Alex Liang
2018年7月2日

SHOW CHAIN
点对点连接协议

NAT类型：

T1 全锥形
T2 受限锥形
T3 端口受限锥形
T4 对称形

1. UDP Keep Alive

点对点udp连接建立后，心跳包请求

Request:

```
{
 'op_code' : 0x100,
 'id' : $USER_ID,
 'seq' : 0,
 'ts' : 0
}
```

2. UDP Connect (Direct) : Tn —> T1

点对点udp直连请求 & 响应

Request:

{

‘op_code’ : 0x101,

‘id’ : $USER_ID,

‘target’ : $TARGET_ID,

‘seq’ : $CLIENT_PACKET_SEQUENCE,

‘ts’ : $CLIENT_TIMESTAMP,

‘sign’ : [op_code, target, seq, ts] RSA Signature

}

Response:

{

‘op_code’ : 0x102,

‘id’ : $USER_ID,

‘seq’ : $CLIENT_PACKET_SEQUENCE,

‘ts’ : $CLIENT_TIMESTAMP,

‘sign’ : [op_code, id, seq, ts] RSA Signature

}

3. UDP Connect (through NAT) : Tn —> T1 —> T(1-3)

Request:

{

‘op_code’ : 0x111,

‘target’ : $TARGET_ID,

‘id’ : $USER_ID,

‘seq’ : $CLIENT_PACKET_SEQUENCE,

‘ts’ : $CLIENT_TIMESTAMP,

‘sign’ : [op_code, target, seq, ts] RSA Signature

}

Pass Request:

{

‘op_code’ : 0x112,

‘target’ : $TARGET_ID,

‘id’ : $USER_ID,

‘seq’ : $CLIENT_PACKET_SEQUENCE,

‘ts’ : $CLIENT_TIMESTAMP,

‘ip’ : $REQUEST_IP,

‘port’ : $REQUEST_PORT,

‘session’ : $T1_GENERATE,

‘sign’ : $sign

}

Response(if target has same ip with requester) :

{

‘op_code’ : 0x113,

‘target’ : $TARGET_ID,

‘id’ : $USER_ID,

‘seq’ : $CLIENT_PACKET_SEQUENCE,

‘ts’ : $CLIENT_TIMESTAMP,

‘sign’ : [op_code, target, id, seq, ts] RSA Signature

}

Response ACK :

{

‘op_code’ : 0x114,

‘id’ : $USER_ID,

‘seq’ : $CLIENT_PACKET_SEQUENCE,

‘ts’ : $CLIENT_TIMESTAMP,

‘session’ : $T1_GENERATE,

‘sign’ : [op_code, session, seq, ts] RSA Signature

}

Pass Response ACK :

{

‘op_code’ : 0x115,

‘id’ : $USER_ID,

‘seq’ : $CLIENT_PACKET_SEQUENCE,

‘ts’ : $CLIENT_TIMESTAMP,

‘ip’ : $REQUEST_IP,

‘port’ : $REQUEST_PORT,

‘session’ : $T1_GENERATE,

‘sign’ : $sign

}

4. UDP Connect (proxy) : Tn —> T1 —> Tn

Request :

{

‘op_code’ : 0x121,

‘target’ : $TARGET_ID,

‘id’ : $USER_ID,

‘seq’ : $CLIENT_PACKET_SEQUENCE,

‘ts’ : $CLIENT_TIMESTAMP,

‘sign’ : [op_code, target, seq, ts] RSA Signature

}

Response :

{

‘op_code’ : 0x122,

‘session’ : $PROXY_GENERATE,

‘target’ : $TARGET_ID,

‘seq’ : $CLIENT_PACKET_SEQUENCE,

‘ts’ : $CLIENT_TIMESTAMP,

‘sign’ : [op_code, target, session, seq, ts] RSA Signature

}

5. UDP Send data (proxy) : Tn —> T1 —> Tn

Request :

{

‘op_code’ : 0x123,

‘target’ : $TARGET_ID,

‘id’ : $USER_ID,

‘data’ : $DATA,

‘session’ : $PROXY_GENERATE,

‘seq’ : $CLIENT_PACKET_SEQUENCE,

‘ts’ : $CLIENT_TIMESTAMP,

‘sign’ : [op_code, target, session, seq, ts] RSA Signature

}

Pass Request:

{

‘op_code’ : 0x124,

‘target’ : $TARGET_ID,

‘id’ : $USER_ID,

‘data’ : $DATA,

‘session’ : $PROXY_GENERATE,

‘seq’ : $CLIENT_PACKET_SEQUENCE,

‘ts’ : $CLIENT_TIMESTAMP

}

Response ACK :

{

‘op_code’ : 0x125,

‘session’ : $PROXY_GENERATE,

‘seq’ : $CLIENT_PACKET_SEQUENCE,

‘ts’ : $CLIENT_TIMESTAMP

}

6. UDP Connect (LAN) : Tn —> Tn

Request :

{

‘op_code’ : 0x131,

‘target’ : $TARGET_ID,

‘id’ : $USER_ID,

‘seq’ : $CLIENT_PACKET_SEQUENCE,

‘ts’ : $CLIENT_TIMESTAMP,

‘sign’ : [op_code, target, seq, ts] RSA Signature

}