CVE-2020-0609 & CVE-2020-0610

Summary:

Pre-authenticated and non-interactive DOS/RCE attack exists in Windows Remote Desktop Gateway(RDG) when sending special requests while connecting using Remote Desktop Protocol(RDP).

Vulnerability:

RDG aka Terminal Services Gateway is a Windows Server component that provides routing for Remote Desktop Protocol(RDP). Rather than users connecting directly to an RDP Server, users instead connect and authenticate to the gateway. Upon successful authentication, the gateway will forward RDP traffic to an address specified by the user, essentially acting as a proxy. The idea is that only the gateway needs to be exposed to the Internet, leaving all RDP Servers safely behind the firewall.

We can connect to RDG using two protocols: DTLS over UDP and HTTP over TLS. Both the CVEs can be exploited only when UDP port 3391 is open and running on Windows Server 2012 R2 or Windows Server 2012 or Windows Server 2016 or Windows Server 2019.

CVE-2020-0609:

UDP is connectionless and packets can arrive in any order. To manage those packets, HandlePacket() is used. Each UDP packet header contains fields like num_fragments(total number of packets), fragment_id(packet's position in the sequence) and fragment_length(length of packet's data). Now, this function has some bugs. this->buffer is the buffer allocated on the heap, and memcpy_s copies the fragment data to the buffer, but multiplying by 1000, but this multiplying offset check is not done before. So, we can overflow into this heap buffer. The check is only on bytes_written +fragment_len which needs to be less than buffer_size.

CVE-2020-0610:

There exists an array frag_received of 32-bit unsigned integers. When a packet is received, the corresponding fragment_id is set. But the fragment_id can be any value between 0 and 65535 in a UDP Packet. Thus, we can overflow into this array with ones/zeroes by sending a UDP

packet with fragment_id b/w 65 and 65535(corresponding bits will be set).

```
int HandlePacket(UDPPacket* packet, DWORD packet_len) {

    if(!this->num_fragments)
        this->num_fragments = packet->num_fragments;

    if(packet->fragment_id > this->num_fragments)
        return error;

    int fragment_id = packet->fragment_id;
    if(this->frag_received[fragment_id])
        return ok;

    if((this->bytes_written + packet->fragment_len) > this->buffer_size)
        return error;

    this->frag_received[fragment_id] = TRUE;

    memcpy_s(&this->buffer[1000 * packet->fragment_id], 1000,
             &packet->fragment, packet->fragment_len);

    this->bytes_written += packet->fragment_len;

    if(this->AllFragmentsReceived()) {
        // do processing
    }
```

Exploit:

The exploit script can be used for scanning purposes and for Denial Of Service(DoS) attacks. We'll be sending a Packet with fragment_id of 0, no_of fragments of 65 and fragment length of 1. Depending on the error code of 0x8000ffff, we can check whether the Service is vulnerable or not. For DoS attacks, we'll be sending packets with fragment_id greater than 64, and sending a large number of packets. This will cause the service to crash and stop.

Though we're able to write the array buffer out of bound with ones/zeroes, it would be difficult to get a RCE.