Monitor Website Traffic with Wireshark

Atal Pandey[a], *Suyog Joshi*[a]

[a]*Department of Computer Science, Student of Computer Science, University of New Mexico, 87106, New Mexico, United States*

KEYWORDS

Wireshark
Online Payments
Nepal

ABSTRACT

This project advocates for an exhaustive examination of web applications using Wireshark as a primary tool for analyzing network packets. Through a thorough examination of network traffic, the goal is to look under the hood to uncover valuable insights related to data transmission, security protocols, and overall performance. The dynamic of online payments has gained pace in Nepal over the last decade. However, from the network perspective no research has been conducted so far to provide an insight on the web traffic of those online payment systems. Through this research, we intend to gain some insights into the internal workings from a network perspective of the systems under consideration.

1 Introduction

This project advocates for an exhaustive examination of web applications using Wireshark as a primary tool for analyzing network packets. Through a thorough examination of network traffic, the goal is to look under the hood to uncover valuable insights related to data transmission, security protocols, and overall performance. The dynamic of online payments has gained pace in Nepal over the last decade. However, from the network perspective no research has been conducted so far to provide an insight on the web traffic of those online payment systems. Through this research, we intend to gain some insights into the internal workings from a network perspective of the systems under consideration.

1.1 Motivation

In the modern age, understanding how applications and webservers exchange data, that is, the data flow of applications and websites is a crucial factor for network administrators, web security professionals, organizations, and casual users alike. This is because of several reasons. Firstly, knowing where data is flowing helps identify vulnerabilities so that we can better protect our data and privacy. The rise of the internet has been meteoric and the threats over the internet are as ubiquitous as the internet itself. Monitoring how your data travels over different devices and networks as it tries to reach its destination helps recognize, reveal, and intercept malicious activity in time to keep data integrity and privacy in check. Secondly, gaining an insight into data flow helps better utilize the network resources to achieve optimum performance. The resources over the internet are limited and several anomalies can arise that can lead to bandwidth congestion and ultimately network failure. Malware (or botnet) activity, DDoS attacks, inadequate network infrastructure, network misconfiguration, and network throttling are some anomalies that we want to avoid. Obviously, looking into data flows helps reveal such anomalies and prevent them. Thirdly, troubleshooting process can be expedited when one has an insight on the data flow. Instead of skimming over issues by following the trial-and-error method to solve network issues, one can better identify issues and reach a feasible solution faster and hence more efficiently when one has an insight into the data flow. Lastly, monitoring data flows can help verify that third-party applications and services are adhering to their service-level agreements and security standards. Social media platforms, payment processors, cloud storage and file sharing services are some third-party applications and services that we use on a daily basis. Monitoring data flows of such applications and services to understand how a user's data is being utilized is crucial to keep data privacy and security in check.

1.2 Contribution

The infrastructure for payment over web applications has gradually evolved in Nepal over the last decade. However, the infrastructure only gained popularity during and after the Covid-19 pandemic. The web applications under consideration are now widely used in Nepal. However, minimal research has been conducted to see how data is flowing over those web applications. Through this research, we intend to gain some insights into the internal workings from a network perspective of the applications under consideration and present a detailed overview of the protocols and safety standards used and provide guidelines to further secure the web traffic for those web applications if needed. We also intend our work to be a source of reference for any subsequent work that might follow.

1.3 Objectives

The primary objectives of this project are:

- Examine Data Transmission: Analyze the patterns and characteristics of data transmission within web applications for identifying potential bottlenecks.
- Evaluate Security Protocols: Investigate the security protocols used by the web applications, identifying any encryption methods, and any potential vulnerabilities.
- Evaluate the ports used: Access and identify the ports utilized in the data transmission process. This involves determining the ports used and why they are used in the process.
- Analyze domains and subdomains: Investigate the domains and subdomains associated with the web application. The aim is to identify the primary web addresses, subcategories, and any other related domains, providing a comprehensive overview of the web application's structural hierarchy.

1.4 Scope

This project focuses on a limited set of web applications, chosen based on online payments done in Nepal. We have chosen both private and government web applications for this project. The analysis will encompass both client-server and server-server communication, providing a holistic view of the network traffic associated with web applications. The web applications chosen for this project are:

- esewa.com.np
- connnectips.com
- khalti.com
- ipay.com.np
- imepay.com.np

1.5 Significance

Understanding how the web applications use internet is very important. It helps make things more secure, faster, and gives users a smoother experience. The findings of this project may offer valuable insights to developers, tech admins, and security professionals, contributing to the ongoing improvement of web application development. Also, since the research data on our topic is very minimal in Nepal, our research will also help serve as a reference to any future work related to network analysis and diagnosis of the web applications under consideration.

2 Web Applications

2.1 Web Application Analysis

Web application analysis involves the examination of network traffic for gaining insights into the functioning and behavior of applications accessed through the internet. Understanding data transmission, security protocols, and performance is crucial for optimizing web applications and ensuring a secure user experience.

2.2 Wireshark Overview

Wireshark is a powerful widely used open-source packet analyzer allowing real-time inspection of data traversing a network. This tool boasts features like packet capturing in real-time, making it beneficial for network administrators, developers and users interested in internet security. With its support for diverse protocols, Wireshark proves versatile in inspecting various layers of computer networks. In this project, Wireshark is used as a tool for packet capturing and looking at the various protocols, ports used and tracking data. Understanding the traffic patterns helps identify anomalies and optimize data transmission efficiency. We also used it for examination of the security protocols such as HTTP, TLS aiding in the identification of vulnerabilities and adherence to best practices. It is also used for evaluating performance metrics like latency, response time, and throughput, contributing to the enhancement of web application performance.

2.3 Relevant Technologies and Tools

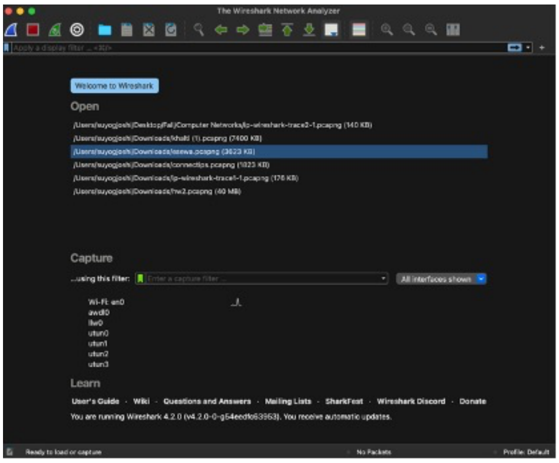
The integration of Wireshark with other analysis tools and methodologies is essential for comprehensive web application analysis. We used other tools as well for this project like browser developer tools, shodan.io, seranking.com, nslookup(in terminal) to help us analyze any vulnerabilities, finding subdomains and real-time monitoring network activity and understand client-server behavior.

3 Methodology

The approach proposed involves the usage of Wireshark in information gathering to analyze application layer protocols used by the most popular money transfer and payment web applications in Nepal to look for any possible vulnerabilities. The web applications we have considered are: connectips.com, esewa.com.np, khalti.com, ipay.com.np, and imepay.com.np. Our approach includes the following steps.

3.1 *Environment Setup*

The initial step is setting up the environment to ensure that we have the necessary permissions and tools for capturing network traffic. In this context, our environment includes several web applications. To replicate real-world scenarios, we've mimicked the actual usage of these websites and done real-world transactions for capturing traffic through Wireshark and additional tools. As a requirement, we have involved an active subscription to a mobile operator in Nepal and used preexisting accounts. The primary tool chosen for analyzing the packets is Wireshark. Also, other helpful tools like shodan.io, web browser tools have been utilized.



```
/Users/suyogjoshi/Desktop/Fall/Computer Networks/ip-wireshark-trace2-1.pcapng (140 KB)
/Users/suyogjoshi/Downloads/khalti (1).pcapng (7400 KB)
/Users/suyogjoshi/Downloads/esewa.pcapng (3623 KB)
/Users/suyogjoshi/Downloads/connectips.pcapng (1023 KB)
/Users/suyogjoshi/Downloads/ip-wireshark-trace1-1.pcapng (176 KB)
/Users/suyogjoshi/Downloads/hw2.pcapng (40 MB)
```

You are running Wireshark 4.2.0 (v4.2.0-0-g54eedfc63953). You receive automatic updates.

3.2 *Packet Capture and Filtering*

After the environment setup, we activate Wireshark to capture real-time data as it flows through the network connected to our web applications under consideration. The steps involve monitoring the actual data exchanges between the website and users and providing a dynamic view of network activity. Filters will be applied to focus on relevant traffic, and capturing process will be initiated during the user interactions with the applications.

3.3 *Login into Websites Under Consideration*

Interact with the websites by logging into the websites and simulating user's action. This step aims to observe the network traffic patterns especially associated with user authentication and website access.

3.4 *Analyze the protocols and ports used*

Using Wireshark to examine the protocols, ports involved in the communication between the user's device and the websites. This involves looking at the details of how data is exchanged during website interactions and examine the patterns of data transmission, identifying the protocols used, packet sizes, and transmission times. The analysis aims to uncover inefficiencies and optimize data transfer for improved performance.

4 Expected Results

We expect to uncover a variety of insights related to web applications, data transmission, and security protocols. Some expected findings and results are as follows.

- □ Identify the data being transmitted between client and server.
- □ Understand patterns of data transfer.
- □ Evaluate the usage of protocols.
- □ Identify any vulnerabilities and potential security risks in the transmission of sensitive user information.
- □ Measure and analyze response times for different requests and identifying any bottlenecks.
- □ Detect any unusual network behaviors that may indicate security threats or technical issues.

By successfully executing this approach, we may be able to contribute to the overall improvement of the web application's security, reliability, and performance.

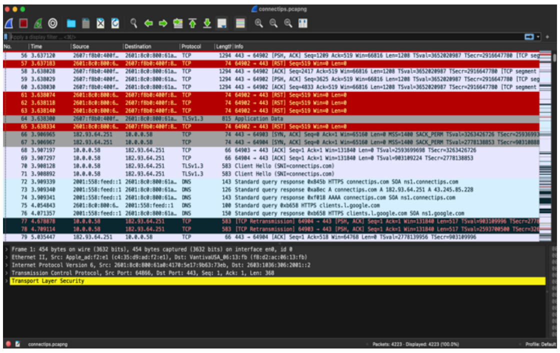
5 Implementation and Findings

After initiating the packet capture process during interaction with the selected web applications, we ensure that the capture aligns with typical user sessions. A typical user session involves logging in to the system, making transactions and logging out. Using Wireshark's filtering capabilities to focus on relevant communication, including packets, protocols, and ports in use along with associated domains and subdomains, the findings observed in a typical user session for our web applications under consideration have been described below.

5.1 *connectips.com*

The IP address 182.93.64.251 and 43.245.85.228 are associated with the domain "connectips.com". During the packet capture lines, packet 66,67, and 68 represents a TCP three-way handshake for a connection from source (10.0.0.58) to destination (182.93.64.251), and similar handshake can be observed for other connections, in lines 435 and 829. Packets with "PSH, ACK" indicates the transmission of the actual data (packet 111, 129, 147). The "Len" field in these lines shows the length of the TCP payload. Packets 1819, 1834, and 1835 indicates the initiation of the connection termination with "FIN, ACK" flags. Packets 64, 434, 853, 857, 1823, 1824 shows TCP reset packets. Also, there are additional connection attempts, like 47, 49, 398, 3717, etc. The protocols used are HTTP/HTTPS. Since, the transaction is over HTTPS, there is an SSL/TLS handshake at the beginning of the connection. The ports used is 443. It is used for secure web traffic for HTTPS. HTTPS is a combination of HTTP and the SSL/TLS protocol. It provides a secure and encrypted communication channel between web server and the user's browser. This encryption helps to protect sensitive information, such as login credentials, personal data, and financial transactions, from eavesdropping and unauthorized access. The domain and subdomain used with their used traffic are shown in the table below.

Domain and Sub domain	Traffic Share
connectips.com	59.58%
login.connectips.com	39.16%
corporatepay.connectips.com	0.55%
uat.connectips.com	0.54%
www.connectips.com	0.17%

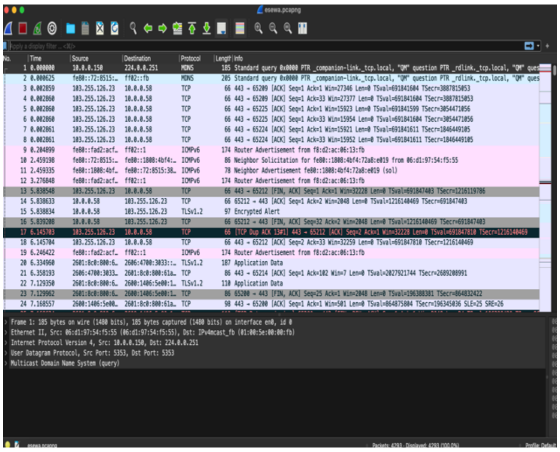


```
No.  Time      Source             Destination        Protocol Length Info
  56 3.637120  2607:f8b0:400f…    2601:8c0:800:61a…  TCP      1294 443 → 64902 [PSH, ACK] Seq=1209 Ack=519 Win=66816 Len=1208 TSval=3652020987 TSecr=2916647780 [TCP seg
  57 3.637183  2601:8c0:800:6…    2607:f8b0:400f:8…  TCP        74 64902 → 443 [RST] Seq=519 Win=0 Len=0
  58 3.638028  2607:f8b0:400f…    2601:8c0:800:61a…  TCP      1294 443 → 64902 [ACK] Seq=2417 Ack=519 Win=66816 Len=1208 TSval=3652020987 TSecr=2916647780 [TCP segment
  59 3.638029  2607:f8b0:400f…    2601:8c0:800:61a…  TCP      1294 443 → 64902 [PSH, ACK] Seq=3625 Ack=519 Win=66816 Len=1208 TSval=3652020987 TSecr=2916647780 [TCP seg
  60 3.638030  2607:f8b0:400f…    2601:8c0:800:61a…  TCP      1294 443 → 64902 [ACK] Seq=4833 Ack=519 Win=66816 Len=1208 TSval=3652020987 TSecr=2916647780 [TCP segment
  61 3.638074  2601:8c0:800:6…    2607:f8b0:400f:8…  TCP        74 64902 → 443 [RST] Seq=519 Win=0 Len=0
  62 3.638118  2601:8c0:800:6…    2607:f8b0:400f:8…  TCP        74 64902 → 443 [RST] Seq=519 Win=0 Len=0
  63 3.638140  2601:8c0:800:6…    2607:f8b0:400f:8…  TCP        74 64902 → 443 [RST] Seq=519 Win=0 Len=0
  64 3.638300  2607:f8b0:400f…    2601:8c0:800:61a…  TLSv1.3   815 Application Data
  65 3.638334  2601:8c0:800:6…    2607:f8b0:400f:8…  TCP        74 64902 → 443 [RST] Seq=519 Win=0 Len=0
  66 3.906965  182.93.64.251      10.0.0.58          TCP        74 443 → 64903 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1400 SACK_PERM TSval=3263426726 TSecr=25936993
  67 3.906967  182.93.64.251      10.0.0.58          TCP        74 443 → 64904 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1400 SACK_PERM TSval=2778138853 TSecr=90310888
  68 3.907197  10.0.0.58          182.93.64.251      TCP        66 64903 → 443 [ACK] Seq=1 Ack=1 Win=131840 Len=0 TSval=2593699698 TSecr=3263426726
  69 3.907297  10.0.0.58          182.93.64.251      TCP        66 64904 → 443 [ACK] Seq=1 Ack=1 Win=131840 Len=0 TSval=903109224 TSecr=2778138853
  70 3.908128  10.0.0.58          182.93.64.251      TLSv1.3   583 Client Hello (SNI=connectips.com)
  71 3.908892  10.0.0.58          182.93.64.251      TLSv1.3   583 Client Hello (SNI=connectips.com)
  72 3.909339  2001:558:feed::1  2601:8c0:800:61a…  DNS       143 Standard query response 0x845b HTTPS connectips.com SOA ns1.connectips.com
  73 3.909340  2001:558:feed::1  2601:8c0:800:61a…  DNS       126 Standard query response 0xa8ec A connectips.com A 182.93.64.251 A 43.245.85.228
  74 3.909341  2001:558:feed::1  2601:8c0:800:61a…  DNS       143 Standard query response 0xf018 AAAA connectips.com SOA ns1.connectips.com
  75 4.054843  2601:8c0:800:6…   2001:558:feed::1   DNS       100 Standard query 0xb658 HTTPS clients.l.google.com
  76 4.071357  2001:558:feed::1  2601:8c0:800:61a…  DNS       150 Standard query response 0xb658 HTTPS clients.l.google.com SOA ns1.google.com
  77 4.678878  10.0.0.58          182.93.64.251      TCP       583 [TCP Retransmission] 64904 → 443 [PSH, ACK] Seq=1 Ack=1 Win=131840 Len=517 TSval=903109996 TSecr=2778
  78 4.709114  10.0.0.58          182.93.64.251      TCP       583 [TCP Retransmission] 64903 → 443 [PSH, ACK] Seq=1 Ack=1 Win=131840 Len=517 TSval=2593700500 TSecr=326
  79 5.035447  182.93.64.251      10.0.0.58          TCP        66 443 → 64904 [ACK] Seq=1 Ack=518 Win=64768 Len=0 TSval=2778139956 TSecr=903109996

> Frame 1: 454 bytes on wire (3632 bits), 454 bytes captured (3632 bits) on interface en0, id 0
> Ethernet II, Src: Apple_ad:f2:e1 (c4:35:d9:ad:f2:e1), Dst: VantivaUSA_06:13:fb (f8:d2:ac:06:13:fb)
> Internet Protocol Version 6, Src: 2601:8c0:800:61a0:4170:5e17:9b63:73eb, Dst: 2603:1036:306:2001::2
> Transmission Control Protocol, Src Port: 64866, Dst Port: 443, Seq: 1, Ack: 1, Len: 368
> Transport Layer Security

connectips.pcapng          Packets: 4223 · Displayed: 4223 (100.0%)          Profile: Default
```

5.2 esewa.com.np

The IP address 103.255.126.24, 103.255.126.46, 103.255.126.22, 103.255.126.21, 103.255.126.23, 103.255.126.20 are associated with the domain "esewa.com.np". Each of these addresses represent a server or a network interface that can be used to access the services provided by "esewa.com.np". It has multiple IP addresses associated with it and indicates a distributed network setup. This can be load balancing, for redundancy or serving users from geographically distributed data centers to reduce latency. When we access the domain through a network service, the request is directed to any of these IP addresses and the decision is made by the DNS resolver, often based on factors like network congestion, distance, or server availability. The packets with "Client Hello" indicate the initiation of a TLS handshake by the client (10.0.0.58) to the server (103.255.126.23). Following the client hello, the sever responds with "Server Hello" packets, indicating that it has accepted the client's request and is ready to establish a secure connection. These Server Hello packets include details about the agreed-upon cryptographic parameters for secure connection. The capture shows multiple instances of the TLS handshake process, with the client initiating the handshake (Client Hello), and the server responding (Server Hello). There is a repetitive nature of the packets that indicates either a repeated attempt to establish a connection or an ongoing communication session. Towards the end of the data, there is a "Client Hello" packet directed at a different IP address (103.65.200.21). This is an attempt to establish a secure connection with a different server. The packets are using the TLSv1.2 protocol version, as the data indicates. Packets 3-8 shows a TCP connection between the source and the destination. Packets 13-16 indicates the termination of a TCP connection, where the source sends a FIN, ACK signal, and the destination acknowledges it. Packets 17-18 show a TCP Dup ACK, indicating a duplicate acknowledgement and this often occurs in response to a retransmitted packet. Packets 683-692 shows the initiation of a new TCP connection from 10.0.0.58 to 103.255.126.23 over port 443. Packets 687-691 shows a TLS handshake with a Client Hello, Server Hello, and Encrypted Handshake Message. Packets 692-699 demonstrate the exchange of Application Data in the TLS-encrypted connection. Packets 695,696, 698 shows retransmitted TCP segments. Packets 715-716 shows an ACK for the FIN, ACK signal and a Syn retransmission for a new connection. Packets 1925-2038 shows the TLS handshake and the termination of a connection. The ports used for communication are 443, 993 and 995. Port 443 is used for secure web traffic for HTTPS. HTTPS is a combination of HTTP and the SSL/TLS protocol. It provides a secure and encrypted communication channel between web server and the user's browser. This encryption helps to protect sensitive information, such as login credentials, personal data, and financial transactions, from eavesdropping and unauthorized access. Port 993 is used for secure email configuration over IMAP protocol. IMAP allows email clients to retrieve message from a mail server, and when it operates over TLS/SSL (encrypted), it uses port 993 to ensure the confidentiality of email data during transmission. Port 995 is used for secure email retrieval using the POP3 protocol. It is similar to IMAPS, as it operates over TLS/SSL to provide encryption for the communication between an email client and the mail server. It ensures that sensitive email content, including usernames and passwords, is transmitted securely. In short, port 443 is associated with secure web browsing (HTTPS), port 993 is associated with secure email retrieval using IMAP(IMAPS), port 995 is associated with secure email retrieval using POP3(POP3S). The domain and subdomain used with their used traffic are shown in the table below.

Domains and Sub domain	Traffic Share
esewa.com.np	29.93%
blog.esewa.com.np	67.16%
merchant.esewa.com.np	2.63%
helpdesk.esewa.com.np	0.15%
uat-merchant.esewa.com.np	0.05%
developer.esewa.com.np	0.04%
nicnepal.esewa.com.np	0.02%
edolpa.esewa.com.np	0%
edn.esewa.com.np	0%
gbics.esewa.com.np	0%


esewa.pcapng
Apply a display filter ... <⌘/>
esewa.pcapng
Packets: 4293 · Displayed: 4293 (100.0%)
Profile: Default

```
No.  Time      Source            Destination        Protocol Length Info
  1  0.000000  10.0.0.150        224.0.0.251        MDNS     185  Standard query 0x0000 PTR _companion-link._tcp.local, "QM" question PTR _rdlink._tcp.local, "QM" ques
  2  0.000625  fe80::72:8515:…   ff02::fb           MDNS     205  Standard query 0x0000 PTR _companion-link._tcp.local, "QM" question PTR _rdlink._tcp.local, "QM" ques
  3  0.002859  103.255.126.23    10.0.0.58          TCP       66  443 → 65209 [ACK] Seq=1 Ack=1 Win=27346 Len=0 TSval=691841604 TSecr=3887815053
  4  0.002860  103.255.126.23    10.0.0.58          TCP       66  443 → 65209 [ACK] Seq=1 Ack=33 Win=27377 Len=0 TSval=691841604 TSecr=3887815053
  5  0.002860  103.255.126.23    10.0.0.58          TCP       66  443 → 65225 [ACK] Seq=1 Ack=1 Win=15923 Len=0 TSval=691841599 TSecr=3054471056
  6  0.002860  103.255.126.23    10.0.0.58          TCP       66  443 → 65225 [ACK] Seq=1 Ack=33 Win=15954 Len=0 TSval=691841604 TSecr=3054471056
  7  0.002861  103.255.126.23    10.0.0.58          TCP       66  443 → 65224 [ACK] Seq=1 Ack=1 Win=15921 Len=0 TSval=691841611 TSecr=1846449105
  8  0.002861  103.255.126.23    10.0.0.58          TCP       66  443 → 65224 [ACK] Seq=1 Ack=33 Win=15952 Len=0 TSval=691841611 TSecr=1846449105
  9  0.204899  fe80::fad2:acf…   ff02::1            ICMPv6   174  Router Advertisement from f8:d2:ac:06:13:fb
 10  2.459198  fe80::72:8515:…   fe80::1808:4bf4:…  ICMPv6    86  Neighbor Solicitation for fe80::1808:4bf4:72a8:e019 from 06:d1:97:54:f5:55
 11  2.459335  fe80::1808:4bf…   fe80::72:8515:38…  ICMPv6    78  Neighbor Advertisement fe80::1808:4bf4:72a8:e019 (sol)
 12  3.276848  fe80::fad2:acf…   ff02::1            ICMPv6   174  Router Advertisement from f8:d2:ac:06:13:fb
 13  5.838548  103.255.126.23    10.0.0.58          TCP       66  443 → 65212 [FIN, ACK] Seq=1 Ack=1 Win=32228 Len=0 TSval=691847403 TSecr=1216119786
 14  5.838633  10.0.0.58         103.255.126.23     TCP       66  65212 → 443 [ACK] Seq=1 Ack=2 Win=2048 Len=0 TSval=1216140469 TSecr=691847403
 15  5.838834  10.0.0.58         103.255.126.23     TLSv1.2   97  Encrypted Alert
 16  5.839208  10.0.0.58         103.255.126.23     TCP       66  65212 → 443 [FIN, ACK] Seq=32 Ack=2 Win=2048 Len=0 TSval=1216140469 TSecr=691847403
 17  6.145703  103.255.126.23    10.0.0.58          TCP       66  [TCP Dup ACK 13#1] 443 → 65212 [ACK] Seq=2 Ack=1 Win=32228 Len=0 TSval=691847810 TSecr=1216140469
 18  6.145704  103.255.126.23    10.0.0.58          TCP       66  443 → 65212 [ACK] Seq=2 Ack=33 Win=32259 Len=0 TSval=691847810 TSecr=1216140469
 19  6.246422  fe80::fad2:acf…   ff02::1            ICMPv6   174  Router Advertisement from f8:d2:ac:06:13:fb
 20  6.334960  2601:8c0:800:6…   2606:4700:3033::…  TLSv1.2  187  Application Data
 21  6.358193  2606:4700:3033…   2601:8c0:800:61a…  TCP       86  443 → 65214 [ACK] Seq=1 Ack=102 Win=7 Len=0 TSval=2027921744 TSecr=2689208991
 22  7.129350  2601:8c0:800:6…   2600:1406:5e00:1…  TLSv1.2  110  Application Data
 23  7.129962  2601:8c0:800:6…   2600:1406:5e00:1…  TCP       86  65200 → 443 [FIN, ACK] Seq=25 Ack=1 Win=2048 Len=0 TSval=196388381 TSecr=864832422
 24  7.168557  2600:1406:5e00…   2601:8c0:800:61a…  TCP       98  443 → 65200 [ACK] Seq=1 Ack=1 Win=501 Len=0 TSval=864875804 TSecr=196345036 SLE=25 SRE=26
```

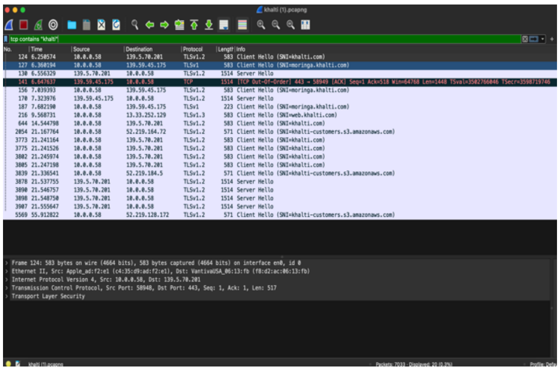
```
> Frame 1: 185 bytes on wire (1480 bits), 185 bytes captured (1480 bits) on interface en0, id 0
> Ethernet II, Src: 06:d1:97:54:f5:55 (06:d1:97:54:f5:55), Dst: IPv4mcast_fb (01:00:5e:00:00:fb)
> Internet Protocol Version 4, Src: 10.0.0.150, Dst: 224.0.0.251
> User Datagram Protocol, Src Port: 5353, Dst Port: 5353
> Multicast Domain Name System (query)
```

5.3 khalti.com

The IP address 139.5.70.201 and 139.5.70.200 are associated with the domain "Khalti.com". The ports used are 443 and 8081. Port 443 is used for the encrypted web traffic using HTTPS. The S in HTTPS stands for secure and indicates that the communication between the web browser and the web server is encrypted using SSL/TLS. It ensures that the data exchanged between client and the server is secure and is difficult to intercept or tampered by the third parties. Port 8081 is used as an alternative for default HTTP port 80. It allows web servers to run on a different port than the standard one. It is useful in scenarios where port 80 is already in use, or when an administrator wants to configure a server to listen on a non-standard port for various reasons.

The domain and subdomain used with their used traffic are shown in the table below.

Domain and Sub domain	Traffic Share
khalti.com	50.26%
blog.khalti.com	41.1%
web.khalti.com	7.45%
admin.khalti.com	0.51%
docs.khalti.com	0.45%
corporate.khalti.com	0.2%
test-admin.khalti.com	0.02%
a.khalti.com	0%
events.khalti.com	0%



```
No.   Time       Source          Destination      Protocol Length Info
 124  6.250574   10.0.0.58       139.5.70.201     TLSv1.2   583  Client Hello (SNI=khalti.com)
 127  6.360194   10.0.0.58       139.59.45.175    TLSv1     583  Client Hello (SNI=moringa.khalti.com)
 130  6.556329   139.5.70.201    10.0.0.58        TLSv1.2  1514  Server Hello
 141  6.647637   139.59.45.175   10.0.0.58        TCP      1514  [TCP Out-Of-Order] 443 → 58949 [ACK] Seq=1 Ack=518 Win=64768 Len=1448 TSval=3502766046 TSecr=3598719746
 156  7.039393   10.0.0.58       139.59.45.175    TLSv1.2   583  Client Hello (SNI=moringa.khalti.com)
 170  7.323976   139.59.45.175   10.0.0.58        TLSv1.2  1514  Server Hello
 187  7.682190   10.0.0.58       139.59.45.175    TLSv1     223  Client Hello (SNI=moringa.khalti.com)
 216  9.568731   10.0.0.58       13.33.252.129    TLSv1.3   583  Client Hello (SNI=web.khalti.com)
 644  14.544798  10.0.0.58       139.5.70.201     TLSv1.2   583  Client Hello (SNI=khalti.com)
2054  21.167764  10.0.0.58       52.219.164.72    TLSv1.2   571  Client Hello (SNI=khalti-customers.s3.amazonaws.com)
3773  21.241164  10.0.0.58       139.5.70.201     TLSv1.2   583  Client Hello (SNI=khalti.com)
3775  21.241526  10.0.0.58       139.5.70.201     TLSv1.2   583  Client Hello (SNI=khalti.com)
3802  21.245974  10.0.0.58       139.5.70.201     TLSv1.2   583  Client Hello (SNI=khalti.com)
3805  21.247198  10.0.0.58       139.5.70.201     TLSv1.2   583  Client Hello (SNI=khalti.com)
3839  21.336541  10.0.0.58       52.219.184.5     TLSv1.2   571  Client Hello (SNI=khalti-customers.s3.amazonaws.com)
3878  21.537755  139.5.70.201    10.0.0.58        TLSv1.2  1514  Server Hello
3890  21.546757  139.5.70.201    10.0.0.58        TLSv1.2  1514  Server Hello
3898  21.548750  139.5.70.201    10.0.0.58        TLSv1.2  1514  Server Hello
3907  21.555647  139.5.70.201    10.0.0.58        TLSv1.2  1514  Server Hello
5569  55.912822  10.0.0.58       52.219.128.172   TLSv1.2   571  Client Hello (SNI=khalti-customers.s3.amazonaws.com)
```

```
> Frame 124: 583 bytes on wire (4664 bits), 583 bytes captured (4664 bits) on interface en0, id 0
> Ethernet II, Src: Apple_ad:f2:e1 (c4:35:d9:ad:f2:e1), Dst: VantivaUSA_06:13:fb (f8:d2:ac:06:13:fb)
> Internet Protocol Version 4, Src: 10.0.0.58, Dst: 139.5.70.201
> Transmission Control Protocol, Src Port: 58948, Dst Port: 443, Seq: 1, Ack: 1, Len: 517
> Transport Layer Security
```

5.4 ipay.com.np

The IP address 43.245.85.179 is associated with the domain "ipay.com.np". The ports used are 80 and 443. Port 80 is the default port for unencrypted web traffic using HTTP. Port 443 is used for the encrypted web traffic using HTTPS. The S in HTTPS stands for secure and indicates that the communication between the web browser and the web server is encrypted using SSL/TLS. It ensures that the data exchanged between client and the server is secure and is difficult to intercept or tampered by the third parties. The domain and subdomain used with their used traffic are shown in the table below.

Domain and Sub domain	Traffic Share
www.ipay.com.np	100%

5.5 imepay.com.np

The domain and subdomain used with their used traffic are shown in the table below.

Domain and Sub domain	Traffic Share
www.imepay.com.np	97.34%
blog.imepay.com.np	2.56%
services.imepay.com.np	0.11%

6 Challenges

We encountered several challenges during this project. Firstly, we couldn't find any previous research work that we could use as a reference point. Also, during network traffic analysis, the traffic was so huge that it difficult to track all of them. To solve this, we looked at few of the captured data that was useful for this project. Also, some websites use privacy to protect their data, hence obfuscating the data and making it difficult for us to understand. Furthermore, we want to respect people's privacy. So, even though we did collect some personal information while conducting the project, we haven't disclosed it in our report. Also, some networks do things in dynamic ways and were complicated for our study. Filtering out unwanted background noise is another challenge that we faced during the project. Despite all these challenges, careful planning and teamwork helped us complete the project successfully.

7 Results and Conclusion

In conclusion, our project aimed to conduct an in-depth analysis of web application, especially focusing on network traffic using Wireshark as a primary tool. The selected web applications, including connectips.com, eswea.com.np, Khalti.com, ipay.com.np, and imepay.com.np, were inspected for understanding data transmission, security protocols, and overall performance. Through our methodology, we were able to achieve several significant findings and insights. We identified patterns and characteristics of data transmission within web applications. We uncovered TCP three-way handshakes, data transmission and connection termination sequence. We also explored SSL/TLS handshakes, indicating secure connections in web applications, and examined encryption methods and potential vulnerabilities in the communication process. We identified specific ports used in data transmission, such as 443 for secure web traffic. We also explored alternatives ports like 8081, providing insights into server configuration choices, port 993 ensures that the communication between the email client and server is encrypted using TLS and 995 uses TLS/SSL for secure communication channel between the client and the server. We also analyzed various domains and subdomains associated with each web application and calculated traffic shares for each domain and subdomains, highlighting the significance of specific components.

Our project successfully met its objectives, and the expected results were achieved. We uncovered valuable insights into data transmission patterns, security protocols, and potential vulnerabilities within the chosen web applications.

8 Future Work

The project focused mostly on application and transport layer protocols of the layered internet protocol stack. An even more exhaustive analysis can be done when the focus is shifted toward the network, link, and physical layers of the internet protocol stack. Perhaps, this can be achieved as a future work. Also, more work is needed on identifying different cyber-attacks that the web applications might be vulnerable to. So, another future work can involve a series of tests simulating a real-life cyber-attack through penetration testing. We believe that being able to gain perspectives as a network administrator and an attacker is important in securing web applications.