Relay Selection”, in Proc. of International Journal on Advances in Networks and Services, Vol. 5, No. 2, PP. 116-127, Jun. 2012.

[18] SA Butt, and T. Jamal, “Frequent Change Request from User to Handle Cost on Project in Agile Model”, in Proc. of Asia Pacific Journal of Multidisciplinary Research 5 (2), 26-42, 2017.

[19] T. Jamal, and P. Amaral, “Flow Table Congestion in Software Defined Networks”, in Proc. of IARIA 12th ICDS, Rome Italy, Mar. 2018.

[20] R. Sofia, P. Mendes, W. Moreira, A. Ribeiro, S. Queiroz, A. Junior, T. Jamal, N. Chama, and L. Carvalho, “Upns: User Provided Networks, technical report: LivingExamples, Challenges, Advantages”, Tech. Rep. SITI-TR- 11- 03, Research Unit in Informatics Systems and Technologies (SITI), University Lusofona, Lisbon Portugal, Mar. 2011.

[21] T. Jamal, and P. Mendes, “Cooperative Relaying in Wireless User-Centric Networks”, Book Chapter In: Aldini, A., Bogliolo, A. (eds.) User Centric Networking. Lecture Notes in Social Networks, Springer, Cham, pp. 171–195, 2014.

[22] T. Jamal, P. Mendes, and A. Zúquete, “Design and Performance of Wireless Cooperative Relaying”, PhD Thesis MAP-Tele, University of Aveiro, Oct. 2013.

[23] T. Jamal, P. Mendes, and A. Zuquete, “RelaySpot: Cooperative Wireless Relaying”, in Proc. of MAP-Tele Workshop, Aveiro, Portugal, May 2011.

[24] T. Jamal, and P. Mendes, “Cooperative Wireless Relaying, Key Factors for Relay Selection”, in Proc. of MAP- Tele Workshop, Porto, Portugal, Dec. 2009.

[25] SA Butt, and T. Jamal, “Study of Black Hole Attack in AODV”, in Proc. of International Journal of Future Generation Communication and Networking, Vol. 10, No.9, pp. 37-48, 2017.

[26] T. Jamal, and SA Butt, “Low-Energy Adaptive Clustering Hierarchy (LEACH) Enhancement for Military Security Operations”, In Proc. Of Journal of Basic and Applied Scientific Research, ISSN 2090-4304, 2017.

[27] T. Jamal, and P. Mendes, “RelaySpot, OMNET++ Module”, Software Simulator Extension In Proc. of COPE- SW-13-05, 2013.

[28] T. Jamal and Z. Haider, "Denial of Service Attack in Cooperative Networks", in Proc. of ArXiv, arXiv: CoRR Vol. arXiv:1810.11070 [cs.NI], Oct. 2018.