

# Performing No-frills File Transfers with Trivial FTP (TFTP)

---



**Nick Russo**

NETWORK ENGINEER

@nickrusso42518 [www.njrusmc.net](http://www.njrusmc.net)



# Agenda



**Why do we need TFTP?**

**TFTP in action with packet analysis**

**Securing TFTP**

**Comparing TFTP with FTP**



# Intention of TFTP

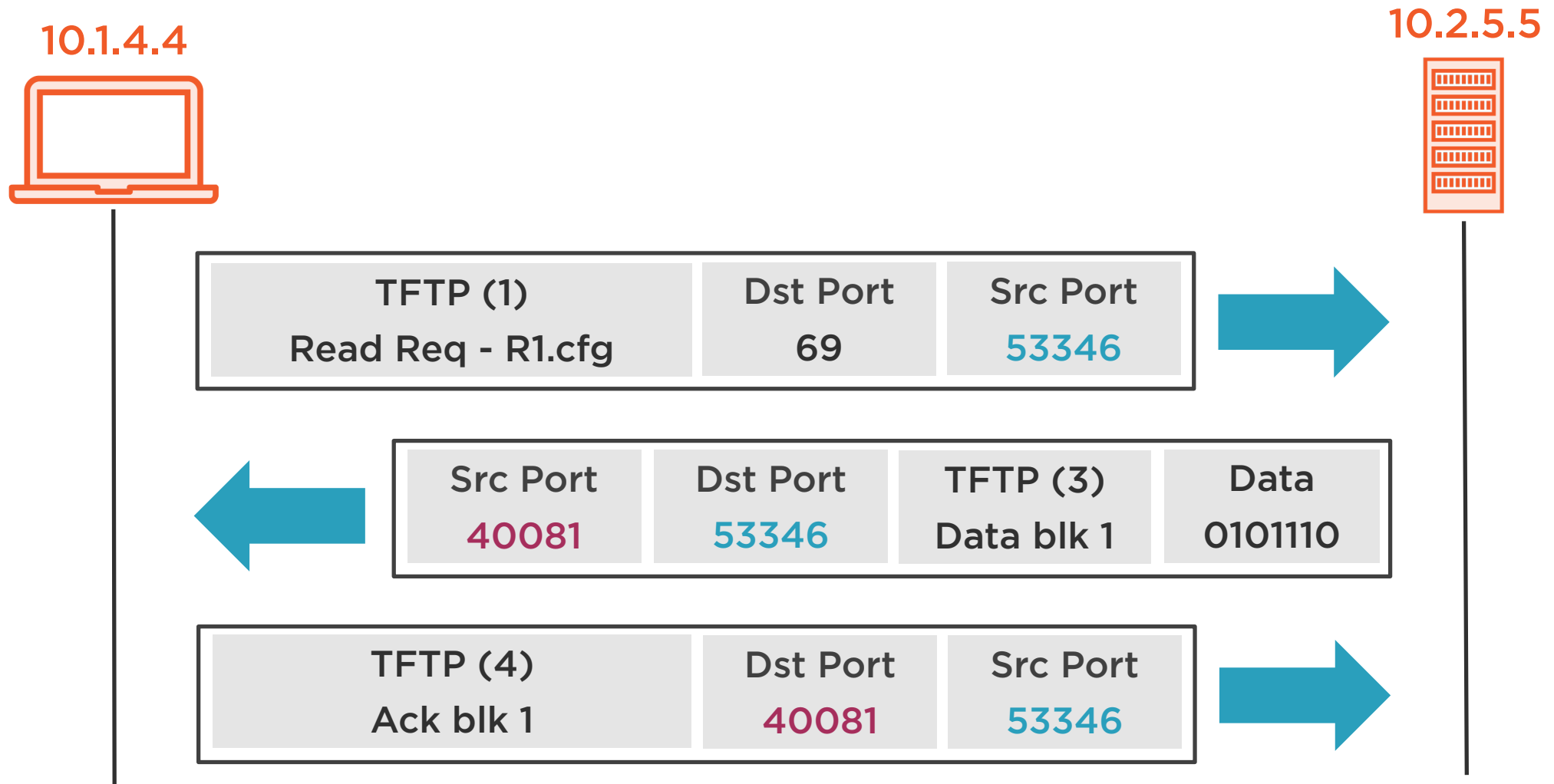
**Simple and easy  
to implement**

**Read and  
write files**

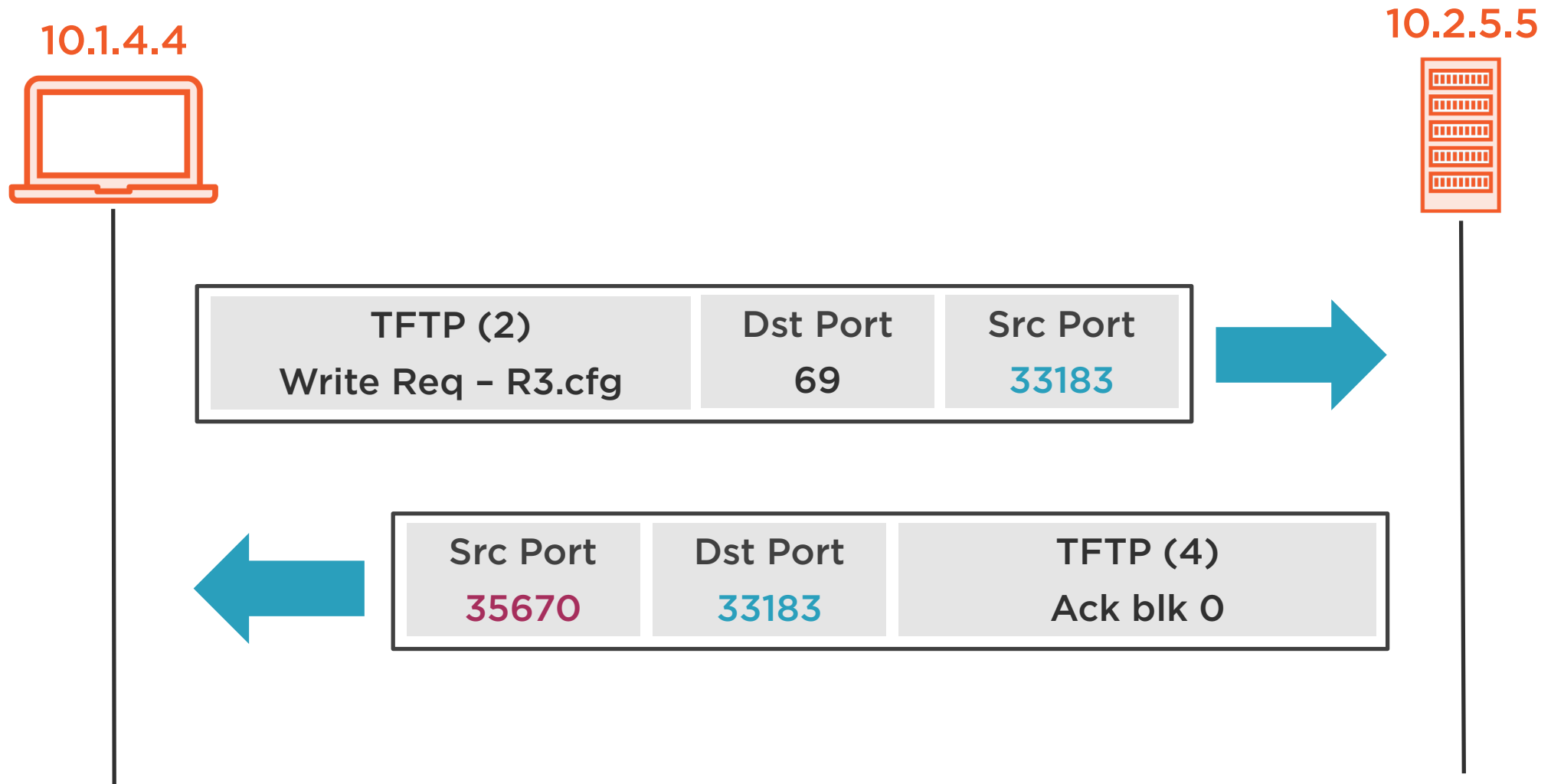
**Implemented  
on UDP**



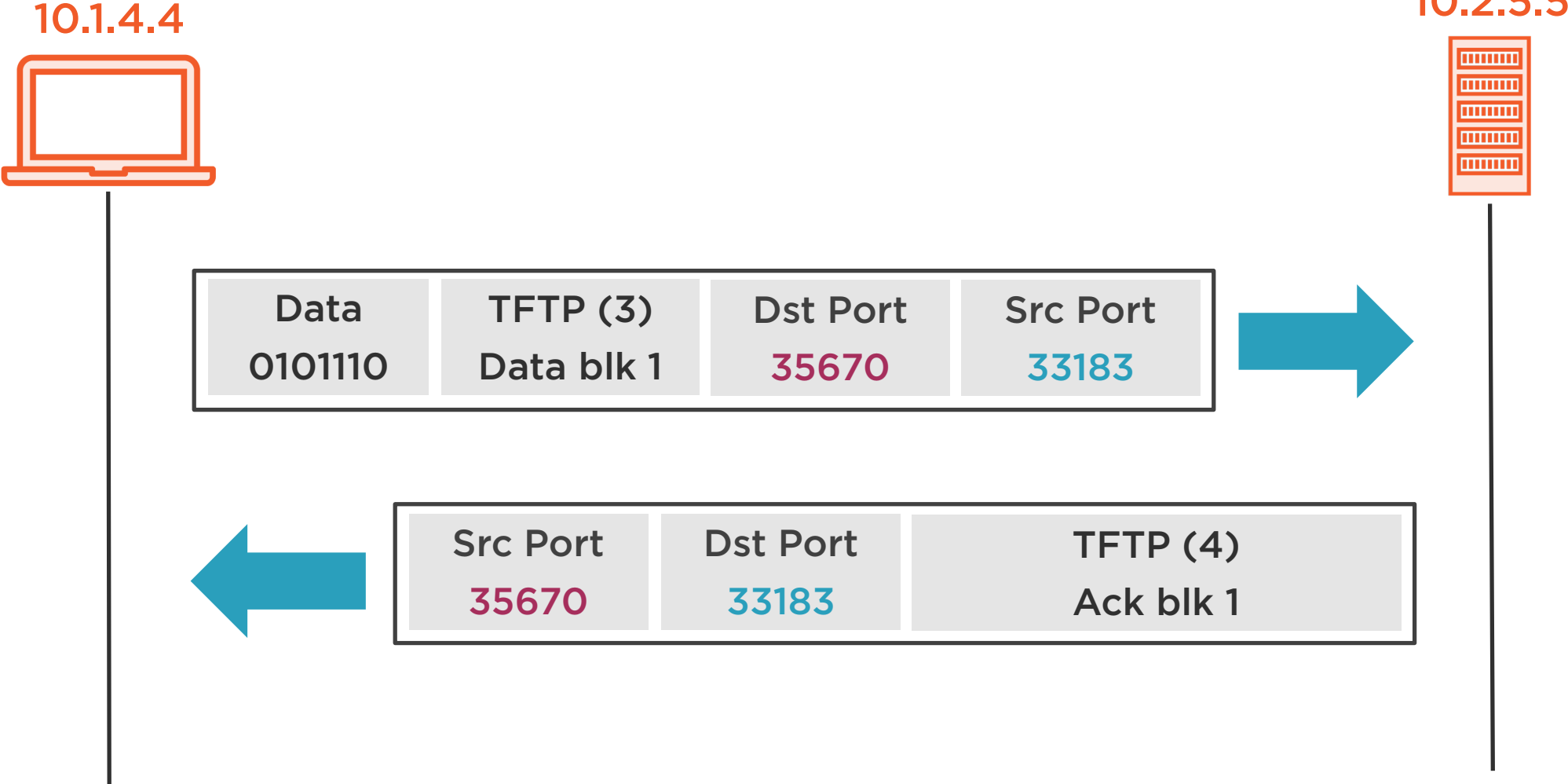
# Downloading Files with TFTP



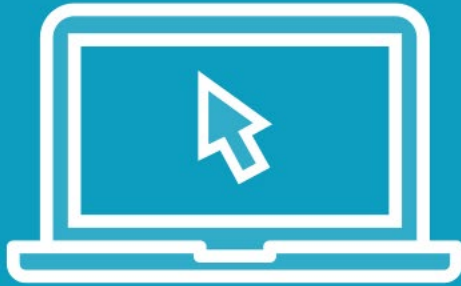
# Upload Setup with TFTP



# Uploading Files with TFTP



Demo



Downloading files with TFTP



# TFTP Download Analysis

No.	Source	Destination	Proto	Src Port	Dst Port	Opcode	Info
1	10.1.4.4	10.2.5.5	TFTP	53346	69	1	Read Request, File: R1.cfg, Transfer type: netascii
2	10.2.5.5	10.1.4.4	TFTP	40081	53346	3	Data Packet, Block: 1 (last)
3	10.1.4.4	10.2.5.5	TFTP	53346	40081	4	Acknowledgement, Block: 1



- ▶ Frame 2: 87 bytes on wire (696 bits), 87 bytes captured (696 bits) on interface 0
- ▶ Ethernet II, Src: 00:00:a6:16:00:01, Dst: 00:0c:29:9e:6d:dd
- ▶ Internet Protocol Version 4, Src: 10.2.5.5, Dst: 10.1.4.4
- ▶ User Datagram Protocol, Src Port: 40081, Dst Port: 53346
- ▼ Trivial File Transfer Protocol

Opcode: Data Packet (3)

[Source File: R1.cfg]

Block: 1

▼ Data (41 bytes)

Data: 68656c6c6f20776f726c640d0a746869732069730d0a5231...

[Length: 41]

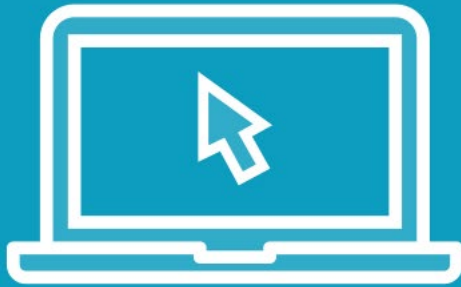
```
0000  00 0c 29 9e 6d dd 00 00  a6 16 00 01 08 00 45 00  ..).m... ..E.
0010  00 49 fc 2e 40 00 3d 11  24 6a 0a 02 05 05 0a 01  .I..@.=. $j.....
0020  04 04 9c 91 d0 62 00 35  b2 4d 00 03 00 01 68 65  ....b.5 .M...he
0030  6c 6c 6f 20 77 6f 72 6c  64 0d 0a 74 68 69 73 20  llo worl d..this
0040  69 73 0d 0a 52 31 27 73  20 63 6f 6e 66 69 67 20  is..R1's config
0050  66 69 6c 65 21 0d 0a
```

← File contents





Demo



**We should test uploads too!**



# TFTP Upload Analysis



No.	Source	Destination	Proto	Src Port	Dst Port	Opcode	Info
1	10.1.4.4	10.2.5.5	TFTP	33183	69	2	Write Request, File: R3.cfg
2	10.2.5.5	10.1.4.4	TFTP	35670	33183	4	Acknowledgement, Block: 0
3	10.1.4.4	10.2.5.5	TFTP	33183	35670	3	Data Packet, Block: 1 (last)
4	10.2.5.5	10.1.4.4	TFTP	35670	33183	4	Acknowledgement, Block: 1

- ▶ Frame 3: 97 bytes on wire (776 bits), 97 bytes captured (776 bits) on interface 0
- ▶ Ethernet II, Src: 00:0c:29:9e:6d:dd, Dst: 00:00:a6:16:00:01
- ▶ Internet Protocol Version 4, Src: 10.1.4.4, Dst: 10.2.5.5
- ▶ User Datagram Protocol, Src Port: 33183, Dst Port: 35670
- ▼ Trivial File Transfer Protocol

Opcode: Data Packet (3)

[DESTINATION File: R3.cfg]

Block: 1

▼ Data (51 bytes)

Data: 68656c6c6f20776f726c640d0a523327732066696c652069...

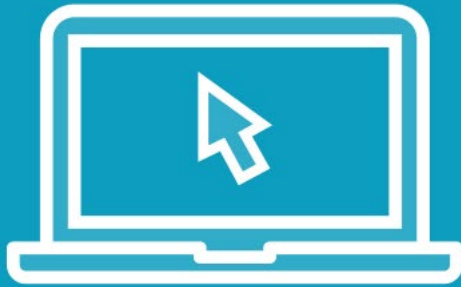
[Length: 51]

0000	00 00 a6 16 00 01 00 0c 29 9e 6d dd 08 00 45 00	..... ).m...E.
0010	00 53 92 97 40 00 40 11 8a f7 0a 01 04 04 0a 02	.S..@.@. ....
0020	05 05 81 9f 8b 56 00 3f fd 48 00 03 00 01 68 65	.....V.? .H....he
0030	6c 6c 6f 20 77 6f 72 6c 64 0d 0a 52 33 27 73 20	llo worl d..R3's
0040	66 69 6c 65 20 69 73 0d 0a 6d 75 63 68 20 6d 6f	file is. .much mo
0050	72 65 20 69 6e 74 65 72 65 73 74 69 6e 67 21 0d	re inter esting!.
0060	0a	.

← File contents



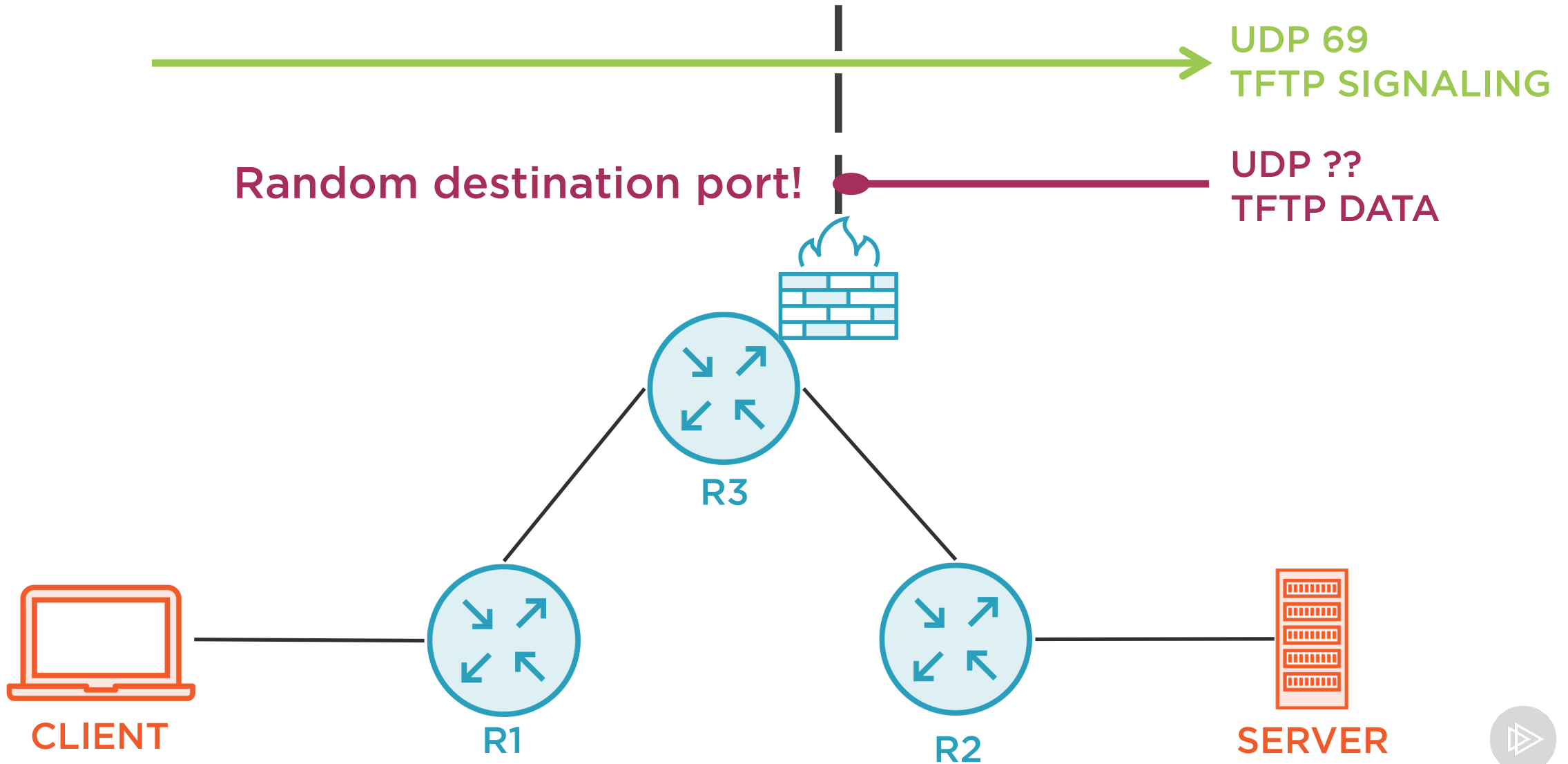
Demo



Attempting to secure TFTP



# TFTP Security Challenge



# Comparing TFTP and FTP

## TFTP

Simple and featureless

For internal use only

UDP with application-level acks

Good for quick transfers

## FTP

Chatty and feature-rich

Some variants work well with FW/NAT

TCP with inherent acks

Good for long-term file sharing

