Adventures in Shellcode Obfuscation! Part 2: Hail Caesar!

redsiege.com/blog/2024/06/adventures-in-shellcode-obfuscation-part-2-hail-caesar

By Red Siege | June 24, 2024

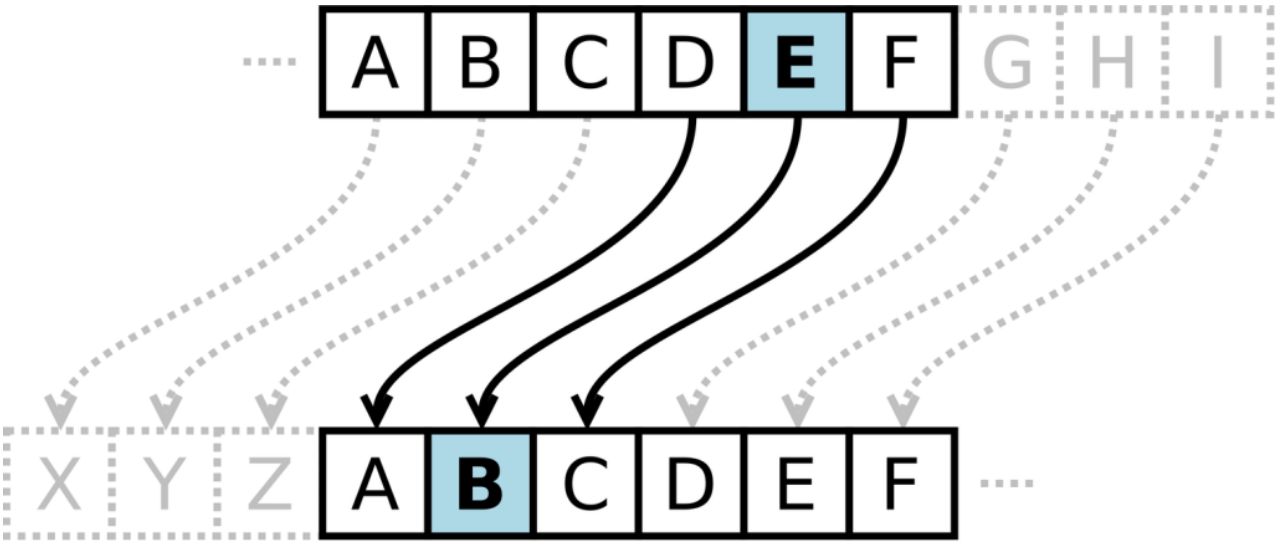
by Mike Saunders, Principal Security Consultant

Watch Video At: https://youtu.be/hjNLylCAmBo

This blog is the second in a series of blogs on obfuscation techniques for hiding shellcode. You can find the rest of the series here. If you'd like to try these techniques out on your own, you can find the code we'll be using on the Red Siege GitHub. Let's look at an encryption method we can use to hide our shellcode.

Caesar Cipher

The Caesar cipher is one of the most simple and well-known encryption schemes, dating to the times of its namesake, Julius Caesar. The Caesar cipher is a substitution cipher, meaning a letter in the plaintext message is shifted by a certain number of characters. For example, you could shift each letter by three. As a result, A would become D, B would become E, and so on.



Caesar Cipher – Three Letter Shift
Cepheus, Public domain, via Wikimedia Commons

You've probably heard of ROT13, another popular example of a Caesar cipher. ROT13 shifts the alphabet by – you guessed it – 13 characters. We can do the same thing with shellcode. To encode our shellcode, we take our shellcode byte and add 13 (0x0d) to it. If the result is greater than 255, we subtract 255 from the result to allow us to wrap around to the beginning of our shellcode alphabet (0x00 to 0xff). Given a list of shellcode bytes, we can use the following Python code to encrypt our shellcode:

```
def caesar(sc_list):

sc = []

for x in sc_list:

if (int(x) + 13) > 255:

sc.append(hex(x + 13 - 256))

else:

sc.append(hex(x + 13))

return sc
```

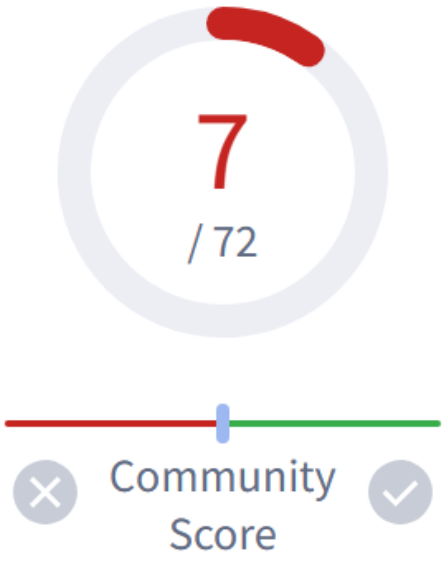
In our loader, we need to reverse that math. We subtract 13 from our shellcode byte. If the result is less than 0, then we add 256. The following C code can be used to decrypt our shellcode. Note the code and comment on line 5. When I tested the compiled program with ThreatCheck, Defender was alerting on the decryption routine. Throwing this printf statement was enough to break the signature to prevent Defender from detecting it.

```
for (int i = 0; i < sizeof(caesar); i++)

{
```

```
if ( (caesar[i] - 13) < 0 )

{

printf(""); // because defender

shellcode[i] = caesar[i] + 256 - 13;

}

else

{

shellcode[i] = caesar[i] - 13;

}

}
```

Given how well-known the Caesar cipher is, certainly all of the AVs out there must detect a program containing Metasploit Meterpreter shellcode, right? Sadly, that appears to not be the case. As you can see below, only 7 engines detected the test program as malicious.

When I tested that, I thought “Surely, this can’t be true!” I generated raw Cobalt Strike beacon shellcode without any evasion capabilities added through the artifact, sleep, or mutator kits. After adding the Caesar ciphered Cobalt Strike shellcode to the test program, the detection rate actually went down to 6! Surprisingly, in 2024, nearly 2000 years after first being used by its namesake, a cipher that can be recognized and broken by children seems to be capable of stymying modern AV.



Caesar Cipher Detection Rate

Try it Yourself

You can find the example code for this article as well as the other articles in this series at the Red Siege GitHub.

Stay Tuned

This blog is part of a larger series on obfuscation techniques. Stay tuned for our next installment!

About Principal Security Consultant Mike Saunders

Mike Saunders is Red Siege Information Security's Principal Consultant. Mike has over 25 years of IT and security expertise, having worked in the ISP, banking, insurance, and agriculture businesses. Mike gained knowledge in a range of roles throughout his career, including system and network administration, development, and security architecture. Mike is a highly regarded and experienced international speaker with notable cybersecurity talks at conferences such as DerbyCon, Circle City Con, SANS Enterprise Summit, and NorthSec, in addition to having more than a decade of experience as a penetration tester. You can find Mike's in-depth technical blogs and tool releases online and learn from his several offensive and defensive-focused SiegeCasts. He has been a member of the NCCCDC Red Team on several occasions and is the Lead Red Team Operator for Red Siege Information Security.

Certifications:
GCIH, GPEN, GWAPT, GMOB, CISSP, and OSCP

Related Stories

View More

Adventures in Shellcode Obfuscation! Part 7: Flipping the Script

By Red Siege | August 1, 2024

by Mike Saunders, Principal Security Consultant This blog is the seventh in a series of blogs on obfuscation techniques for hiding shellcode. You can find the rest of the series […]

Learn More
Adventures in Shellcode Obfuscation! Part 7: Flipping the Script

Out of Chaos: Applying Structure to Web Application Penetration Testing

By Red Siege | July 25, 2024

By Stuart Rorer, Security Consultant As a kid, I remember watching shopping contest shows where people, wildly, darted through a store trying to obtain specific objects, or gather as much […]

Learn More
Out of Chaos: Applying Structure to Web Application Penetration Testing

Adventures in Shellcode Obfuscation! Part 6: Two Array Method

By Red Siege | July 23, 2024

by Mike Saunders, Principal Security Consultant This blog is the sixth in a series of blogs on obfuscation techniques for hiding shellcode. You can find the rest of […]

Learn More
Adventures in Shellcode Obfuscation! Part 6: Two Array Method