

# FIDLing with decompilers

---

Ryan Warns & Carlos Garcia Prado

*Diggers of APIs, FireEye*

002  
**HITB LOCKDOWN**  
livestream



# Agenda

- Source code analysis... without source code?
- Decompilers are hard
- IDA is hard
- FIDL to the rescue!
- Scaling things up!

Ryan Warns, Carlos Garcia Prado

FIDLing with decompilers

# Who dis?

- Ryan Warns
  - Mega-Nerd
  - Likes the Windows kernel and long walks on the beach
  - Reversed malware -> wrote malware -> reversed malware
- Carlos Garcia Prado (Carl OS)
  - Particle Physics background
  - <3 IDA, Python, Windows stuff
  - Automated > Manual
  - Around 10 years in offensive (security) roles

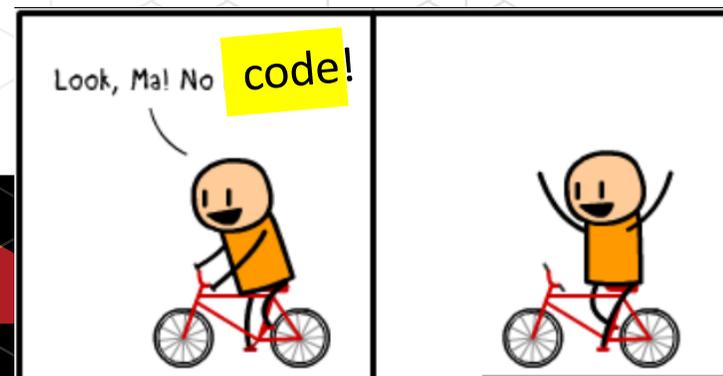




# Other interesting work around decompilation

- Everything from @pat0is
  - <https://github.com/patois>
- HexRaysPyTools
  - <https://github.com/igogo-x86/HexRaysPyTools>

# Source Code Analysis... without Source Code?





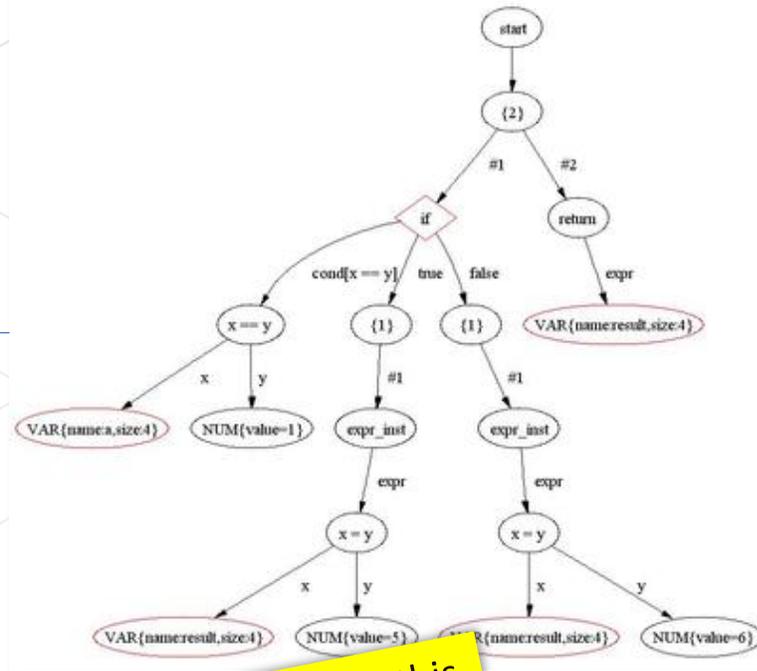
# What is the problem?

# Q&D Compilation concepts

- Compiling source -> binary is a **lossy** process
- The info we lose is the same information source code analysis requires
  - Structure/object sizes, members
  - Code flow harder to track
  - Relationships between data structures
- We need to do some processing to try to recover this lost information
  - Once we have it our binary analysis is close to our source code analysis

# Q&D Decompilation concepts

```
int func(int a)
{
    int result;
    if ( a == 1 )
    {
        result = 5;
    }
    else
    {
        result = 6;
    }
    return result;
}
```



IDA uses this  
A LOT

```
mov    eax, [ebx+0x04]
add    eax, [ebx+0x08]
sub    [ebx+0x0C], eax
```

$m[ebx+12] := m[ebx+12] - (m[ebx+4] + m[ebx+8]);$





# More on (de)compilation

<https://llvm.org/docs/tutorial/MyFirstLanguageFrontend/index.html>



# IDA Decompiler



```

28  _DWORD cmd_string[8]; // [rsp+60h] [rbp-1868h] BYREF
29  _DWORD get_results[34]; // [rsp+80h] [rbp-1848h] BYREF
30  char v27; // [rsp+108h] [rbp-17C0h]
31  struct _PROCESS_INFORMATION v28[10]; // [rsp+180h] [rbp-1748h] BYREF
32  struct _STARTUPINFOA v29[2]; // [rsp+280h] [rbp-1648h] BYREF
33  char http_contents[256]; // [rsp+380h] [rbp-1548h] BYREF
34  char _subcommand[1024]; // [rsp+480h] [rbp-1448h] BYREF
35  char v32[4136]; // [rsp+880h] [rbp-1048h] BYREF
36
37  (sub_403600)(lpThreadParameter);
38  do
39  {
40      while ( 1 )
41      {
42          while ( 1 )
43          {
44              while ( 1 )
45              {
46                  while ( 1 )
47                  {
48                      do
49                      {
50                          Sleep(dwMilliseconds);
51                          cgp_send_http_msg_wrap_GET(get_results, "45.66.250.110", "80", g_buffer_hello_uri);
52                          cmd_get_result = strtok(get_results, "&");
53                      }
54                      while ( !cmd_get_result );
55                      strcpy(cmd_string, cmd_get_result);
56                      subcommand = strtok(0i64, "&");
57                      if ( subcommand )
58                          strcpy(_subcommand, subcommand);
59                      if ( strcmp(cmd_string, "sleep") )
60                          break;
61                      delay_timer = atoi(_subcommand);
62                      if ( delay_timer > 100 )
63                          dwMilliseconds = delay_timer;
64                  }

```

```

1  |
2  void FUN_0040060c(void)
3
4  {
5      uint uVar1;
6      int iVar2;
7      ulong uVar3;
8      int iVar4;
9      bool bVar5;
10     uint local_18;
11     int local_14;
12
13     write(1,"Welcome to packedup for r2crackmes :) \nFlag << ",0x30);
14     read(0,&DAT_00601080,0x2c);
15     iVar2 = 0x400614;
16     iVar4 = 0xe2;
17     uVar3 = 0;
18     do {
19         uVar1 = (uint)(byte){(char)uVar3 + *(char *){local_18}};
20         local_18 = ((uint)uVar3 & 0xfffff00 | uVar1) >> 4 | uVar1 << 0x1;
21         uVar3 = (ulong)local_18;
22         iVar2 = iVar2 + 1;
23         iVar4 = iVar4 + -1;
24     } while (iVar4 != 0);
25     local_14 = 0x2c;
26     do {
27         bVar5 = (int)local_18 < 0;
28         uVar1 = local_18 << 1;
29         local_18 = uVar1 | (uint)bVar5;
30         if ((uVar1 & 0xff | (uint)bVar5) !=
31             (uint)(byte){(JUNK_004007a0){(long){local_14 + -1}} ^ (DAT_
32                 )}) {
33             write(1,"Try again!\n",0xd);
34             goto LAB_004006f6;
35         }
36         local_14 = local_14 + -1;
37     } while (local_14 != 0);
38     write(1,"Yep! you got the flag :) \n",0x1c);
39 LAB_004006f6:
40         /* WARNING: Subroutine does not return */
41     exit(0);

```



**Ain't nobody got time for [Tab]**

# Enter FIDL

fireeye / FIDL Watch

Code Issues 1 Pull requests 0 Actions Projects 0 Security 0 Insights

A sane API for IDA Pro's decompiler. Useful for malware RE and vulnerability research <https://fidl.readthedocs.io>

ida decompiler api vulnerability research reversing malware

29 commits 1 branch 0 packages 3 releases

Branch: master New pull request

Carl OS	Updated offline documentation and Sphinx config
FIDL	Updated offline documentation and Sphinx config
.gitignore	Initial commit
.readthedocs.yml	Updated readthedocs info
LICENSE	Added MIT license
MANIFEST.in	Initial commit
README.md	README shields
README.rst	Initial commit
requirements.txt	RTD fix
setup.py	Bumped version

Ifak Guilfanov @ilfak

Hey, our API is usable but I get your point.

m0n0sapiens @m0n0sapiens · Nov 25, 2019

What? A library that wraps IDA decompiler API and makes it usable? \*and\* documented?

We just released a thing.

Blog: [fireeye.com/blog/threat-re...](https://fireeye.com/blog/threat-re...)

Github: [github.com/fireeye/FIDL](https://github.com/fireeye/FIDL)

Docs!!!: [fidl.readthedocs.io/en/latest/](https://fidl.readthedocs.io/en/latest/)

12:54 AM · Nov 26, 2019 · Twitter Web App

Rolf Rolles @RolfRolles

Replying to @m0n0sapiens and @bdcht

The phrasing here is kind of inflammatory. The Hex-Rays API has a learning curve, but the documentation has increased substantially lately (see [hexrays.com/products/decom...](https://hexrays.com/products/decom...)) and I find it plenty usable

11:15 AM · Nov 26, 2019 · Twitter Web App

Ifak Guilfanov @ilfak · Dec 3, 2019

This is wonderful!

m0n0sapiens @m0n0sapiens · Dec 2, 2019

Thanks to @marc\_etienne\_ and other fine folks from ESET, #FIDL works in IDA 7.4 and Python 3!

[github.com/fireeye/FIDL](https://github.com/fireeye/FIDL)

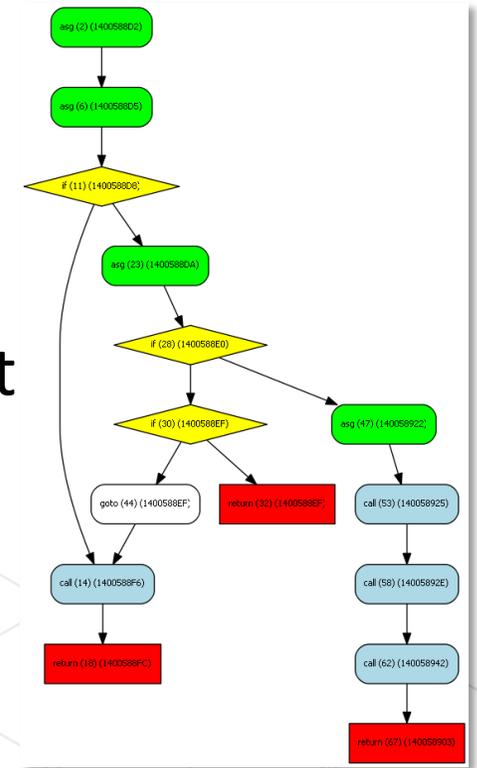
Show this thread



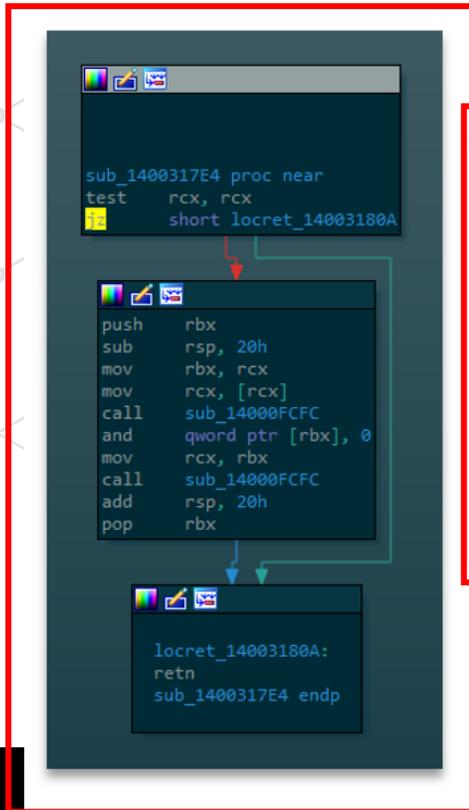
# All hail controlFlowinator

- Data structure representing **one function**
- Mix between CFG and decompilation
- Picture a CFG where every node is a high level code const
  - if
  - assignment
  - function call
  - return
  - etc.

CFG: Control Flow Graph

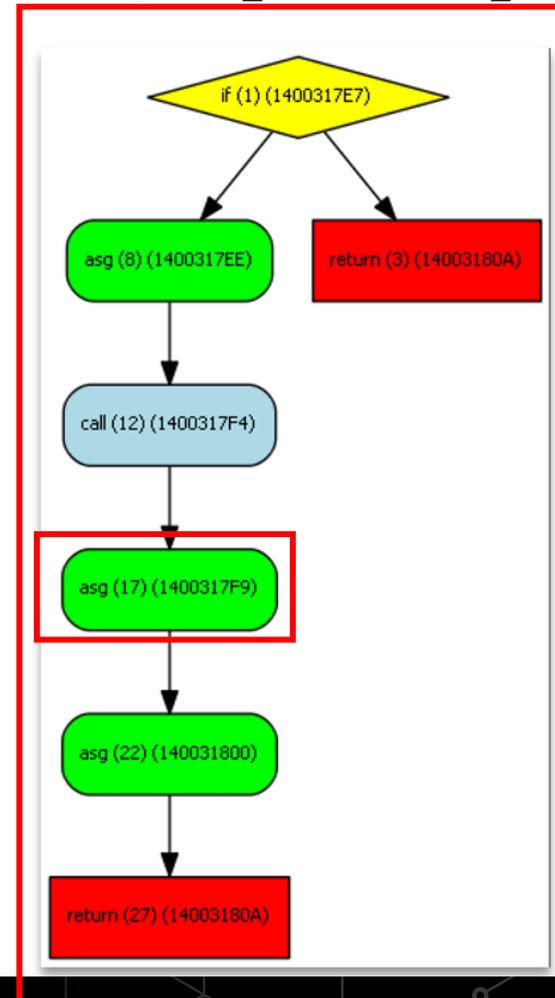


# FLARE IDA Decompiler Library (FIDL)

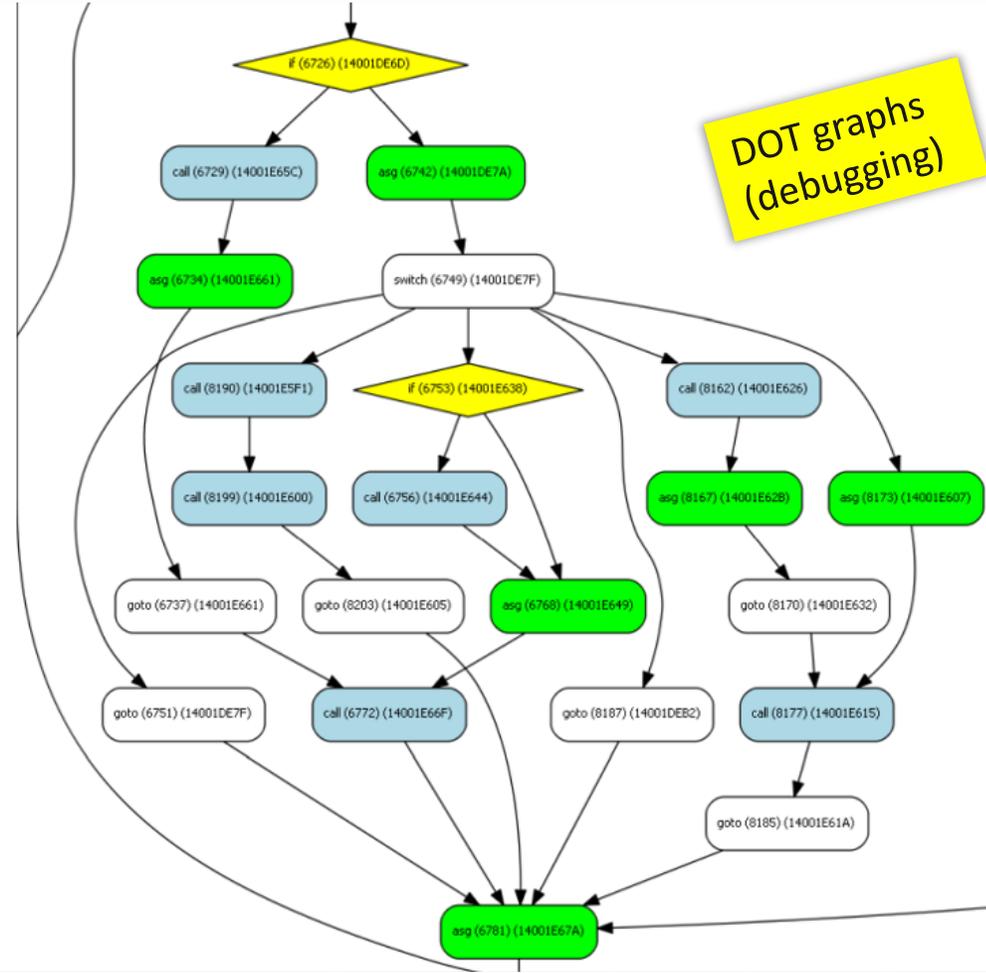
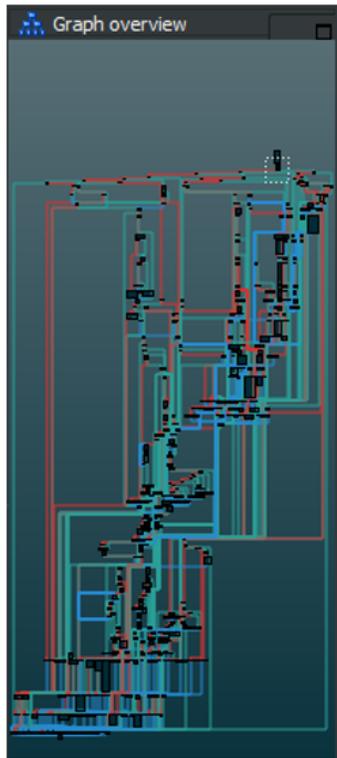


```

1 __int64 __fastcall sub_1400317E4(void **a1)
2 {
3     void **v1; // rbx
4     __int64 result; // rax
5
6     if ( !a1 )
7         return result;
8     v1 = a1;
9     sub_14000FCFC(*a1);
10    *v1 = 0i64;
11    result = sub_14000FCFC(v1);
12    return result;
13 }
  
```

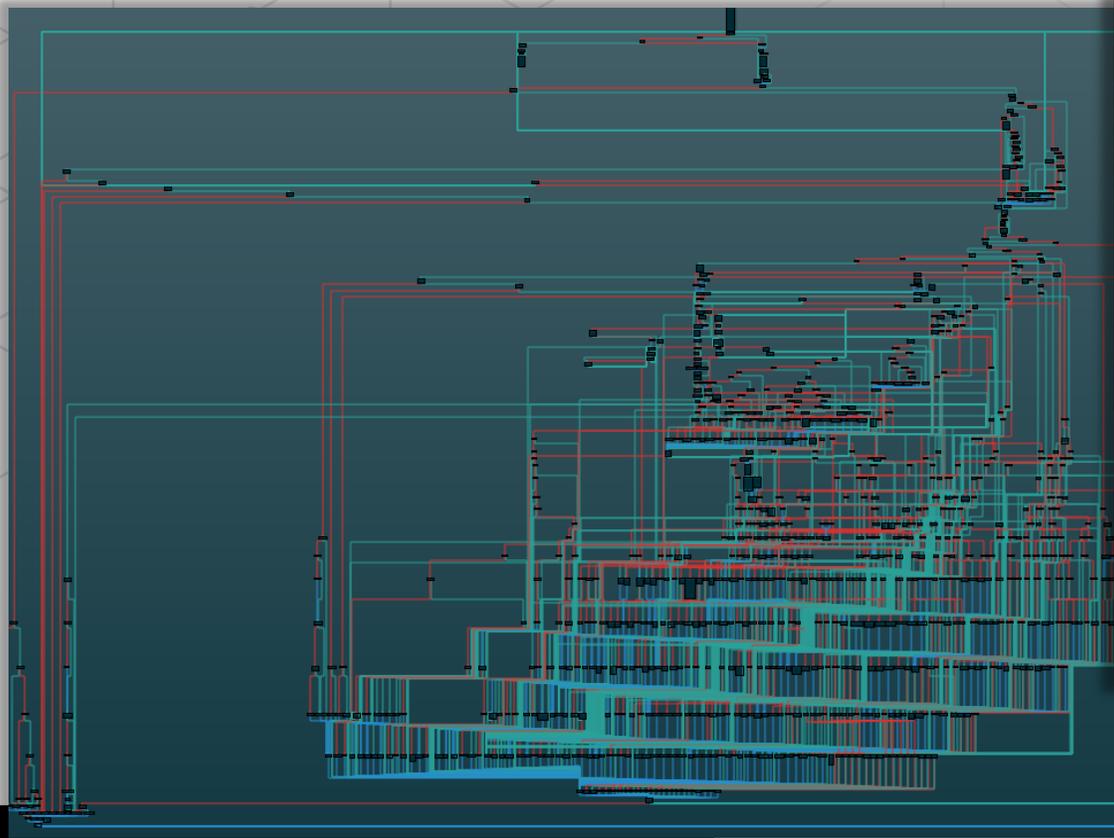


# FLARE IDA Decompiler Library (FIDL)



DOT graphs (debugging)

# DOT can't keep up :')



```
C:\WINDOWS\system32\cmd.exe

(dot.exe:16284): Pango-WARNING **: couldn't load font "Sans serif Not-
back to "Sans Not-Rotated 8", expect ugly output.
dot: failure to create cairo surface: out of memory
```

Choose Just-In-Time Debugger

An unhandled win32 exception occurred in dot.exe [16284].

Available Debuggers:

- New instance of Visual Studio Community 2017

```
Python>c
< _main_ .controlFlowinator instance at 0x0000022B6628B148>
Python c = controlFlowinator(ea=here(), fast=False)
```

# Batteries included

- It contains several interesting information by default
  - Function calls
  - Local variables
  - Arguments
  - Return type

```

1 Python>import FIDL.decompiler_utils as du
2 Python>c = du.controlFlowinator(ea=here(), fast=False)
3 Python>c
4 <FIDL.decompiler_utils.controlFlowinator instance at 0x000001D756DB21C8>

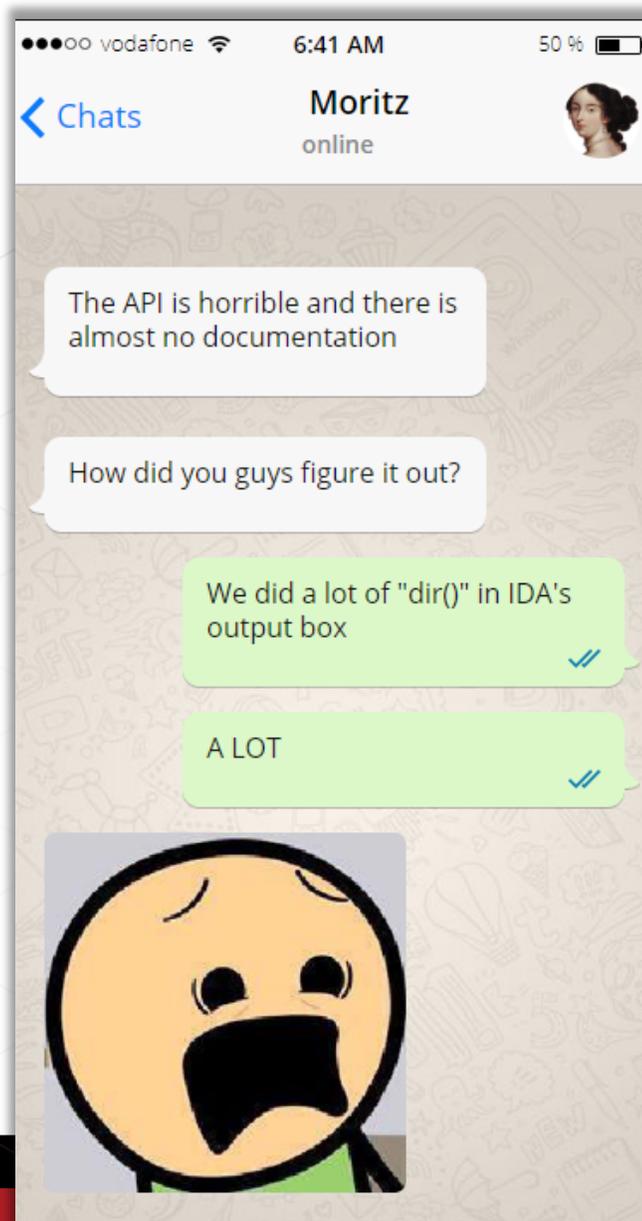
```

```

1 Python>c.lvars
2 Name: v4
3   Type name: __int64
4   Size: 8
5 Name: v5
6   Type name: const __m128i *
7   Size: 8
8 Complex type: __m128i
9 Pointed object: const __m128i
10 <snip...>
11 Name: WideCharStr
12   Type name: __int16[256]
13   Size: 512
14 Array type: __int16

```

# Diggers of APIs



**BUT  
WHY?!?**



# I just want to do a simple thing

- Find a specific *function call* within a function
- Search for any *argument* that is a string
- Get that ASCII string
- Sounds easy right?

```
SetWindowTextA(v3, v7);
sub_14000FCFC(v7);
SetDlgItemTextA(
    v3,
    1002,
    "PuTTY is copyright 1997-2017 Simon Tatham.\r\n"
    "\r\n"
    "Portions copyright Robert de Bath, Joris van Rantwijk,
    "Nicolas Barry, Justin Bradford, Ben Harris, Malcolm Smit
    "ite, and CORE SDI S.A.\r\n"
    "\r\n"
    "Permission is hereby granted, free of charge, to any pe
    "tation files (the \"Software\"), to deal in the Softwar
    "s to use, copy, modify, merge, publish, distribute, sub
    "rsons to whom the Software is furnished to do so, subje
    "\r\n"
    "The above copyright notice and this permission notice s
    " Software.\r\n"
    "\r\n"
    "THE SOFTWARE IS PROVIDED \"AS IS\", WITHOUT WARRANTY OF
    "HE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICU
    "IGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
    "E, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTW
return li64;
```

# IDA's idea of easy ;)

```
Output window
Hex-rays version 7.2.0.181105 has been detected
Decompida v.0.5 "nerdilicious" loaded...

Python>c = controlFlowinator(ea=here(), fast=False)
Python>for co in c.calls:
    if co.name == 'SetDlgItemTextA':
        for arg in co.args.values():
            if arg.type == 'string':
                print arg.val

Python>
PuTTY is copyright 1997-2017 Simon St Laurent

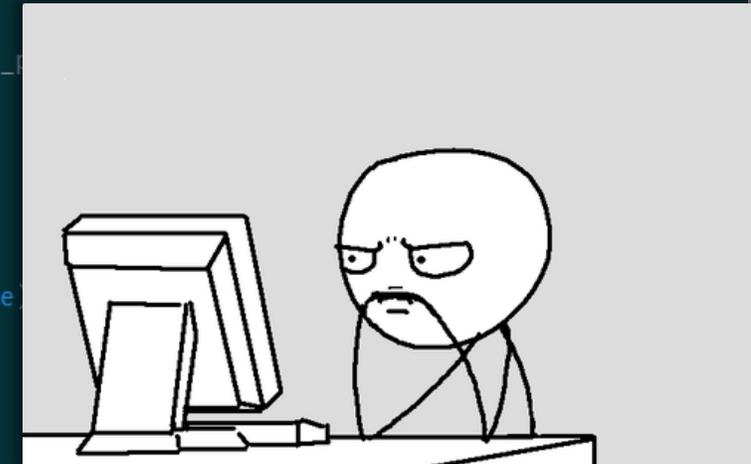
Portions copyright Robert de Batho, Joris van Rantwijk, Delian
Delchev, Andreas Schultz, Jeroen Massar, Wez Furlong, Nicolas
Barry, Justin Bradford, Ben Harris, Malcolm Smith, Ahmad
Khalifa, Markus Kuhn, Colin Watson, Christopher Staite, and
CORE SDI S.A.

Permission is hereby granted, free of charge, to any person
obtaining a copy of this software and associated documentation
```



Named  
tuples!!!

```
1 import idc
2 import idaapi
3
4 import ida_hexrays
5
6
7 class visitor(idaapi.ctree_visitor_t):
8     def __init__(self, cfunc, result):
9         idaapi.ctree_visitor_t.__init__(self, idaapi.CV_FAST)
10        self.cfunc = cfunc
11        self.result = result
12
13    def visit_expr(self, i):
14        if i.op == idaapi.cot_call:
15            # look for calls to GetProcAddress
16            if idc.Name(i.x.obj_ea) == "SetDlgItemTextA":
17
18                for idx in xrange(len(i.a)):
19                    # ASCSTR_C = 0
20                    # Check to see if the second argument is a C string
21                    if idc.GetStringType(i.a[idx].obj_ea) == 0:
22                        da_string = idc.GetString(i.a[idx].obj_ea, -1, 0)
23                        self.result.add(da_string)
24
25        return 0
26
27
28    def main():
29        if not idaapi.init_hexrays_...
30            return False
31
32        result = set([])
33
34        cfunc = decompile(here())
35        v = visitor(cfunc, result)
36        v.apply_to(cfunc.body, None)
37
38        print result
39
40
41 if __name__ == "__main__":
42     main()
```





# Much moar

```
def main():
    STRINGS_ = decrypt_strings()
    calls = du.find_all_calls_to(f_name='t_string_decryption')

    for co in calls:
        if co.args and co.args[0].type == 'number':
            temp = STRINGS_[co.args[0].val:]
            temp = temp.split('\x00',1)[0]
            du.create_comment(co.c,co.ea,'%x %s' % (co.args[0].val,temp))
```

# Much moar

- Interacting with the decompiler view (GUI)
  - display\_node
  - display\_path
  - display\_line\_at
  - **display\_all\_calls\_to**

Direction	Type	Address	Text
	p	sub_180168716+D5	call cgp_mesa_debug_logging
Do...	p	sub_1801687F9+D9	call cgp_mesa_debug_logging
Do...	p	sub_1801688E0+F4	call cgp_mesa_debug_logging
Do...	p	sub_1801689E2+107	call cgp_mesa_debug_logging
Do...	p	sub_180168AF7+11B	call cgp_mesa_debug_logging

```

Output window
- 1801680BB 46:
- 1801682AB 46:
- 1801684B5 47:
- 1801686C7 47:
- 1801687EB 33: cgp_mesa_debug_logging((__int64)v4, 0x501u, "vbo_TexCoordP1ui");
- 1801688D2 33: cgp_mesa_debug_logging((__int64)v4, 0x501u, "vbo_TexCoordP1uiv");
- 1801689D4 39: cgp_mesa_debug_logging((__int64)v4, 0x501u, "vbo_TexCoordP2ui");
- 180168AE9 38: cgp_mesa_debug_logging((__int64)v4, 0x501u, "vbo_TexCoordP2uiv");
- 180168C12 44: cgp_mesa_debug_logging((__int64)v4, 0x501u, "vbo_TexCoordP3ui");
- 180168D52 40: cgp_mesa_debug_logging((__int64)v4, 0x501u, "vbo_TexCoordP3uiv");
- 180168E9E 46: cgp_mesa_debug_logging((__int64)v4, 0x501u, "vbo_TexCoordP4ui");
- 180169004 42: cgp_mesa_debug_logging((__int64)v4, 0x501u, "vbo_TexCoordP4uiv");
- 1801690F0 35: cgp_mesa_debug_logging((__int64)v6, 0x501u, "vbo_MultiTexCoordP1ui");
- 1801691E0 35: cgp_mesa_debug_logging((__int64)v6, 0x501u, "vbo_MultiTexCoordP1uiv");
- 1801692EC 41: cgp_mesa_debug_logging((__int64)v6, 0x501u, "vbo_MultiTexCoordP2ui");
- 18016940B 40: cgp_mesa_debug_logging((__int64)v6, 0x501u, "vbo_MultiTexCoordP2uiv");
- 18016953F 46: cgp_mesa_debug_logging((__int64)v6, 0x501u, "vbo_MultiTexCoordP3ui");
- 180169689 42: cgp_mesa_debug_logging((__int64)v6, 0x501u, "vbo_MultiTexCoordP3uiv");
- 1801697E0 48: cgp_mesa_debug_logging((__int64)v6, 0x501u, "vbo_MultiTexCoordP4ui");
- 180169950 44: cgp_mesa_debug_logging((__int64)v6, 0x501u, "vbo_MultiTexCoordP4uiv");
- 180169ADC 40: cgp_mesa_debug_logging((__int64)v4, 0x501u, "vbo_NormalP3ui");
- 180169C72 40: cgp_mesa_debug_logging((__int64)v4, 0x501u, "vbo_NormalP3uiv");
- 180169DFD 40: cgp_mesa_debug_logging((__int64)v4, 0x501u, "vbo_ColorP3ui");

Python du.display_all_calls_to("cgp_mesa_debug_logging")
  
```

# Much moar

- Much of the basic instrumentation is built-in
  - specifically the tedious stuff
- FIDL implements **generic functionality**
  - building blocks
- You can focus on higher-level logic
  - heuristics for vulnerability research, malware analysis, etc.
  - the fun stuff 😊





# A more complete example

<https://fidl.readthedocs.io/en/latest/tutorial.html#a-more-complete-example>

A photograph of an industrial robotic welding station. Two robotic arms are positioned to weld a metal component. Bright orange sparks are flying from the welding point, illuminating the scene. The background shows a factory environment with other machinery and structural elements. A yellow rectangular box is overlaid on the center of the image, containing the text "Scale it up!".

Scale it up!

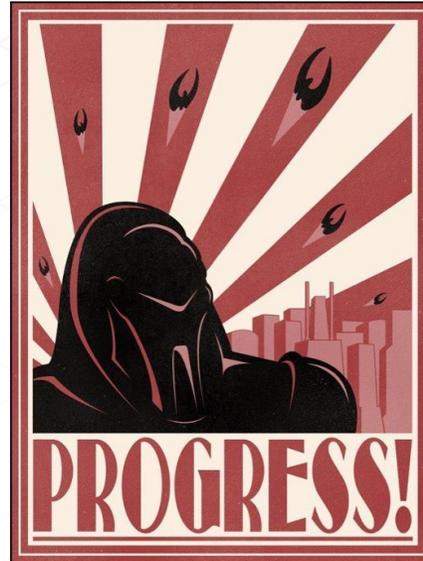
# Scaling up <sup>beta</sup>

- Initially, our workflow
  - open IDA
  - load binary
  - execute FIDL script
  - close IDA
  - repeat



# Scaling up <sup>beta</sup>

- The Citadel
  - Web interface
  - Drag & Drop
  - Distributed processing
  - IDA Headless execution
  - Dockerz!





RICK AND MORTY  
**TALES FROM**  
*the* **CITADEL**





# The Citadel::Home

The Citadel [Home](#) [Upload Campaign](#) [Files](#) [Archives](#) [Danger Zone](#) [ Logged in as: car\_Los <carlos.g.prado@gmail.com> ] [Logout](#)

## Index

Campaign	Progress	Product Name	Product Version	Date	Actions
Dummy Test [files]	<div style="width: 100%;"><div style="width: 100%;"></div></div>	Nothing to see here	1	2020-05-08 16:11:55.773515	<a href="#">DELETE</a> <a href="#">ARCHIVE</a> <a href="#">RE-RUN</a> <a href="#">CLEAN-UP</a>

(C) The Citadel 2019



# The Citadel::Upload

The Citadel [Home](#) [Upload Campaign](#) [Files](#) [Archives](#) [Danger Zone](#) [Logged in as: carl\_os <carlos.g.prado@gmail.com>] [Logout](#)

## Upload Campaign

Campaign options

Campaign Name  
That's Offensive!!!

Product Name  
OTF

Product Version  
13.37

Description  
Oh boi, here I go again

**Upload files**

Drop folder here!

# The Citadel::Upload

All files uploaded!

## Analysis Workflow

- [drag me](#) Attack surface Analysis
- [drag me](#) Decompiler-based bug hunting tools
- [drag me](#) Assembly-based bug hunting tools
- [drag me](#) Perform a dummy test. Debugging purposes.
- [drag me](#) User defined 1
- [drag me](#) User defined 2
- [drag me](#) User defined 2

- [drag me](#) Attack surface Analysis
- [drag me](#) Decompiler-based bug hunting tools

Create Campaign!





# The Citadel::New Campaign

## Index

Campaign	Progress	Product Name	Product Version	Date	Actions
<a href="#">Dummy Test [files]</a>	<div style="width: 100%;"><div style="background-color: purple; height: 10px;"></div></div>	Nothing to see here	1	2020-05-08 16:11:55.773515	<a href="#">DELETE</a> <a href="#">ARCHIVE</a> <a href="#">RE-RUN</a> <a href="#">CLEAN-UP</a>
<a href="#">That's Offensive!!! [files]</a>	<div style="width: 25%;"><div style="background-color: gray; height: 10px;"></div></div>	OTF	13.37	2020-05-28 15:11:26.977844	<a href="#">DELETE</a> <a href="#">ARCHIVE</a> <a href="#">RE-RUN</a> <a href="#">CLEAN-UP</a>



# The Citadel::Results

The Citadel [Home](#) [Upload Campaign](#) [Files](#) [Archives](#) [Danger Zone](#) [Logged in as: carl\_os <carlos.g.prado@gmail.com>] [Logout](#)

## Campaign Results

- Campaign Name: **Dummy Test**
- Product: Nothing to see here
- Description: Oh boi, here I go again
- Date: 2020-05-08 16:11:55.773515
- Status: *complete* [[mark as complete](#)]

pscp.exe

TOGGLE REVIEWED

Filename	Bug Class	Location	Bug Notes
<a href="#">pscp.exe.idb</a>	Complexity	0x413784	sub_413784: 231.00
<a href="#">pscp.exe.idb</a>	Complexity	0x434E80	__input: 180.00
<a href="#">pscp.exe.idb</a>	Complexity	0x416193	sub_416193: 178.00
<a href="#">pscp.exe.idb</a>	Complexity	0x40F430	sub_40F430: 171.00
<a href="#">pscp.exe.idb</a>	Complexity	0x40114C	sub_40114C: 160.00
<a href="#">pscp.exe.idb</a>	Complexity	0x43365D	__output: 114.00
<a href="#">pscp.exe.idb</a>	Complexity	0x438AC4	__strgtold12: 80.00
<a href="#">pscp.exe.idb</a>	Complexity	0x42FA3A	__store_winword: 71.00
<a href="#">pscp.exe.idb</a>	Complexity	0x410DA7	sub_410DA7: 60.00
<a href="#">pscp.exe.idb</a>	Complexity	0x427C01	sub_427C01: 59.00



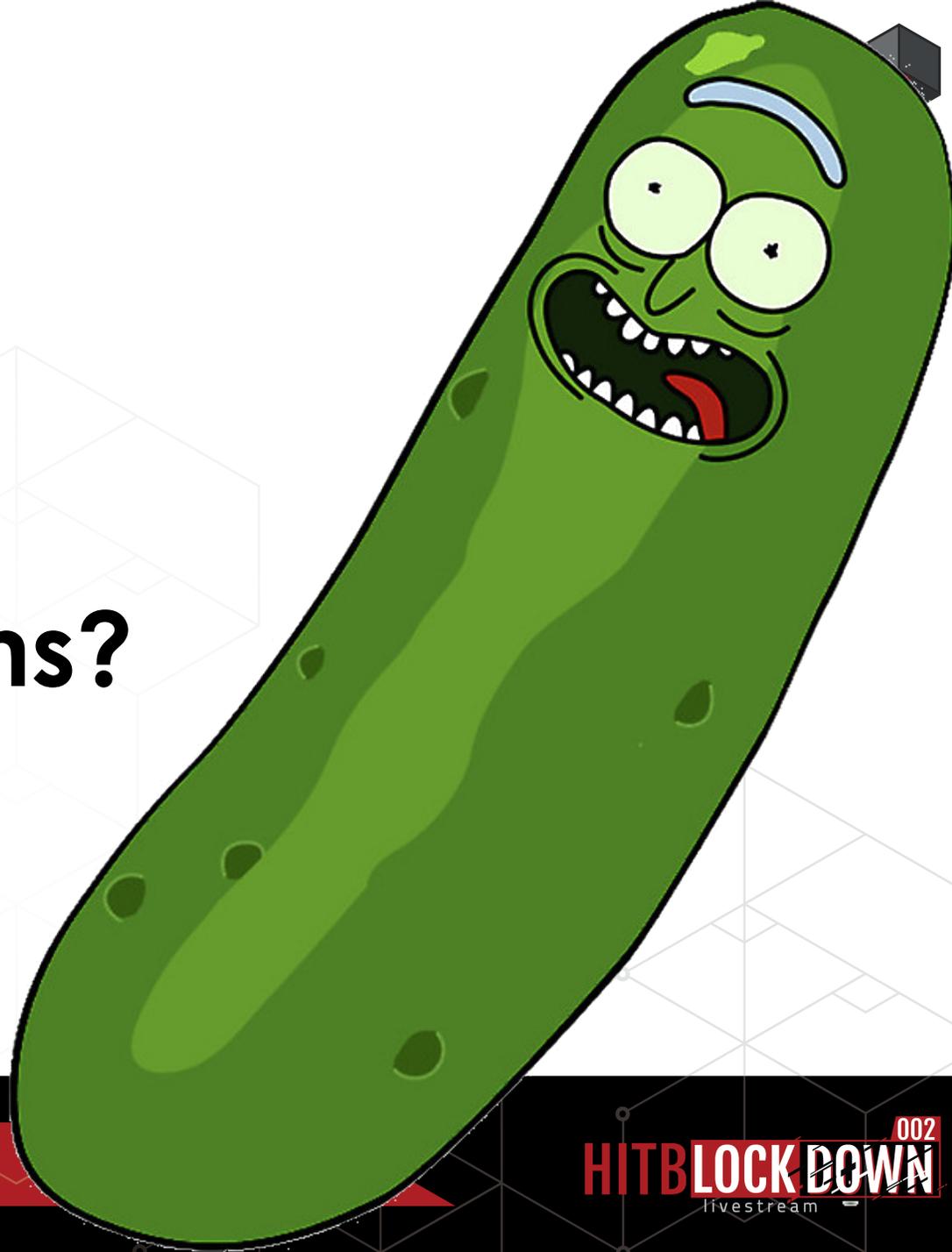
# Where do I get it?

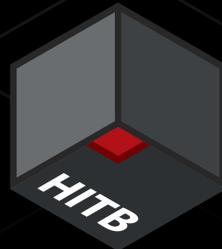
- <https://github.com/fireeye/FIDL>
- <https://fidl.readthedocs.io/en/latest/>
  - Online documentation! (*automatically updated*)

# Decompilation is the future. I've seen it with my eyes...

```
0022 // DWORD __stdcall GetFileSize(HANDLE hFile, LPDWORD lpFileSizeHigh);
0023 // BOOL __stdcall GetFileAttributesExW(LPCWSTR lpFileName, GET_FILEEX_INFO_LEVELS fInfoLevelId, LPVOID lpFileInfom
0024 HMODULE __stdcall GetModuleHandleW(LPCWSTR lpModuleName);
0025 // BOOL __stdcall SetFileTime(HANDLE hFile, const FILETIME *lpCreationTime, const FILETIME *lpLastAccessTime, const FI
0026 // BOOL __stdcall WriteFile(HANDLE hFile, LPCVOID lpBuffer, DWORD nNumberOfBytesToWrite, LPDWORD lpNumberOfBy
0027 lpOverlapped);
0028 // DWORD __stdcall GetFileAttributesW(LPCWSTR lpFileName);
0029 // HANDLE __stdcall CreateFileW(LPCWSTR lpFileName, DWORD dwDesiredAccess, DWORD dwShareMode, LPSECURITY
0030 lpSecurityAttributes, DWORD dwCreationDisposition, DWORD dwFlagsAndAttributes, HANDLE hTemplateFile);
0031 // int __stdcall GetProcAddress(HMODULE hModule, LPCSTR lpProcName);
0032 FARPROC __stdcall GetProcAddress(HMODULE hModule, LPCSTR lpProcName);
0033 // BOOL __stdcall CloseHandle(HANDLE hObject);
0034 // BOOL __stdcall DeleteFileW(LPCWSTR lpFileName);
0035 // LPWSTR __stdcall GetTempPathW(LPCWSTR lpString1, LPCWSTR lpString2);
0036 // BOOL __stdcall SetFileAttributesW(LPCWSTR lpFileName, DWORD dwFileAttributes);
0037 // HANDLE __stdcall GetCurrentProcess()
v5 = GetModuleHandleW(L"NTDLL.dll");
v4 = GetProcAddress(v5, "ZwGetInformationFile");
if (!v4)
{
    v7 = "L_DWORD *%a1 + 4);
    v13 = "L_DWORD *%a1,
    v8 = "L_DWORD *%a2,
    v14 = v7;
    v9 = "L_DWORD *%a2 + 4);
    (result = ((int (__stdcall *)(void *, char *, int *, signed int, signed int))(v4))(a4, &v12, &v13, 40, 4)) != 0)
    result = SetFileTime(a4, (const FILETIME *%a1, (const FILETIME *%a2, (const FILETIME *%a3);
    return result;
0038 // void __stdcall SetLastError(DWORD dwErrCode);
0039 // int __stdcall GetTempPathW(LPCWSTR lpString1, LPCWSTR lpString2);
0040 // DWORD __stdcall GetCurrentThreadId();
0041 // DWORD __stdcall GetCurrentThreadId();
```

**Questions?**





# Thank You!

**HITB** **LOCKDOWN** <sup>002</sup>  
livestream

Ryan Warns, Carlos Garcia Prado <{ryan.warns, carlos.garcia}@fireeye.com>