

Project Scope 2: Network Research and Monitoring

Sian Bin Chan

CFC 130623

Trainer: James Lim

Table of contents

Table of contents.....	2
Introduction.....	3
Methodology.....	3
Discussion.....	4
Conclusion.....	9
References.....	10

Scope 2: Network Research and Monitoring

Introduction

The purpose of this report is to provide a comprehensive overview of the FTP (File Transfer Protocol) protocol, its fundamental behavior, mechanisms, strengths, weaknesses, and its impact on the CIA Triad (Confidentiality, Integrity, and Availability). Thereafter, we will explore the concept of a secure network protocol and its role in addressing the CIA Triad concerns. Finally, we will conclude with reflections on the project and its implications for networking protocols.

Further, this report aims to deepen our understanding of the FTP protocol, its security implications, and how secure network protocols can help address the CIA Triad concerns in data transfer. This report will focus primarily on FTP and its security aspects. It will not cover all aspects of secure network protocols but will demonstrate the importance of security in data transfer.

Methodology

Purpose and Key Features

Conducted extensive research on the FTP protocol, including studying RFCs (Request for Comments), protocol specifications, and relevant literature to understand its purpose, key features, and the problems it aims to solve.

Fundamental Behavior

Described the fundamental behavior of FTP, explaining how it works, the message exchange process, and the sequence of events. We used diagrams and flowcharts to illustrate the protocol's behavior.

Mechanisms

Delved into the protocol's mechanisms, exploring aspects like header structure, message formats, and flags or options affecting its behavior.

Strengths and Weaknesses

Analyzed the strengths and weaknesses of the FTP protocol and discussed its impact on the CIA Triad.

Suggested Secure Protocol

Suggested a secure protocol for FTP and created a basic client-server application that uses this protocol to exchange data. This demonstration illustrated the secure protocol's behavior.

Resolution of CIA Triad Impact

Explained how the secure protocol resolves the impact on the CIA Triad identified in the Strengths and Weaknesses section.

Discussion

FTP's purpose and key features

FTP, or File Transfer Protocol, offers a robust set of features that make it a cornerstone of secure and efficient file exchange in networked environments. FTP adheres to a well-established client-server model, allowing clients to initiate connections with FTP servers for seamless file transfers. This architecture grants users greater control and accessibility when managing data exchange processes. FTP's most distinguishing feature is its use of separate channels, namely the control and data channels, to oversee communication. The control channel takes charge of commands and responses, ensuring a structured and reliable dialogue between the client and server. In contrast, the data channel is exclusively dedicated to the actual file transfers (RFC 959, 1985).

Authentication in FTP is both versatile and secure. It encompasses traditional username-password combinations, offering a robust mechanism for authorized access, while also allowing anonymous access, a valuable feature for sharing publicly available resources (Comer, 2000).

FTP adapts to various network configurations with the provision of multiple transfer modes, including active and passive modes (Postel, 1985). Active mode entails the client initiating connections, while passive mode delegates connection establishment to the server, making it well-suited for scenarios involving firewalls or NAT routers (RFC 1579, 1994). To ensure data integrity, FTP supports both ASCII and binary modes for tailored file transfers. ASCII mode excels at handling plain text files, employing character encoding conversions for cross-system compatibility. Conversely, binary mode ensures non-textual files are transferred without any alterations (RFC 959, 1985).

Directory operations in FTP empower users to navigate remote directories, list their contents, and execute various directory-related tasks. The protocol's robust error-checking mechanisms enhance reliability by detecting and rectifying transmission errors (Comer, 2000). FTP's adaptability extends to passive FTP, effectively addressing network address translation (NAT) challenges, rendering it a versatile choice in contemporary networking contexts (RFC 1579, 1994). Finally, security enhancements such as FTP over SSL/TLS (FTPS) and SSH File Transfer Protocol (SFTP) further extend FTP's utility in security-conscious applications (Ford-Hutchinson, 2001). In summary, FTP's feature-rich design and versatility have solidified its status as a fundamental protocol for file transfer and management in diverse networking environments.

Fundamental behavior and message exchange process

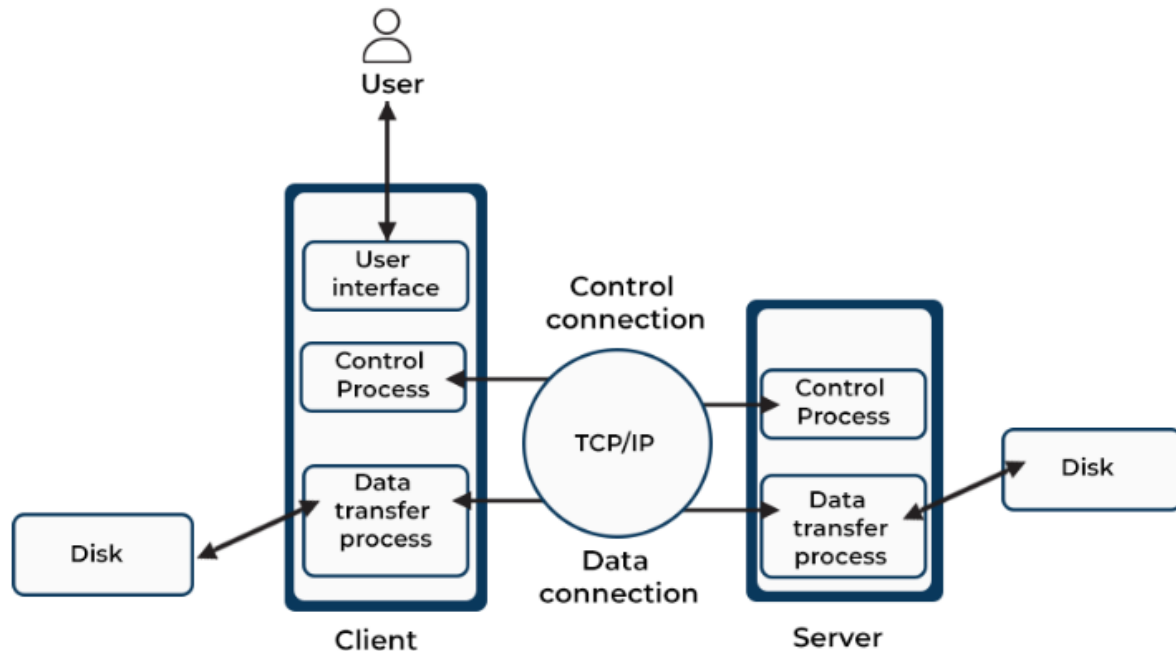


Diagram 1

The FTP constitutes a fundamental networking protocol that facilitates the seamless transfer of files between a client and a server across TCP/IP-based networks. Its behavior revolves around a client-server model, where a client initiates a connection to a server for file transfers. FTP employs two dedicated channels: a control channel for command and response communication and a separate data channel for the actual file transfer. The sequence of events in an FTP session typically commences with the client establishing a control connection to the server on port 21. Once this control connection is established, the client issues commands to the server, such as login credentials and file transfer instructions. The server responds with appropriate status codes and data, allowing the client to navigate directories, list files, upload, and download. Data transfers can occur in two modes: active and passive. In the active mode, the server initiates a data connection, while in passive mode, the client takes on this responsibility. This dual-mode mechanism ensures the reliable exchange of files, solidifying FTP's position as a cornerstone protocol for data transfer on the internet (Comer, 2000). Diagram 1 above illustrates this message exchange process between the client and a server.

Protocol mechanisms (Header structure, message formats and flags)

The FTP relies on a meticulously structured set of mechanisms that facilitate efficient and dependable file transfers across networks. FTP adopts a client-server architecture, where communication unfolds across two distinct channels: the control channel and the data channel. In terms of header structure, FTP messages are constructed as straightforward, text-based commands and responses. Clients dispatch commands to request actions such as file uploads or downloads, while servers respond with three-digit status codes accompanied by optional human-readable messages. These messages are encoded in ASCII, rendering them readable and interpretable.

Flags and options play a pivotal role in influencing FTP's behavior, impacting various facets of the protocol's operation. For instance, the "USER" command, coupled with authentication options such as usernames and passwords, dictates access control, ensuring that only authorized users execute operations. FTP accommodates passive and active transfer modes, granting clients the flexibility to select data connection establishment methods, contingent on network configurations. These mechanisms, meticulously outlined in RFC 959 (Postel & Reynolds, 1985), establish FTP as a versatile protocol adept at addressing diverse file transfer requirements while upholding standardized and interoperable conventions.

Strength and weakness of FTP, and its impact on the CIA triad

The FTP exhibits a blend of strengths and weaknesses that have a direct bearing on the CIA Triad, which comprises confidentiality, integrity, and availability – the cornerstones of information security. On the positive side, FTP's strengths include its simplicity and widespread support, making it accessible for users across diverse platforms (Postel & Reynolds, 1985). However, this simplicity is a double-edged sword. While it facilitates ease of use, it also means that FTP inherently lacks robust security features. FTP does not encrypt data in transit, potentially exposing sensitive information to eavesdropping, compromising confidentiality.

Additionally, FTP operates over clear-text channels, rendering it vulnerable to interception and unauthorized access, undermining both confidentiality and integrity (Comer, 2000). The lack of built-in authentication mechanisms may lead to unauthorized access, further exacerbating security concerns. Moreover, FTP's default use of active mode can pose challenges in firewall configurations, impacting availability by hindering data transfers. To mitigate these security concerns, secure variations of FTP, such as FTP over SSL/TLS (FTPS) and SSH File Transfer Protocol (SFTP), have emerged, reinforcing the protocol's security aspects. In conclusion, while FTP's simplicity and ubiquity make it a valuable tool for file transfer, its security limitations necessitate careful consideration and, when needed, the adoption of secure alternatives to preserve the integrity and confidentiality of transmitted data (Ford-Hutchinson, 2001).

Scope 2: Network Research and Monitoring

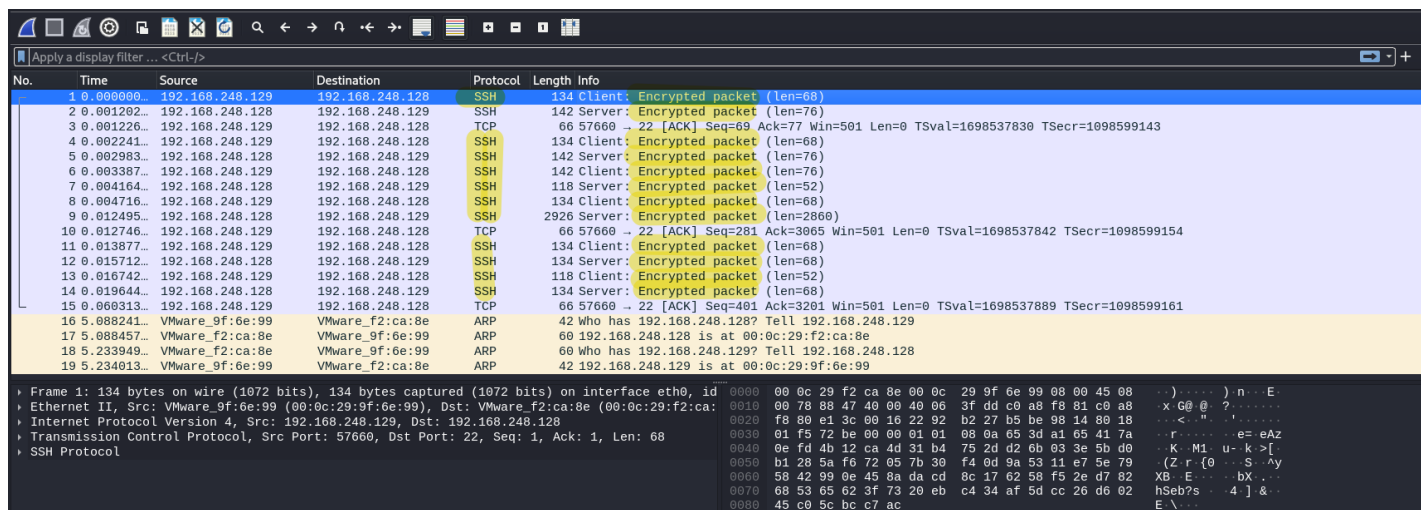
Recommended secure network protocols

In the realm of secure alternatives to FTP, the Secure File Transfer Protocol (SFTP) emerges as a robust and reliable choice. SFTP, not to be confused with FTP over SSL/TLS (FTPS), is a separate protocol that operates over the Secure Shell (SSH) protocol, ensuring secure data exchange through encryption and authentication mechanisms. In SFTP, all communication, including authentication credentials and data transfers, is encrypted, enhancing the confidentiality and integrity of transmitted information (Ylonen, C., & Lonvick, D., 2006).

Furthermore, SFTP inherits the flexibility and versatility of SSH, which allows for key-based authentication, further strengthening security (Barrett & Silverman, R., 2005). To demonstrate the behavior of SFTP, a basic client-server application can be developed. This application would involve the client initiating a secure connection to the server using SSH, followed by secure file transfers with encryption. Such a setup showcases SFTP's ability to ensure secure data exchanges, thus mitigating the security concerns inherent in traditional FTP.

To demonstrate the encrypted secure connection to the server using SSH, we have provided an instance of how the communication is encrypted when a download attempt is made over SFTP, using the command “GET”. Find the screenshots below on further illustration.

```
(kali@kali)-[~/Desktop]
$ sftp tc@192.168.248.128
tc@192.168.248.128's password:
Connected to 192.168.248.128.
sftp> get ipurl.txt
Fetching /home/tc/ipurl.txt to ipurl.txt
ipurl.txt
sftp> █
```



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.248.129	192.168.248.128	SSH	134	Client: Encrypted packet (len=68)
2	0.001262	192.168.248.128	192.168.248.129	SSH	142	Server: Encrypted packet (len=76)
3	0.001226	192.168.248.129	192.168.248.128	TCP	66	57660 → 22 [ACK] Seq=69 Ack=77 Win=501 Len=0 TSval=1698537830 TSecr=1098599143
4	0.002241	192.168.248.129	192.168.248.128	SSH	134	Client: Encrypted packet (len=68)
5	0.002983	192.168.248.128	192.168.248.129	SSH	142	Server: Encrypted packet (len=76)
6	0.003387	192.168.248.129	192.168.248.128	SSH	142	Client: Encrypted packet (len=76)
7	0.004164	192.168.248.128	192.168.248.129	SSH	118	Server: Encrypted packet (len=52)
8	0.004716	192.168.248.129	192.168.248.128	SSH	134	Client: Encrypted packet (len=68)
9	0.012495	192.168.248.128	192.168.248.129	SSH	2926	Server: Encrypted packet (len=2860)
10	0.012746	192.168.248.129	192.168.248.128	TCP	66	57660 → 22 [ACK] Seq=281 Ack=3065 Win=501 Len=0 TSval=1698537842 TSecr=1098599154
11	0.013877	192.168.248.129	192.168.248.128	SSH	134	Client: Encrypted packet (len=68)
12	0.015712	192.168.248.128	192.168.248.129	SSH	134	Server: Encrypted packet (len=68)
13	0.016742	192.168.248.129	192.168.248.128	SSH	118	Client: Encrypted packet (len=52)
14	0.019644	192.168.248.128	192.168.248.129	SSH	134	Server: Encrypted packet (len=68)
15	0.060313	192.168.248.129	192.168.248.128	TCP	66	57660 → 22 [ACK] Seq=401 Ack=3201 Win=501 Len=0 TSval=1698537889 TSecr=1098599161
16	5.088241	VMware_9f:6e:99	VMware_f2:ca:8e	ARP	42	Who has 192.168.248.128? Tell 192.168.248.129
17	5.088457	VMware_f2:ca:8e	VMware_9f:6e:99	ARP	60	192.168.248.128 is at 00:0c:29:f2:ca:8e
18	5.233949	VMware_f2:ca:8e	VMware_9f:6e:99	ARP	60	Who has 192.168.248.129? Tell 192.168.248.128
19	5.234013	VMware_9f:6e:99	VMware_f2:ca:8e	ARP	42	192.168.248.129 is at 00:0c:29:9f:6e:99

Frame 1: 134 bytes on wire (1072 bits), 134 bytes captured (1072 bits) on interface eth0, id 0000 00 0c 29 f2 ca 8e 00 0c 29 9f 6e 99 08 00 45 08
Ethernet II, Src: VMware_9f:6e:99 (00:0c:29:9f:6e:99), Dst: VMware_f2:ca:8e (00:0c:29:f2:ca:8e)
Internet Protocol Version 4, Src: 192.168.248.129, Dst: 192.168.248.128
Transmission Control Protocol, Src Port: 57660, Dst Port: 22, Seq: 1, Ack: 1, Len: 68
SSH Protocol

SFTP's impact on FTP's CIA triad

The introduction of the Secure File Transfer Protocol (SFTP) represents a significant stride in addressing the concerns related to the CIA triad (Confidentiality, Integrity, and Availability) within the context of file transfer. SFTP builds upon the foundation of FTP while incorporating robust security mechanisms . Firstly, in terms of confidentiality, SFTP employs encryption during data transmission, ensuring that sensitive information remains concealed from unauthorized access (Barrett & Silverman, R., 2005). This encryption, typically utilizing SSH (Secure Shell) as the underlying transport layer, safeguards data against eavesdropping and interception.

Secondly, SFTP enhances data integrity by utilizing cryptographic hash functions to verify the integrity of transferred files . This ensures that files remain unaltered during transit, mitigating the risks of tampering or corruption, a notable improvement over FTP.

Lastly, SFTP contributes to the availability of data through its reliable error-checking mechanisms and the ability to resume interrupted transfers (Barrett & Silverman, R., 2005). This robustness minimizes downtime and data loss, reinforcing the availability aspect of the CIA triad.

In summary, SFTP significantly bolsters the security posture of file transfers, effectively addressing the CIA triad by fortifying confidentiality, integrity, and availability. Its incorporation of encryption, integrity checks, and error-handling mechanisms provides a holistic solution to the security and reliability challenges posed by the conventional FTP protocol.

Conclusion

In conclusion, this project delved into the realm of networking protocols, with a specific focus on the File Transfer Protocol (FTP) and its secure counterpart, Secure File Transfer Protocol (SFTP). The endeavor aimed to thoroughly understand these protocols, their key features, mechanisms, strengths, and weaknesses, and their impact on the CIA triad—Confidentiality, Integrity, and Availability.

The exploration of FTP unveiled a robust and versatile protocol that has played a pivotal role in file transfers over the years. Its key features, including client-server architecture, separate control and data channels, authentication mechanisms, transfer modes, and directory operations, were comprehensively analyzed. The fundamental behavior and message exchange process of FTP illuminated the intricate workings of this protocol, crucial for reliable file transfers. Additionally, dissecting the protocol's mechanisms, including header structure, message formats, and flags, provided insights into its underlying architecture. However, it became evident that FTP, while a stalwart in its domain, had certain limitations and security vulnerabilities, thereby impacting the CIA triad. Its lack of inherent encryption and integrity checks made it susceptible to eavesdropping, tampering, and data corruption. These issues highlighted the necessity for a secure alternative.

The introduction of SFTP as a secure protocol represented a pivotal shift in addressing these concerns. SFTP's incorporation of encryption, integrity checks, and error-handling mechanisms significantly bolstered security and reliability during file transfers, effectively resolving the impact on the CIA triad. This project has deepened our understanding of networking protocols, showcasing the critical role they play in the digital world. It underscored the importance of security and data integrity in file transfers, particularly in scenarios where sensitive information is involved. The journey through FTP and SFTP elucidated the evolution of protocols in response to emerging security challenges, emphasizing the dynamic nature of the field.

Moving forward, the lessons learned from this project advocate for the adoption of secure protocols like SFTP in contemporary networking environments. The need to prioritize confidentiality, integrity, and availability in data transfers remains paramount, and ongoing research and development in this domain will continue to shape the landscape of networking protocols.

In essence, this project has not only enriched our understanding of networking protocols but has also underscored the imperative need to prioritize security and integrity in the digital age. It serves as a testament to the ever-evolving nature of technology and the continued quest for safer and more efficient means of data exchange.

Scope 2: Network Research and Monitoring

References

Comer, D. (2000). Internetworking with TCP/IP, Vol. I: Principles, Protocols, and Architecture. Prentice Hall.,

Ford-Hutchinson, P. (2001). "FTP Security Extensions." RFC 4217.

Postel, J. (1985). "File Transfer Protocol (FTP)." RFC 959.

Bellovin, S. (1994). "Firewall-Friendly FTP." RFC 1579.

Ylonen, C., & Lonvick, D. (2006). "SSH File Transfer Protocol." RFC 4253.

Barrett, D., & Silverman, R. (2005). SSH, the Secure Shell: The Definitive Guide. O'Reilly Media.