

A cartoon illustration of SpongeBob SquarePants in a thinking pose, with his hand to his chin. He is wearing his signature white shirt and red tie. The background is a vibrant, stylized underwater scene with various sea anemones, bubbles, and a blue wavy shape that frames the text.

The Krabby Patty Secret Recipe

An Intro to Shellcode Running

Live at Neon Temple with @DiscOrdantMel0dy

What Do?



Introduction



What is a shellcode loader?



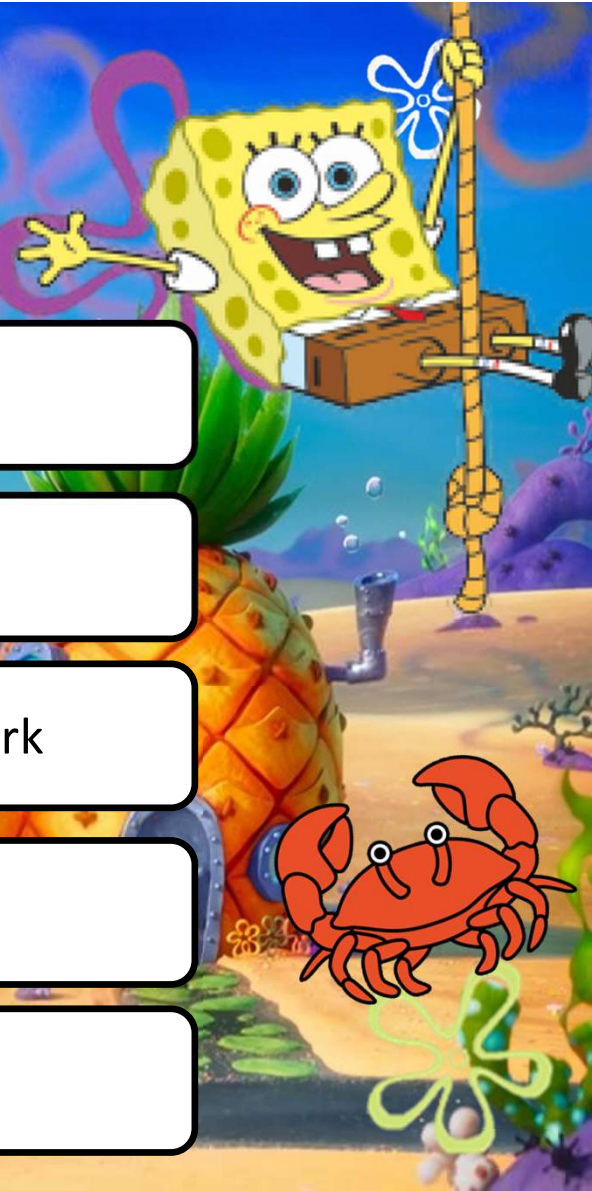
Windows API Calls within C# .NET Framework



A Simple Shellcode Loader



Questions



Some Definitions and Stuff

Since your host is too dumb to remember
to use one term or the other

Implant == RAT (remote access tool)
Shellcode Loader == Shellcode Runner

TRIGGER WARNING:

In the following hour you will be exposed to:

Bad Humor (and possibly language)

Even Worse Code

(Shellcode Running is a Programming Sport)

Microsoft Windows

ENOUGH SPONGEBOB TO MAKE YOUR EYES BLEED!!!



A cartoon illustration of SpongeBob SquarePants is positioned on the left side of the slide. He is yellow with large blue eyes, a wide smile showing his teeth, and is wearing his signature brown square pants and red tie. He appears to be peeking from behind a white rounded rectangle that contains the main text. The background of the slide is a light blue gradient with a dashed black border around the white rectangle. A blue textured banner is at the top right.

Whoami

DiscOrdantMelOdy (Chris Russell)

PowerShell and C# Nerd (Note: Not a real developer!)

♥'s Malware Development

Leads Offensive Operations Group @ Fulcrum Technology Solutions

Enjoys sticking his DLL in strange programs

WTF is a shellcode loader?

A shellcode loader is a wrapper program to execute shellcode:

- Shellcode can not be called on its own as it is raw bytes meant to be executed directly from a process's memory space.
- Shellcode is only executed if it is copied into the memory of a running process and then the execution pointer is aligned with the beginning of the shellcode.



OK... WTF is Shellcode Then?



Shellcode is:

- A small bit of low-level code used to carry out a specified set of instructions.
- Usually used to load remote access tool (RAT) or execute commands on a remote host.

Shellcode is not:

- A fully functional program
- Something that can be executed from the command line or GUI of an operating system (OS)



WTF is a shellcode loader?

To execute shellcode a loader must do four things:

1. Import the shellcode representation into a byte array.
2. Allocate (the right amount of) memory inside a process.
3. Copy the bytes contained in the shellcode array to the allocated memory region.
4. Set the Execution Instruction Pointer (EIP) to the beginning of the shellcode.

Everything else is just evasion!



How Do? (Windows API Calls)



To perform these four functions, we need to leverage Windows API Calls.

A Windows API call is a function provided by the OS (or from a DLL file) to accomplish a specific task.

They can only be called from inside a program.



Managed vs Unmanaged Code

Managed Code:

- Easy to use (short-hand) functions which call lower-level APIs
- Imported into the program with the “Using” command
- Relies on the .NET Framework being Installed

Unmanaged Code:

- Used to call any other Windows API call.
- Are imported from an external source (DLL File) using DLLImport keyword.
- Each function from the DLL is imported separately.



A cartoon illustration of SpongeBob SquarePants, a yellow sponge with large blue eyes, a wide smile, and a red tie. He is holding a brown rectangular object, possibly a sign or a piece of paper, and is standing on a purple, textured surface. The background is a light blue sky with a single white bubble.

Five Minute Windows Internals

Windows OS's (really ALL OS's) are basically divided into two logical spaces (or scopes)

Userland:

- This is the space you traditionally think of when you think of an OS.
- The GUI and all external applications run in this space.

Kernel:

- This is where the OS does all its internal calls.
- The security rules of the OS state that userland CANNOT interact with the Kernel directly.

Windows only SUPPORTS applications interacting with the Kernel through specific APIs defined in Kernel32.dll

RTFM.....

You can find all the available windows API calls on Technet
<https://learn.microsoft.com/en-us/windows/win32/api/>

Let's look at a simple one we are going to use later...

<https://learn.microsoft.com/en-us/windows/win32/api/synchapi/df-synchapi-waitforsingleobject>



Snudge



Step 3 – Copy Shellcode to Memory – Marshal.Copy

This API call also requires 4 parameters:

1. A source array of bytes to copy from
2. A start index of that array (usually 0 to start from beginning).
3. A pointer to an allocated memory destination
4. The length of the SOURCE array to be copied to destination

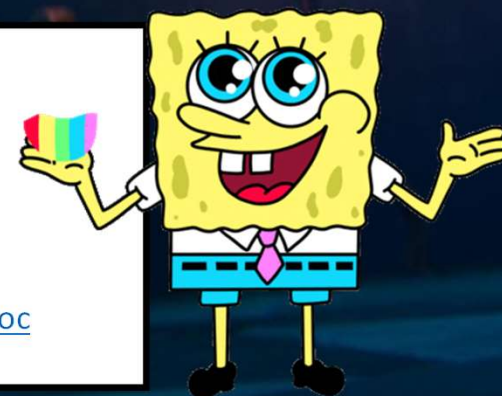
<https://learn.microsoft.com/en-us/dotnet/api/system.runtime.interopservices.marshal.copy?view=net-8.0>

Step 2 – Allocate Memory – VirtualAlloc

This API call requires 4 parameters:

1. Base address of the allocated memory region.
2. Size of space to be allocated.
3. Type of memory allocation.
4. Memory protection mask of the allocated memory.

<https://learn.microsoft.com/en-us/windows/win32/api/memoryapi/nf-memoryapi-virtualalloc>







Step 4.5 – WaitForSingleObject

This API call only requires 2 parameters:

1. A special type of pointer called a handle (thread pointer)
2. An unsigned integer defining how long the OS should wait before terminating the thread

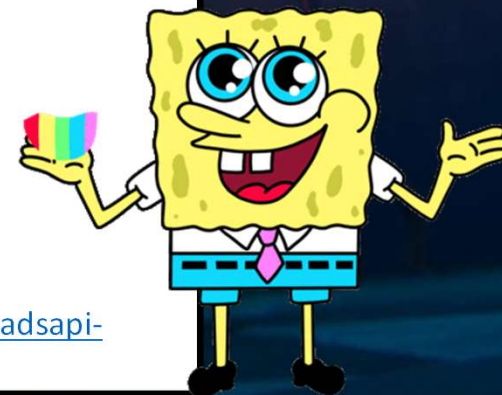
<https://learn.microsoft.com/en-us/windows/win32/api/synchapi/nf-synchapi-waitforsingleobject>

Step 4 – Set Execution Pointer – CreateThread

This API call requires 6 parameters (With both In and Out Parameters):

1. A pointer to the security attributes for the thread.
2. An unsigned integer which describes the stack size.
3. A pointer to the memory space containing instructions.
4. A pointer to the thread parameters.
5. An unsigned integer which sets the thread creation flags
6. An unsigned integer which sets the thread ID

<https://learn.microsoft.com/en-us/windows/win32/api/processthreadsapi/nf-processthreadsapi-createthread>







Shellcode Storage

There are a many ways to store your shellcode inside your loader.


You can even design your loader to fetch the shellcode from a remote location!

Today I am going to show you the simplest way which is to hardcode your shellcode into the program.




DEMO TIME


Shellcode Loader RECAP




Import hard-coded shellcode string to byte array



Allocate memory equal to shellcode size with VirtualAlloc



Copy shellcode to allocated memory region with Marshal.Copy



Set EIP to beginning of shellcode with CreateThread



PROFIT!!!!



Questions, Comments, Manifestos?

-Cochise The Poet Laureate

There are no dumb questions, only dumb people.

-Some Pretentious Asshole Calling Himself DiscoOrdant Melody KEK



