

Forum:[Hack Forums](#)

Thread: [\[TuT\] \[HQ\] Converting ASM to Shellcode](#) | [What is ASM](#) | [What is shellcode \[HQ\]](#) [/TuT]

lillix - April 6, 2013, 11:03 pm

In this tutorial we will cover

What is ASM?

What is shellcode?

Windows shell coding (coming soon)

Linux shell coding

What is ASM?

An assembly programming language is a low-level programming language for computers, be it a mobile device, a microwave, or a personal computer. Every single programmed piece of technology on the planet contains somewhat a large quantity of machine code instruction, which, obviously instructs the piece of technology to do certain things. Machine code is eventually converted into an executable file by a compiler, although other information sources would refer to these as an 'assembler', although if you are just starting, the two objects contain the exact same characteristics. Low level languages have been around for over 60 years, in contrast to high level programming languages, which have been around only 10-20 years.

What is shellcode?

Shellcode is a small piece of code, rarely used in the programming world, known to more hackers than the 'average joe'. Shellcode can be used to take advantage of exploits, hereby being called the 'payload'. Exploits, payloads and vulnerabilities are separate things: A vulnerability is a flaw in a machine/application's programming. An exploit is a piece of code that someone has discovered that can take advantage of the vulnerability at hand. A payload is a packet which is injected into the exploit, then giving the attacker control of the machine (s)he has compromised. Shellcode isn't always used for bad intentions.

Local Shellcode

Local shellcode is where an attacker has limited permissions to a machine, but can identify and exploit a possible vulnerability, and if this exploit succeeds, the attacker's permission is raised to a higher level, allowing (s)he to apply more damage to the machine.

Remote Shellcode

Remote shellcode is where the attacker doesn't have local & direct access to the machine (s)he is attempting to access, and instead uses an internet connection to accomplish the goal (s)he is trying to achieve. These type of attacks usually take place through GET and POST packets being

distributed back and forth. These packets are malicious and contain code that spawns a shell on the attacker's victim.

Windows shell coding (coming soon)

Linux shell coding

Requirements:C++ IDE ([Code::Blocks](#) / [Visual C++](#) / [Dev C++](#))

Linux box/linux virtual machine ([VirtualBox](#) / [VMWare](#))

gcc (sudo apt-get install gcc)

ld (not sure where this software package is located)

nasm (sudo apt-get install nasm)

objdump (not sure where this software package is located)

arwin ([source link](#))

To test our shell code once we have obtained it, we will have to make a new C++ project, so go ahead and make that, then in your main class, put this inside it:

```
Code #include <fcntl.h>
#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <stdlib.h>
#include <string.h>
#include <iostream>
#include "windows.h"
```

```
int main(int argc, char **argv)
{
    char shellcode[] =
        "Shellcode goes here";

    printf("Shellcode Length is: %d",strlen(shellcode));

    LPVOID lpAlloc = NULL;
    void (*pfunc)();

    lpAlloc = VirtualAlloc(0, 4096, MEM_COMMIT, PAGE_EXECUTE_READWRITE);

    if(lpAlloc == NULL){
        printf("Memory isn't allocated!\n");
        return 0;
    }
}
```

```

memcpy(lpAlloc, shellcode, strlenA((LPCSTR)shellcode) + 1);
pfunc = (void (*)())lpAlloc;
pfunc();
return 0;
}

```

If you run it now, it won't work, it will stop working upon the time when it finds no hex's in the char array.

To start off easy, we'll just make an exit function.

```

Code ;exit.asm
[SECTION .text]
global _start
_start:
xor eax, eax ;exit is syscall 1
mov al, 1 ;exit is syscall 1
xor ebx,ebx ;zero out ebx
int 0x80

```

The above code is ASM, although these languages are hard to comprehend, the power behind them is unbelievable. Open up a text editor, and paste the code in, and save the file as a .asm. Once you have saved the file, open a terminal window, and navigate to where you saved the file. Once you are in the file's directory; execute the following command:

```
Code nasm -f elf filename.asm; ld -o filename filename.o; objdump -d filename;
```

If you executed that command correctly, you should now see a table of hex codes that looks like the following:

```

Code 08048080 <_start>:
8048080: b0 01 mov $0x1,%al
8048082: 31 db xor %ebx,%ebx
8048084: cd 80 int $0x80

```

If you see that, then you're going the right way! The hex numbers: Code b0 01 31 db cd 80 are the machine instructions, and this is how your machine sees code after your compiler/assembler has compiled it. We want these hex numbers. If you're familiar with the shellcode syntax, you'll know that \x comes before the hex number, so eventually we want these hex numbers to look like:

```
Code \xb0\x01\x31\xdb\xcd\x80
```

And remember our char array called shell code? Insert this shell code into that char array, then run your program. If you have successfully undertaken the tasks, you shouldn't be getting any errors. If you're using code::blocks and your getting an error, change your compiler settings to default and you should be fine.

Thanks for reading.

EDIT: For shellcode exploits, I recommend you venture to <http://exploit-db.com>

That's it for tonight, I'll add more tomorrow when I wake up.

Makaveli_Tupac - April 6, 2013, 11:08 pm

Thanks for diz you sexi beast, Luv'd eet <3

2564lert - April 6, 2013, 11:32 pm

Thanks, this helped a lot, keep it up!

??? -? ? ??? ??? - April 7, 2013, 5:20 am

Thanks alot, keep your work up man! ;)

mibbzz - May 23, 2013, 1:40 pm

Woah thanks for this, it helped me out A TON!!

Sadlarry404 - May 23, 2013, 3:52 pm

Code section .text

```
global _start
_start:
push 0x2
pop eax
int 0x80
jmp _start
nasm -f elf /path/to/file.asm
ld -o name /path/to/file.o
```

Don't run unless you feel like rebooting your system.

xMannyG - May 25, 2013, 12:56 pm

Thanks for the amazing tutorial and the work you went through to make it. :) Vouch Very HQ user.

curiousclocker - May 25, 2013, 10:53 pm

How to convert shell code to EXE? Or How to use Shell Code?

lillix - June 1, 2013, 5:34 am

(May 25, 2013 10:53 PM)curiousclocker Wrote: [\[CITE\]](#) How to convert shell code to EXE? Or How to use Shell Code?

How to use shell code. If you want a shell code to EXE application, I suggest you look here:
<http://zeltser.com/reverse-malware/convert-shellcode.html>

?YY? - June 9, 2013, 5:38 am

Nice tutroial <3. Enjoyed it.

Hope this helps some peeps.

Makaveli_Tupac - June 9, 2013, 5:39 am

Thanks for this.

Hope it helps some people.

-Makaveli_Tupac

lillix - June 9, 2013, 5:39 am

(Jun 9, 2013 05:38 AM)bdfrady1234 Wrote: [\[CITE\]](#) Nice tutroial <3. Enjoyed it.

Hope this helps some peeps.

Thanks. I'll add more to the tutorial later.

??a?i?a?io?™ - July 6, 2013, 2:18 pm

You always have such HQ tutorials, mate!

Gotta make sure I don't miss any! ?

Eric Berne - July 6, 2013, 2:37 pm

Nice introduction to shellcodes, doesn't miss out the little details and explains them unlike some other tutorials.

Personally I am interested in Windows part and I hope you update it. Anyhow, great tutorials mate.