TPKT, Version: 3, Length: 54

Version: 3
Reserved: 0
Length: 54

ITU-T Rec X.224

Length: 49
1110 … = Code: Connection Request (0x0e)
SRC-REF: 0x0000
0000 …. = Class: Class 0 (0x00)
RDP Routing Token: Cookie: msts=378652682.15629.0000

```
00 00 0c 07 ac 01 5c 26  0a 6c 79 74 08 00 45 00      ......\& .lyt..E.
00 5e 52 ba 40 00 80 06  8f 8e 0a c8 25 f9 0a c8      .^R.@... ....%...
dc c8 de 90 0d 3d 3d 8e  ad b1 65 b1 96 46 50 18      .....==. ..e..FP.
01 04 23 aa 00 00 03 00  00 36 31 e0 00 00 00 00      ..#...... .61.....
00 43 6f 6f 6b 69 65 3a  20 6d 73 74 73 3d 33 37      .Cookie:  msts=37
38 36 35 32 36 38 32 2e  31 35 36 32 39 2e 30 30      8652682. 15629.00
30 30 0d 0a 01 00 08 00  03 00 00 00                  00...... ....
```

- ***Set the session persistence as configured in the session persistence catalog***

```
$myPersistenceProfile = "TEST WMS RDP Persistence";
```

- ***Check to see if this is a new connection or traffic from an existing flow***

```
if ( !connection.data.get("first_run") ){
```

- ***Reads the TPKT Header, 4 bytes but skips the first 2 bytes 03 and 00 in this example***

```
$tpkt_head = request.get(4); $tpkt_head = string.skip($tpkt_head,2);
```

```
00 00 0c 07 ac 01 5c 26  0a 6c 79 74 08 00 45 00      ......\& .lyt..E.
00 5e 52 ba 40 00 80 06  8f 8e 0a c8 25 f9 0a c8      .^R.@... ....%...
dc c8 de 90 0d 3d 3d 8e  ad b1 65 b1 96 46 50 18      .....==. ..e..FP.
01 04 23 aa 00 00 [03 00  00 36] 31 e0 00 00 00 00    ..#..... .61.....
00 43 6f 6f 6b 69 65 3a  20 6d 73 74 73 3d 33 37      .Cookie:  msts=37
38 36 35 32 36 38 32 2e  31 35 36 32 39 2e 30 30      8652682. 15629.00
30 30 0d 0a 01 00 08 00  03 00 00 00                  00...... ....
```

- ***Populates the $rest_head variable by converting the remaining 2 byte header '00 36' to an integer '54' then subtracts 4 This leaves us with the remaining 50 byte payload, confirming there is indeed one***

```
$rest_head = string.bytesToInt($tpkt_head) - 4;
$total = $tpkt_head;
```

- ***As the variable $rest_head is greater than 0, due to the 50 bytes payload the script continues***

```
if( $rest_head > 0 ){
        $header = request.get(string.bytesToInt($total));
        string.skip($header,13);
```

- ***It now populates the $header variable by converting the 54 byte payload contained in $total to an integer less the leading 13 bytes, highlighted below in green***

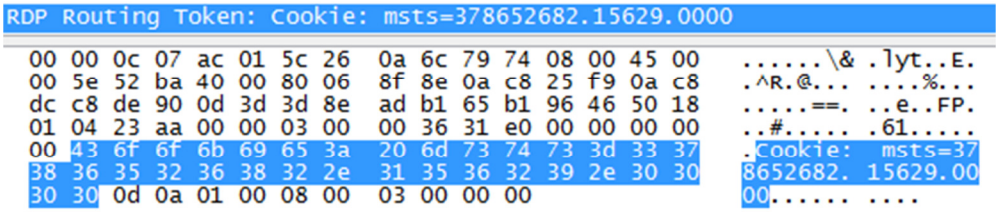
XX YY 03 00 00 36 31 e0 00 00 00 00 43 6f 6f 6b 69 65 3a

00 43 6f 6f 6b 69 65 3a 20 6d 73 74 73 3d 33 37 C o o k i e :

38 36 35 32 36 38 32 2e 31 35 36 32 39 2e 30 300

30 30 0d 0a 01 00 08 00 03 00 00 00

Note: there are 2 leading bytes not visible in the packet noted as XX and YY as highlighted above in green.



43 6f 6f 6b 69 65 3a 20 6d 73 74 73 3d 33 37

38 36 35 32 36 38 32 2e 31 35 36 32 39 2e 30 300

30 30 0d 0a 01 00 08 00 03 00 00 00

- ***We now search the $header variable for '0d 0a', the picture above shows that they trail the routing token.***

```
        $pos = string.find( $header, string.hexdecode("0D0A") );
    if( $pos > 0 ){
      $poss_token = string.left( $header, $pos );
          if( string.regexmatch( $poss_token, "[C|c]ookie:\\s(msts|mstshash)=(.+)$" ) ){
                $kval = $1; $vval = $2;
```

- ***If $pos matches condition i.e. contains 'OD OA' strip it and any following bytes.***
- ***The $poss_token variable is then populated with the remainder which in this case is the cookie a.k.a Routing Token.***

Routing Token Hex Value

43 6f 6f 6b 69 65 3a 20 6d 73 74 73 3d 33 37

38 36 35 32 36 38 32 2e 31 35 36 32 39 2e 30 300

30 30

Routing Token as clear text

Cookie: msts=378652682.15629.0000

- ***Lastly we search the $poss_token for key words Cookie or cookie / msts or mstshash to populate $kval and $vval respectively)***

```
if( $kval == "mstshash" ){
```

- ***If we have an mstshash cookie, let the session be load balanced normally, the script will then skip to.***

```
log.info("discard");
        connection.discard();
```

```
} else if( $kval == "msts" && string.regexmatch( $vval, "^(\\d+)\\.(\\d+)\\.\\d+$" ) ) {
        log.info("ip: ".$1);
        log.info("port: ".$2);
```

- ***Here we test the $kval variable to ensure it contains "msts" and $vval for a dot separated integer, the first position of the integer populates $1 then second $2 the last position is ignored (***msts=378652682.15629.0000)

- ***Here we parse the captured "msts=" cookie to extract the back end node info to route the connection***

```
        $ip = string.bytesToDotted( string.reverse( string.intToBytes( $1, 4 ) ) );
        $port = string.bytesToInt( string.reverse( string.intToBytes( $2, 2 ) ) );
        $node = $ip.":".$port; # NB: not IPv6 safe
        connection.setPersistence( $myPersistenceProfile);
        connection.setPersistenceNode( $node );
      } else {
        log.info("discard");
        connection.discard();
      }
    }
  }
}
# Set the connection run flag to prevent us from running this rule again on traffic from the same flow
connection.data.set("first_run","yes");
```