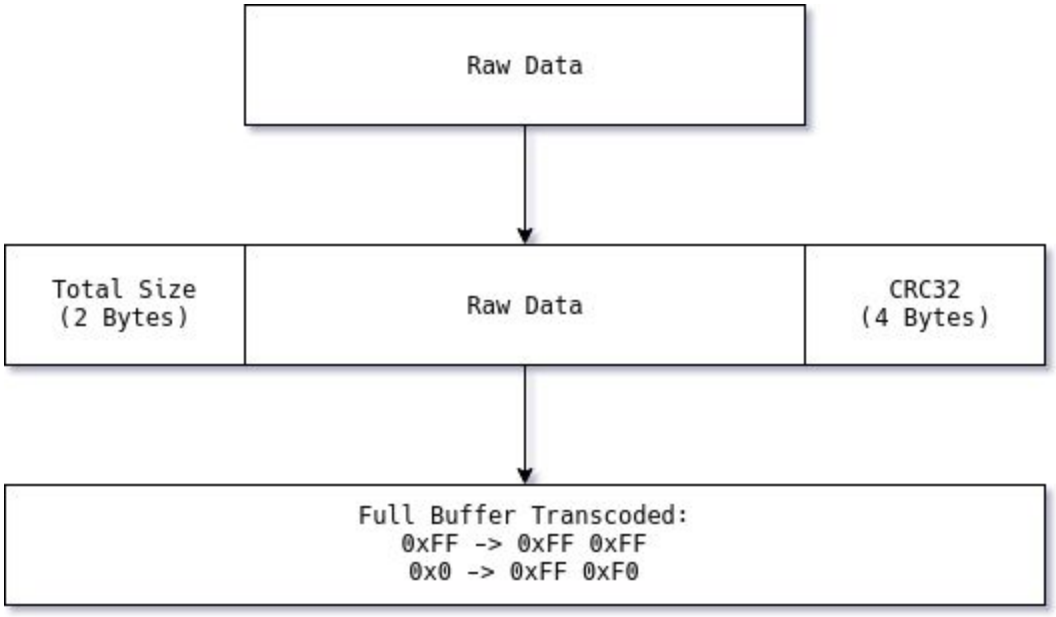
8505 Asn 1 - Design

Elementary my dear Watson

Isaac Morneau; A00958405

Encoding scheme

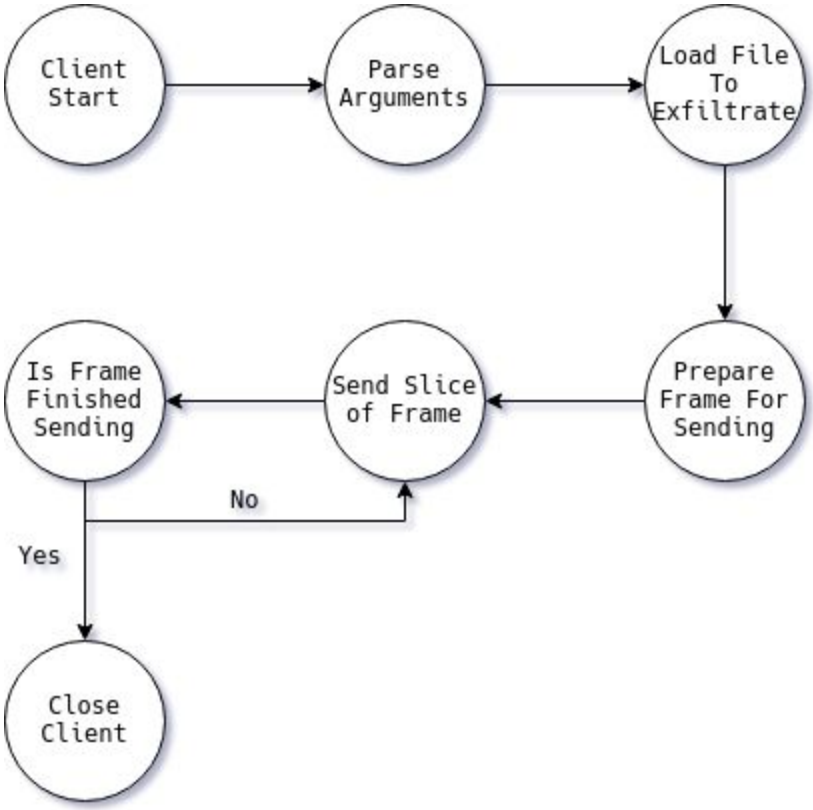


The Exfiltration of this application is, in its simplest form, using the source port of UDP frames to send two bytes at a time.

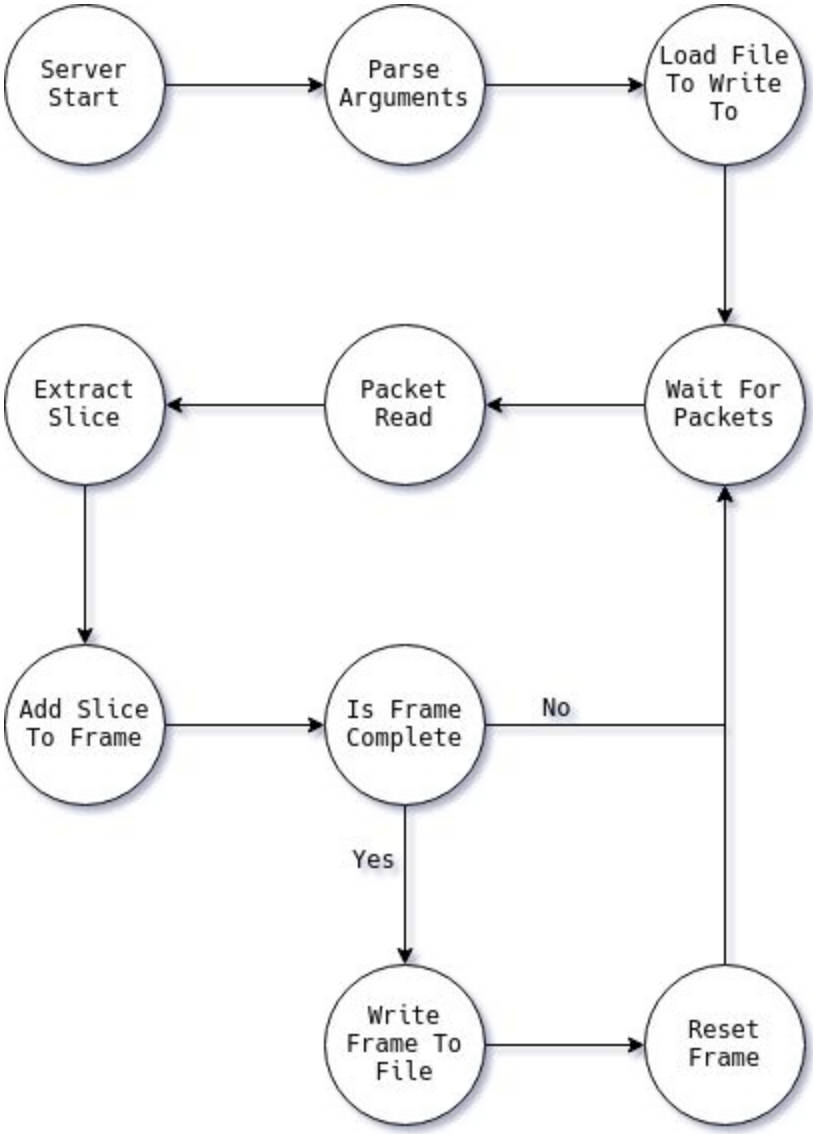
As UDP is unreliable and there is very little information transmitted at a time addition metadata was needed. This required an additional layer on top of the raw data to be transmitted which I called encoder frames as diagrammed in the second line above. This scheme also requires some values to be blacklisted, such as zeros. This is because a zero in the source port indicates that the sending process does not care about the return (described by RFC 768) but in this case it is better to mimic ephemeral ports used in unbound UDP packets. To solve this all data to be sent is transcoded into more appropriate values as seen in the final pass of the above diagram. On the receiving side the exact opposite is performed to retrieve the data.

FSM

Client


Client Start
Parse Arguments
Load File To Exfiltrate
Prepare Frame For Sending
Send Slice of Frame
Is Frame Finished Sending
No
Yes
Close Client

Server


Server Start
Parse Arguments
Load File To Write To
Wait For Packets
Packet Read
Extract Slice
Add Slice To Frame
Is Frame Complete
No
Yes
Write Frame To File
Reset Frame

Pseudocode

Client Start

Allocate the buffers for sending
Initialize Raw Sockets
Goto Parse Arguments

Parse Arguments

Get path of file to send
Get port to send to
Get delay of sending
If flag is set daemonize the process
Goto Load File To Exfiltrate

Load File To Exfiltrate

Open the file
Read it into a temporary buffer
Goto Prepare Frame For Sending

Prepare Frame For Sending

Load temporary buffer into an encoder frame
Finalize frame encoding
Goto Send Slice Of Frame

Send Slice Of Frame

Get next two bytes from the frame
Construct a raw socket UDP packet
Set the source port to the next two bytes
Send the packet
Goto Is Frame Finished

Is Frame Finished Sending

Check if there are more bytes to send
If there are
    Goto Send Slice Of Frame
Otherwise
    Goto Close Client

Close Client

Clean up buffers
Close Sockets
Exit

Server Start

Initialize Buffers
Initialize Sockets
Goto Parse Arguments

Parse Arguments

Get port to listen to
Get file path to write to
Goto Load File To Write To

Load File To Write To

Create or Open the file specified
Create the buffer to hold data to write
Goto Wait For Packets

Wait For Packets

Wait for epoll to signal data to be read
Goto Packet Read

Packet Read

Read the packet and buffer the address information
Goto Extract slice

Extract Slice

From the source, get the source port
Convert the port into bytes
Goto Add Slice to Frame

Add Slice To Frame

Add the bytes to an encoder frame

Goto Is Frame Complete

Is Frame Complete

If the frame is now full
    Goto Write Frame To File
Otherwise
    Goto Wait For Packets

Write Frame To File

Decode the data in the frame
Write buffer to the loaded file
Goto Reset Frame

Reset Frame

Reinitialize the frame to empty
Goto Wait For Packets