

## Test BC-HLR-24

"O B/C, quando atuando como destino de um upload TFTP, deve implementar e enviar um ACK ou erro, garantindo confiabilidade do transporte UDP."

### Procedimento realizado

1. Análise do código em [modulo\\_bc/components/tftp/tftp.c](#)
2. O B/C deve estar ligado, em modo de manutenção
3. O GSE deve estar logado, conectado no wifi do B/C e o operador deve ter escolhido uma imagem para envio clicando em transferir
4. Verificar o recebimento de pacotes posteriores(se recebeu outros é porque ACK do anterior foi enviado)

### Resultados obtidos

Código mostrando envio do ack quando destino de um upload TFTP(handle\_wrq)

```
void handle_wrq(int sock, struct sockaddr_in *client, char *filename, lur_data_t *lur_file)
{
    socklen_t recv_len = sizeof(recv_addr);

    uint8_t lur_buf[256];
    size_t total_received = 0;

    while (1)
    {
        ssize_t n = recvfrom(transfer_sock, &pkt, sizeof(pkt), 0,
                             (struct sockaddr *)&recv_addr, &recv_len); /*BC-LLR-27*/
        if (n < 0)
        {
            ESP_LOGW(TAG, "Timeout ou erro no recebimento (errno=%d)", errno);
            break;
        }

        if ( ntohs(pkt.opcode) != OP_DATA || ntohs(pkt.data.block) != expected_block) /*BC-LLR-89*/
        {
            /* BC-LLR-18*/
            ESP_LOGW(TAG, "Pacote inesperado (opcode=%d, block=%d)",
                     ntohs(pkt.opcode), ntohs(pkt.data.block));
            upload_failure_count++;
            continue;
        }

        int data_len = n - 4;
        if (total_received + data_len <= sizeof(lur_buf))
        {
            memcpy(lur_buf + total_received, pkt.data.data, data_len);
            total_received += data_len;
        }

        ack.opcode = htons(OP_ACK); /*BC-LLR-90*/
        ack.block = htons(expected_block);
        sendto(transfer_sock, &ack, 4, 0, (struct sockaddr *)client, sizeof(*client)); /*BC-LLR-28*/

        ESP_LOGI(TAG, "Bloco %d recebido (%d bytes)", expected_block, data_len);
        expected_block++;

        if (data_len < BLOCK_SIZE)
            break; // Último bloco
    }
}
```

Logs com recebimento de pacotes posteriores confirmando ack anterior recebido:

```

I (183546) tftp: Bloco 1568 recebido (512 bytes)
I (183746) tftp: Bloco 1569 recebido (512 bytes)
I (183786) tftp: Bloco 1570 recebido (512 bytes)
I (184136) tftp: Bloco 1571 recebido (512 bytes)
I (184316) tftp: Bloco 1572 recebido (512 bytes)
I (184526) tftp: Bloco 1573 recebido (512 bytes)
I (184566) tftp: Bloco 1574 recebido (512 bytes)
I (184906) tftp: Bloco 1575 recebido (512 bytes)
I (185096) tftp: Bloco 1576 recebido (512 bytes)
I (185156) tftp: Bloco 1577 recebido (512 bytes)
I (185346) tftp: Bloco 1578 recebido (512 bytes)
I (185366) tftp: Bloco 1579 recebido (512 bytes)
I (185626) tftp: Bloco 1580 recebido (512 bytes)
I (185806) tftp: Bloco 1581 recebido (512 bytes)
I (185826) tftp: Bloco 1582 recebido (512 bytes)
I (186016) tftp: Bloco 1583 recebido (512 bytes)
I (186376) tftp: Bloco 1584 recebido (512 bytes)
I (186576) tftp: Bloco 1585 recebido (512 bytes)
I (186596) tftp: Bloco 1586 recebido (512 bytes)

```

