Hijacking client software for fuzz and profit

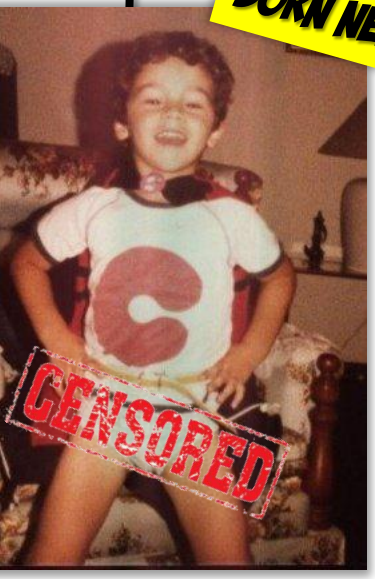
Overview

- Introduction.
- Fuzzing 101. *yawn*
- The need for a different approach.
  - Abusing the client.
- A possible implementation. Boyka.
- Experiment.
- Conclusion.

$ whoami

PARTICLE PHYSICIST

BORN NERD



PROSPECTIVE EMPLOYERS

F**K THAT SH*T

$ whoami


LICHTBILDAUSWEIS
IDENTITY CARD / CARTE IDENTITÉ
OF NORWAY
PRIME MINISTER
NORWEGEN
28.12.2017

Let's be clear

This work is shit

LET'S BE CLEAR

THIS WORK IS 

"ESSENTIALLY FLAWED"

What this is about

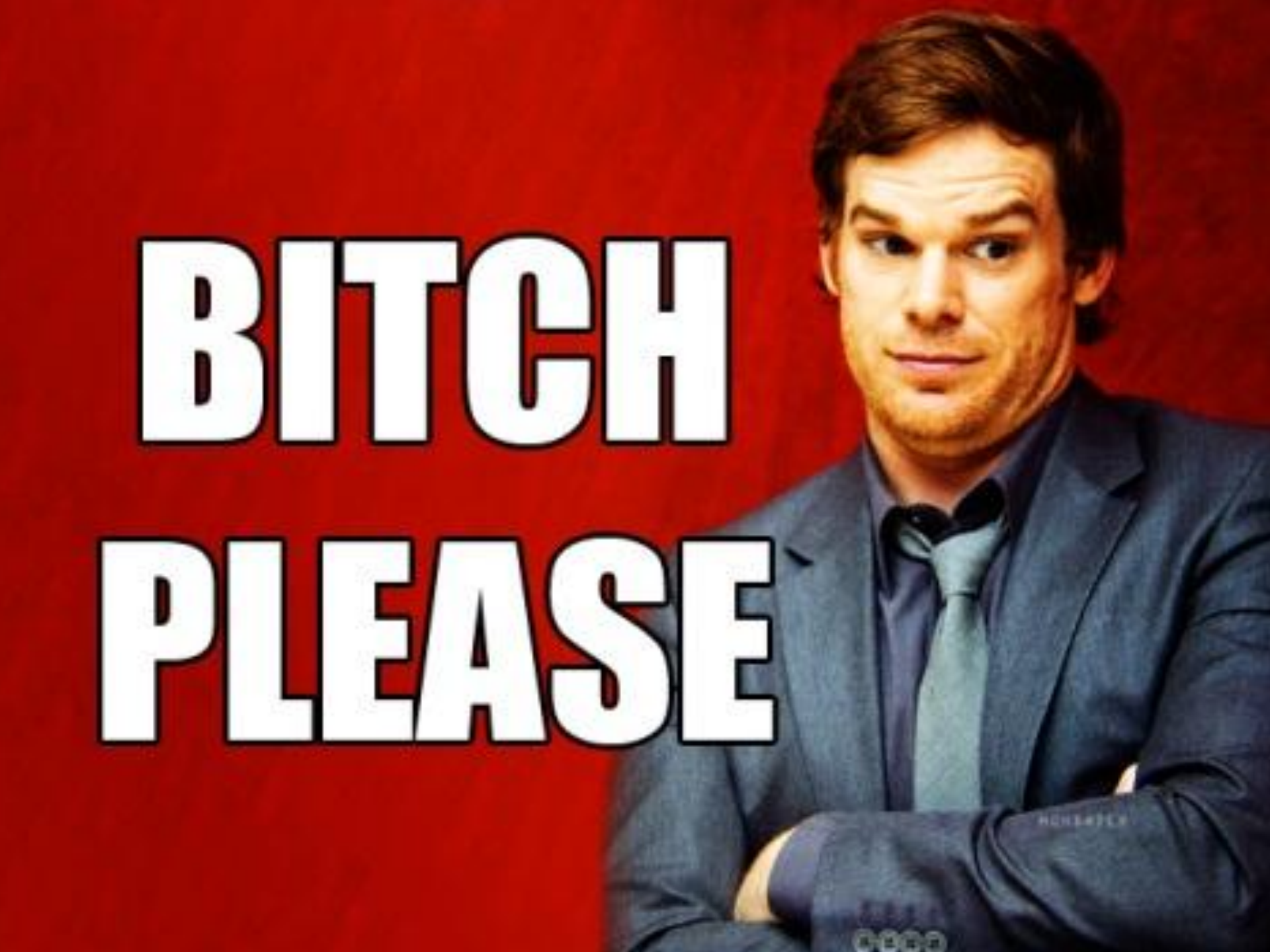
- Interesting approach to software testing
- Touching things you are not supposed to
- Breaking stuff (if you're lucky!)
- Multiple references to pop culture
  - and chocolate!

Everybody Luvs Chocolate

Look for the ***ChocoQuiz*** Icon

DISCLAIMER

Fuzzing 101


BITCH
PLEASE

Fuzzing is like violence:
if it doesn‘t solve your problems,
you are not using enough of it.

stolen from the nokogiri guys 鋸


BRUTE FORCE

MONSTER CONFIGURATION

```python
from sulley import *

s_initialize("user")
s_static("USER")
s_delim(" ")
s_string("yomama")
s_static("\r\n")

s_initialize("pass")
s_static("PASS")
s_delim(" ")
s_string("issofat")
s_static("\r\n")

s_initialize("cwd")
s_static("CWD")
s_delim(" ")
s_string("c: ")
s_static("\r\n")

s_initialize("dele")
s_static("DELE")
s_delim(" ")
s_string("c:\\test.txt")
s_static("\r\n")
```

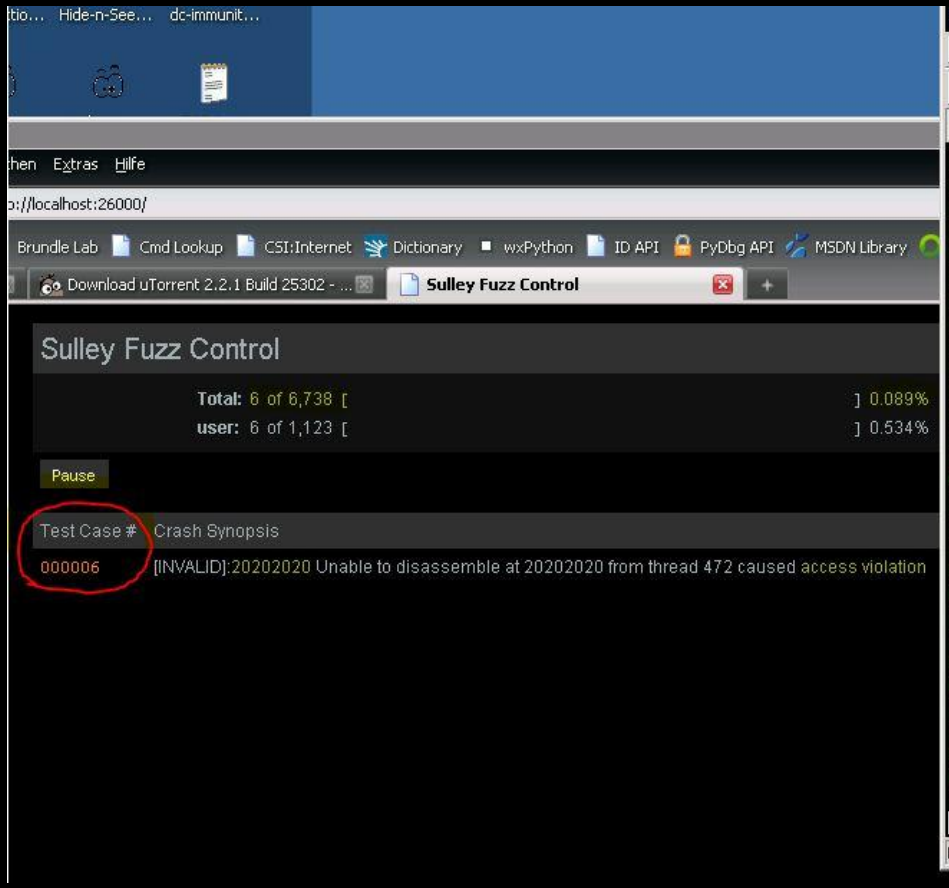
ftp.py - protocol

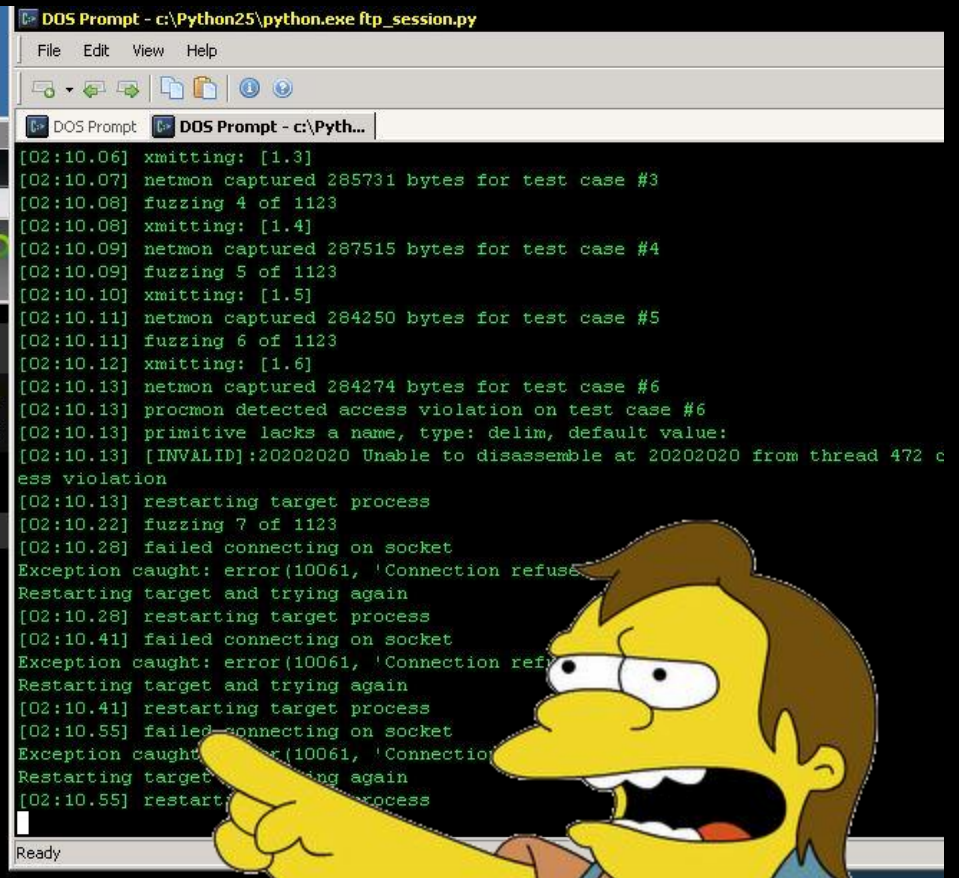
```python
from sulley import *
from requests import ftp # this is our ftp.py file

sess           = sessions.session(session_filename="audits/freefloatftp.session")
target         = sessions.target("192.168.1.11", 21)
target.netmon  = pedrpc.client("192.168.1.11", 26001) # NetMonitor (packets)
target.procmon = pedrpc.client("192.168.1.11", 26002) # ProcMonitor (crashes)
target.procmon_options = { "proc_name" : "FTPServer.exe" }


sess.add_target(target)


sess.connect(s_get("user"))
sess.connect(s_get("user"), s_get("pass"))

sess.connect(s_get("pass"), s_get("cwd"))
sess.connect(s_get("pass"), s_get("dele"))
sess.connect(s_get("pass"), s_get("mdtm"))
sess.connect(s_get("pass"), s_get("mkd"))

sess.fuzz()
```

ftp_session.py


And Now
PITCH
WASTE
I Wait
ICANHASCHEEZBURGER.COM

CRASH! BOOM! BANG! HAHA!





PRECISE CRASH INFORMATION

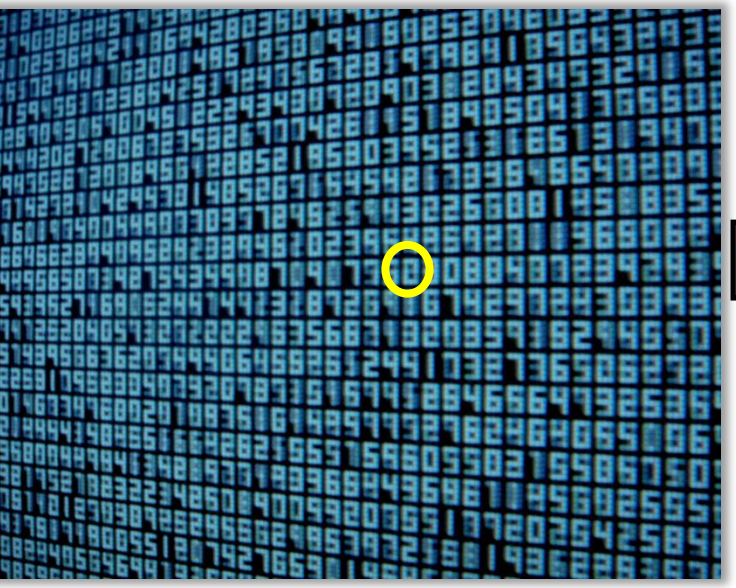
http://localhost:26000/view_crash/6

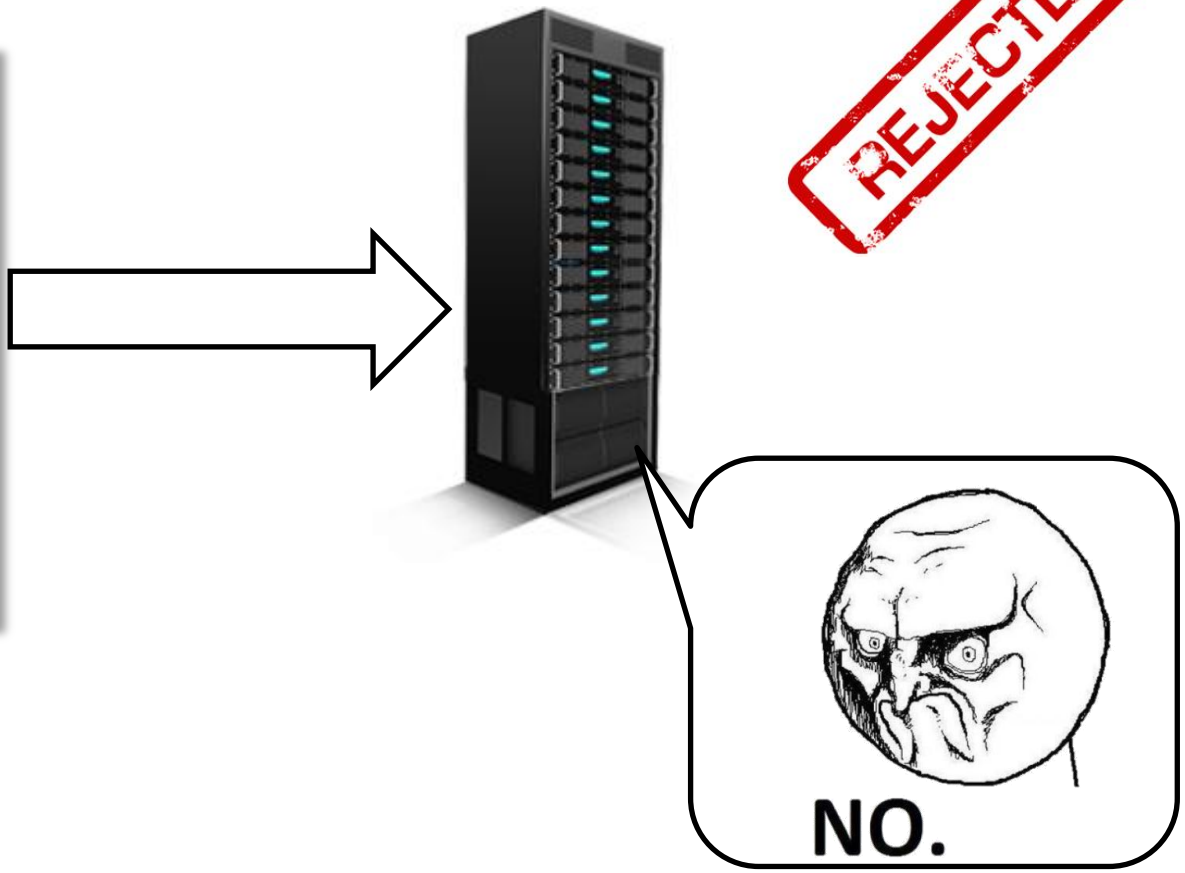
```
[INVALID]:20202020 Unable to disassemble at 20202020 from thread 472 caused access violation
when attempting to read from 0x20202020

CONTEXT DUMP
  EIP: 20202020 Unable to disassemble at 20202020
  EAX: 00000216 (       534) -> N/A
  EBX: 00000002 (         2) -> N/A
  ECX: 0014d3c0 (   1364928) -> F unt authority\systemzA (heap)
  EDX: 7c90e514 (2089870612) -> N/A
  EDI: 003b19d5 (   3873237) ->  (heap)
  ESI: 0040a44e (   4236366) -> N/A
  EBP: 003b1298 (   3871384) -> N/A
  ESP: 00b2fc2c (  11729964) ->
  +00: 20202020 ( 538976288) -> N/A
  +04: 20202020 ( 538976288) -> N/A
  +08: 20202020 ( 538976288) -> N/A
  +0c: 20202020 ( 538976288) -> N/A
  +10: 20202020 ( 538976288) -> N/A
  +14: 20202020 ( 538976288) -> N/A

disasm around:
        0x20202020 Unable to disassemble

SEH unwind:
        ffffffff -> kernel32.dll:7c839ad8 push ebp
```

There's always a but


There is NO documentation at all. :((((((

I CAN ALWAYS TRY DUMB FUZZING!





Think about checksums…

Checksum = SHA1(Data)
SHA1: 160 bits
$P(right) = 1/2^{160} \approx 1/10^{48}$
10^{48} = 1k·1T·1T·1T·1T·1T

Dumb Fuzzing…


F*CK!

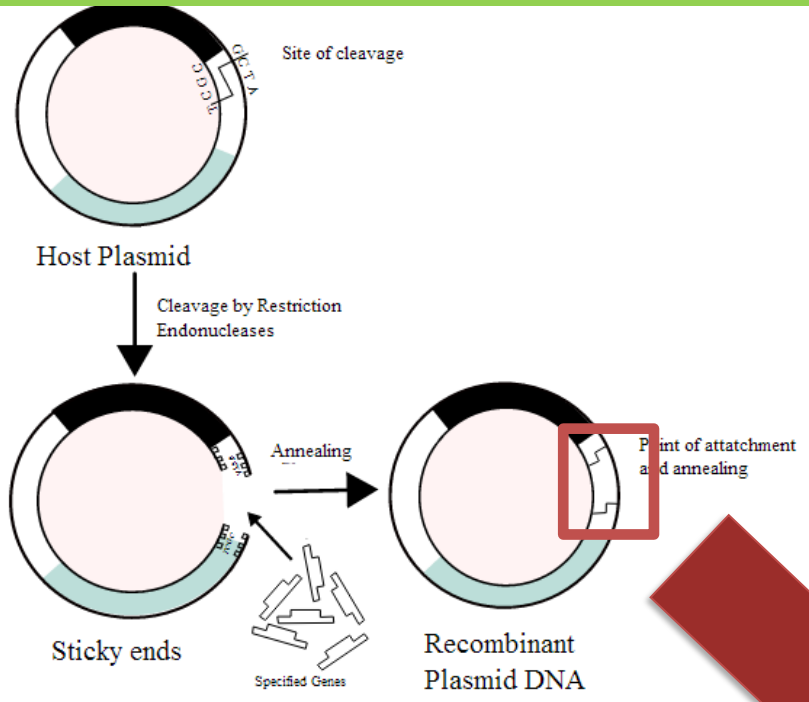
IS EVERYTHING LOST ?

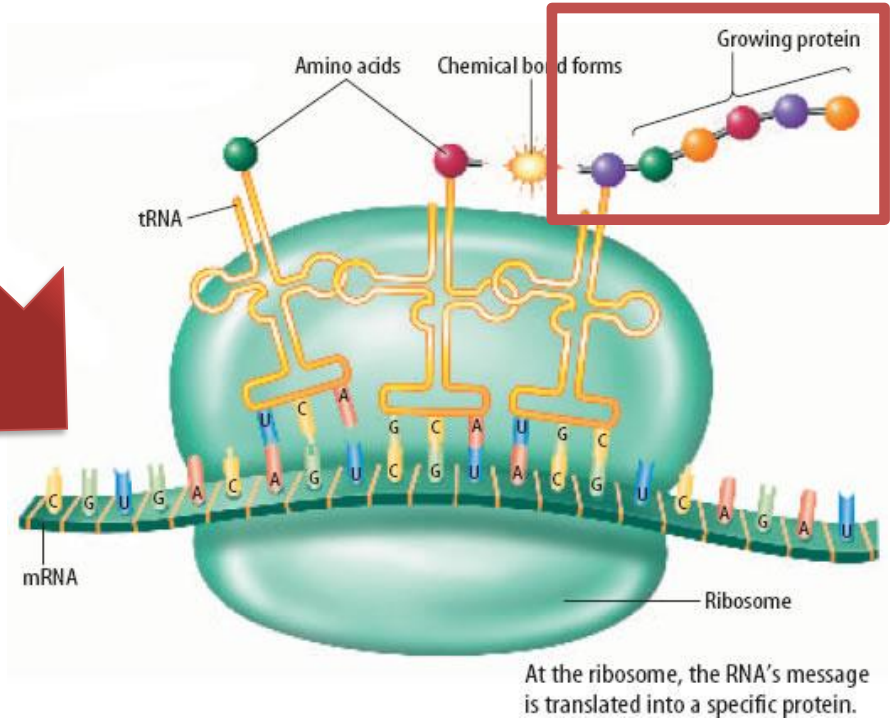
The need for a *different* approach

THEN ALONG CAME my wife

- **Biochemist**
- Works doing protein… something
- I suspect she really works doing…

BIOTECH WILL SAVE THE WORLD OR KILL US ALL

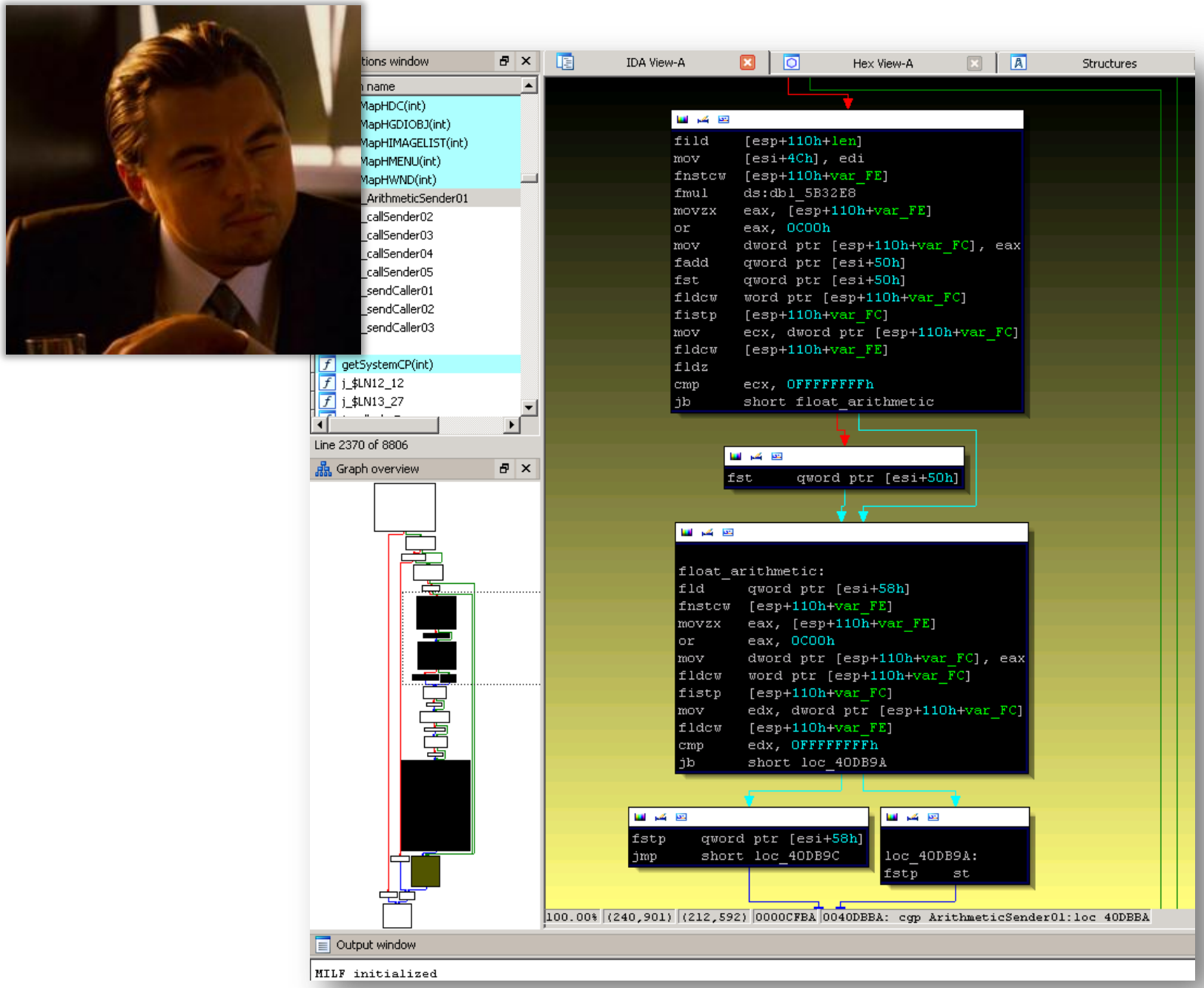


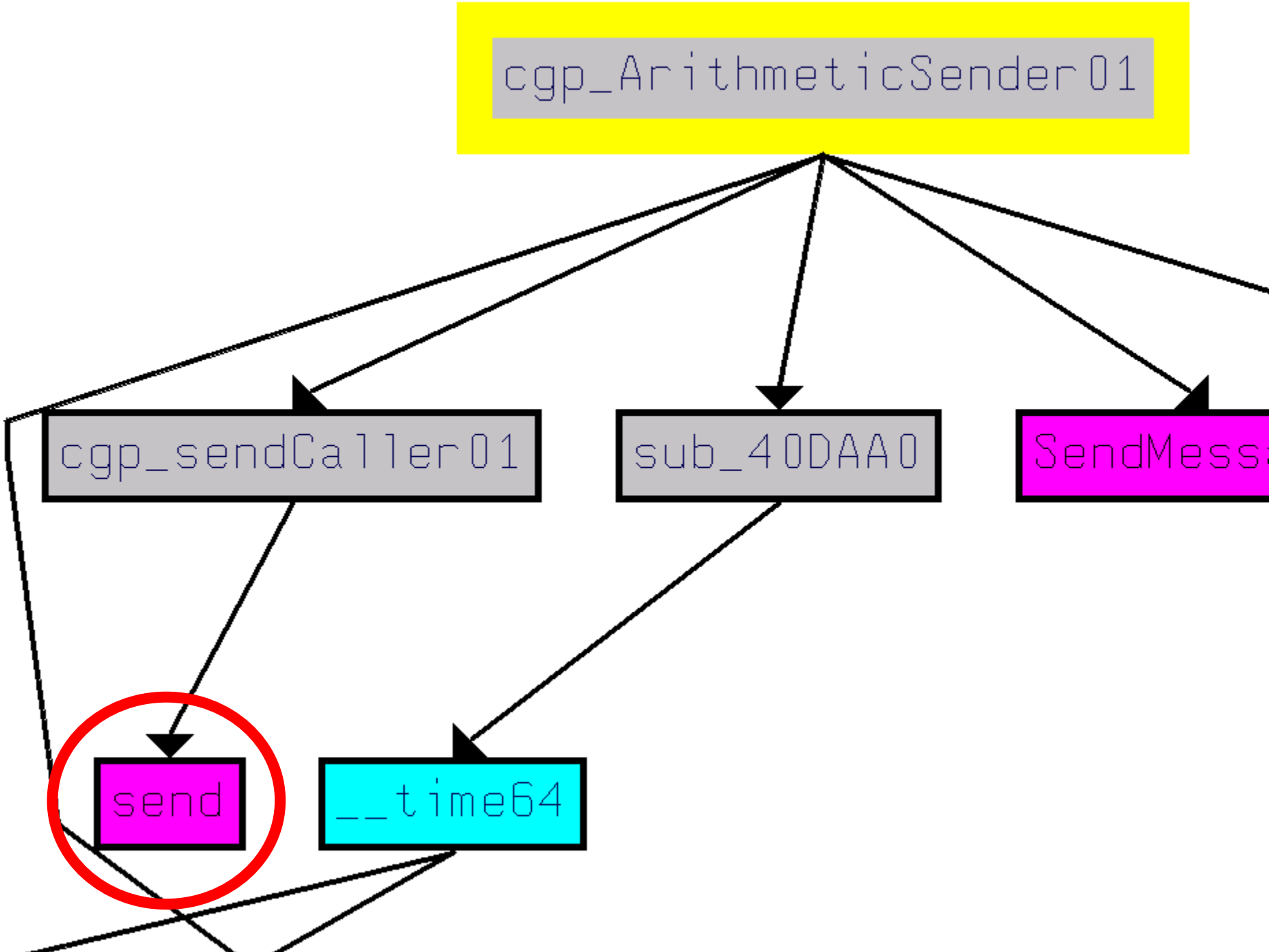


At the ribosome, the RNA's message is translated into a specific protein.


M&M's
SNICKERS

In a nutshell...


tions window
name
MapHDC(int)
MapHGDIOBJ(int)
MapHIMAGELIST(int)
MapHMENU(int)
MapHWND(int)
_ArithmeticSender01
_callSender02
_callSender03
_callSender04
_callSender05
_sendCaller01
_sendCaller02
_sendCaller03
getSystemCP(int)
j_$LN12_12
j_$LN13_27
Line 2370 of 8806
Graph overview
IDA View-A
Hex View-A
Structures
fild [esp+110h+len]
mov [esi+4Ch], edi
fnstcw [esp+110h+var_FE]
fmul ds:dbl_5B32E8
movzx eax, [esp+110h+var_FE]
or eax, 0C00h
mov dword ptr [esp+110h+var_FC], eax
fadd qword ptr [esi+50h]
fst qword ptr [esi+50h]
fldcw word ptr [esp+110h+var_FC]
fistp [esp+110h+var_FC]
mov ecx, dword ptr [esp+110h+var_FC]
fldcw [esp+110h+var_FE]
fldz
cmp ecx, 0FFFFFFFFh
jb short float_arithmetic
fst qword ptr [esi+50h]
float_arithmetic:
fld qword ptr [esi+58h]
fnstcw [esp+110h+var_FE]
movzx eax, [esp+110h+var_FE]
or eax, 0C00h
mov dword ptr [esp+110h+var_FC], eax
fldcw word ptr [esp+110h+var_FC]
fistp [esp+110h+var_FC]
mov edx, dword ptr [esp+110h+var_FC]
fldcw [esp+110h+var_FE]
cmp edx, 0FFFFFFFFh
jb short loc_40DB9A
fstp qword ptr [esi+58h]
jmp short loc_40DB9C
loc_40DB9A:
fstp st
100.00% (240,901) (212,592) 0000CFBA 0040DBBA: cgp ArithmeticSender01:loc_40DBBA
Output window
MILF initialized


cgp_ArithmeticSender01
cgp_sendCaller01
sub_40DAA0
SendMess
send
__time64

SIMPLE ARGUMENTS

```
; int __stdcall cgp_ArithmeticSender01(char *buf, int len, int)
cgp_ArithmeticSender01 proc near

var_FE= word ptr -0FEh
var_FC= qword ptr -0FCh
Dest= dword ptr -0F4h
var_4= dword ptr -4
buf= dword ptr  4
len= dword ptr  8
arg_8= dword ptr  0Ch

sub     esp, 100h
mov     eax, stackCookie
xor     eax, esp
mov     [esp+100h+var_4], eax
push    ebp
mov     ebp, [esp+104h+buf]
```


IT GETS EXCITING
WESLEYAN

Microsoft® Research

Detours

= userland hooking

= amazing stuff

= dynamic binary instrumentation

= AWESOME stuff !!!

Microsoft Detours

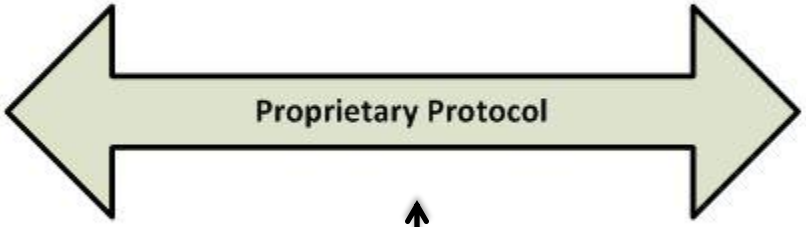
- **Library for intercepting arbitrary Win32 binary functions**.

- Interception code is applied dynamically **at runtime**.
- **Replaces the first few instructions** of the target function
- with an **unconditional jump** to the detour function.

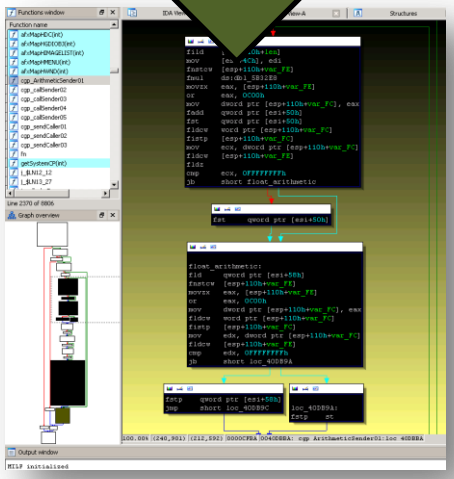
- **Replace** or **extend** the target function.

WHAT CAN POSSIBLY GO WRONG?

Server



Client (software)



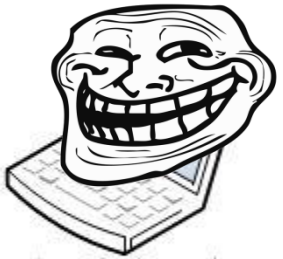
Developer:
“Can‘t touch this!“

WHAT CAN POSSIBLY GO WRONG?

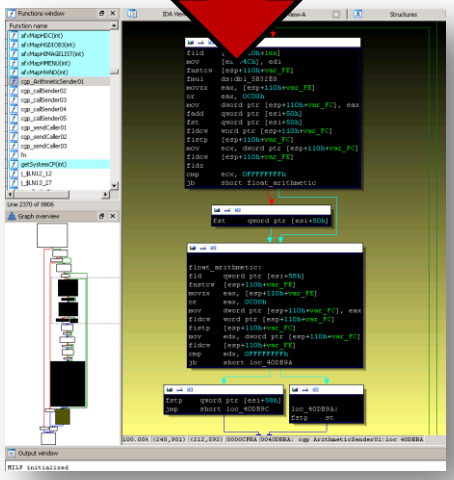
"Hmm… Can't touch this?"

Server

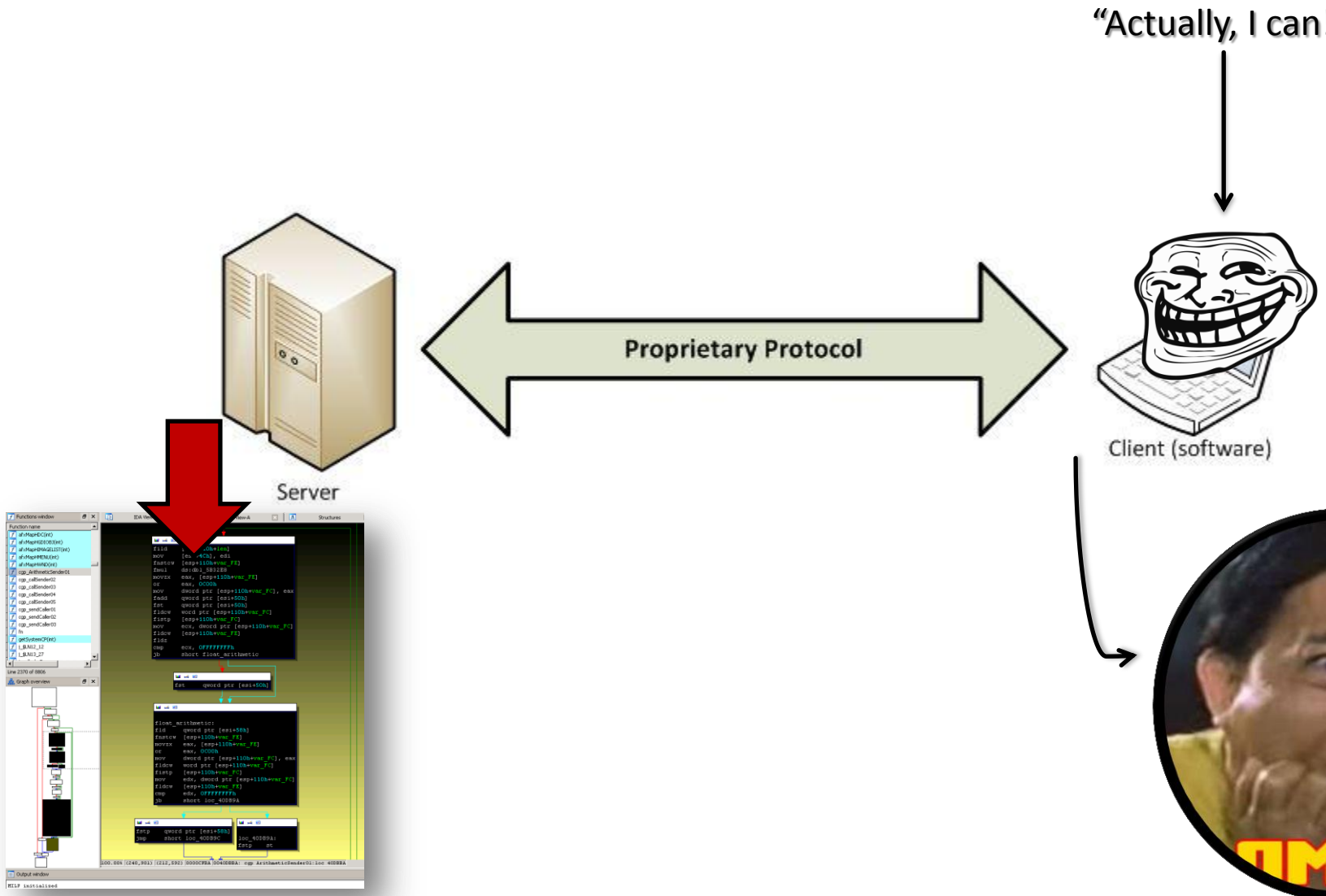
Proprietary Protocol



Client (software)



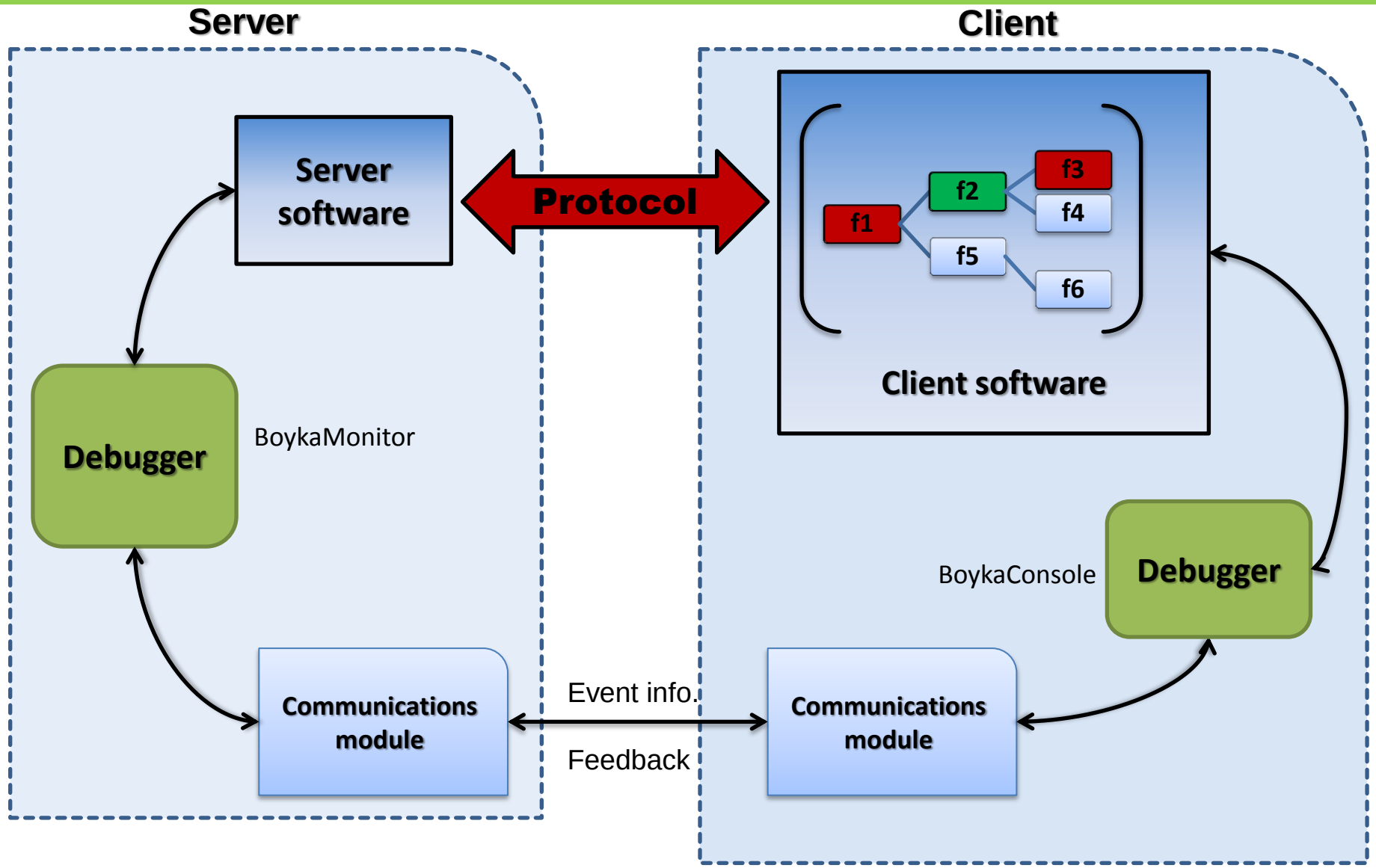
WHAT CAN POSSIBLY GO WRONG?



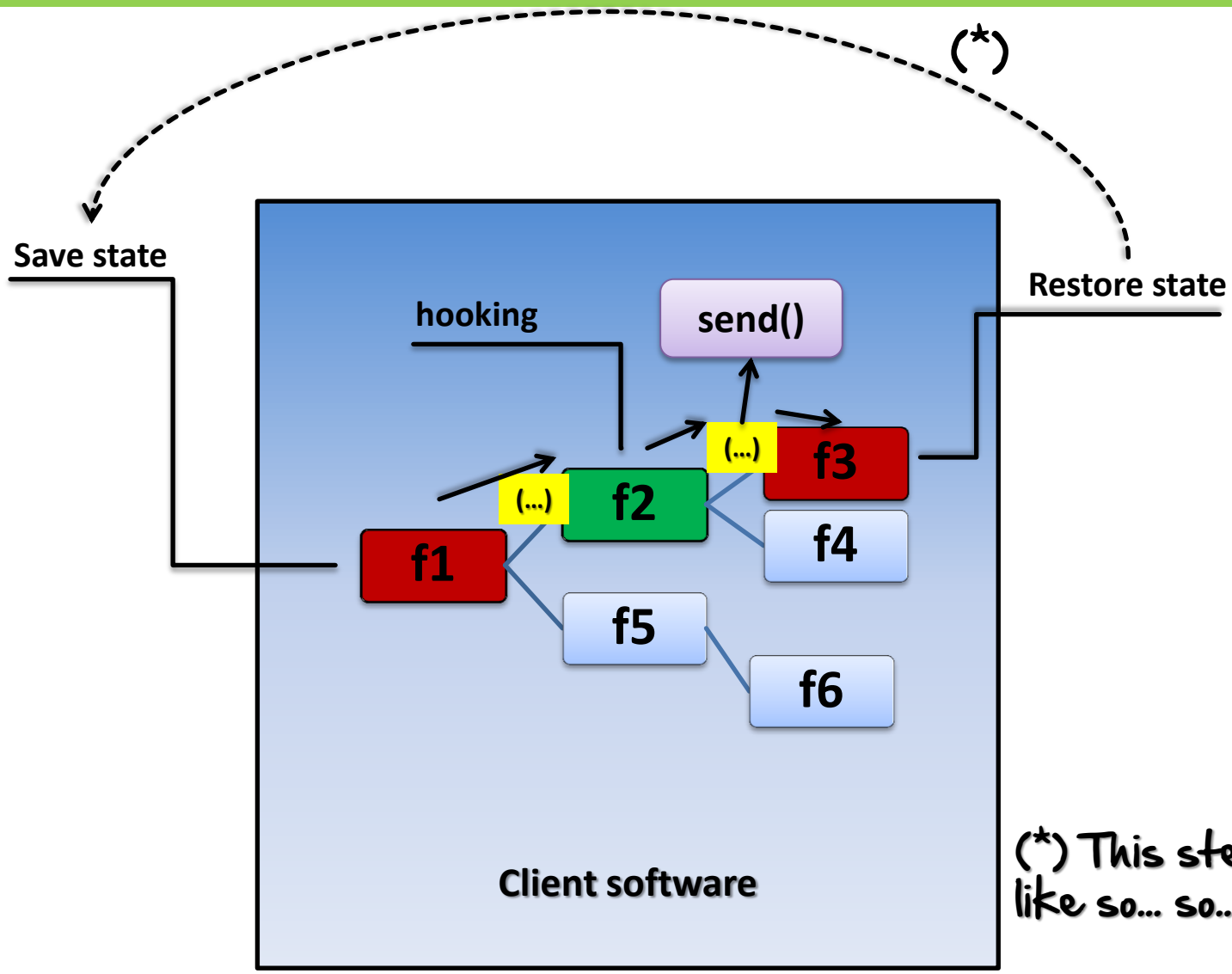
LONG STORY SHORT...


PLUMBING TIME

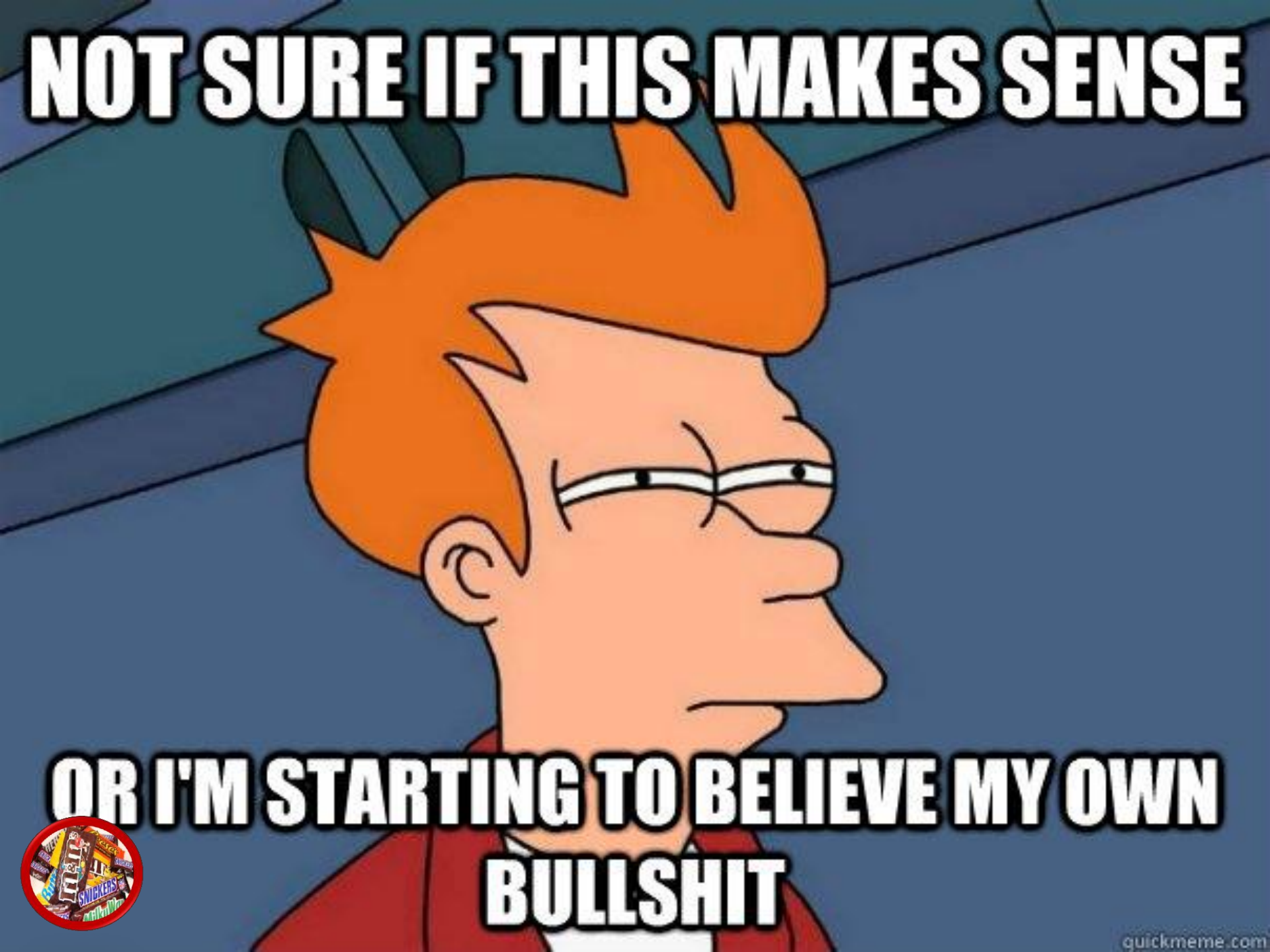
Overview (from a million miles away)



Overview (from a thousand miles away)



(*) This step works like so... so... right now

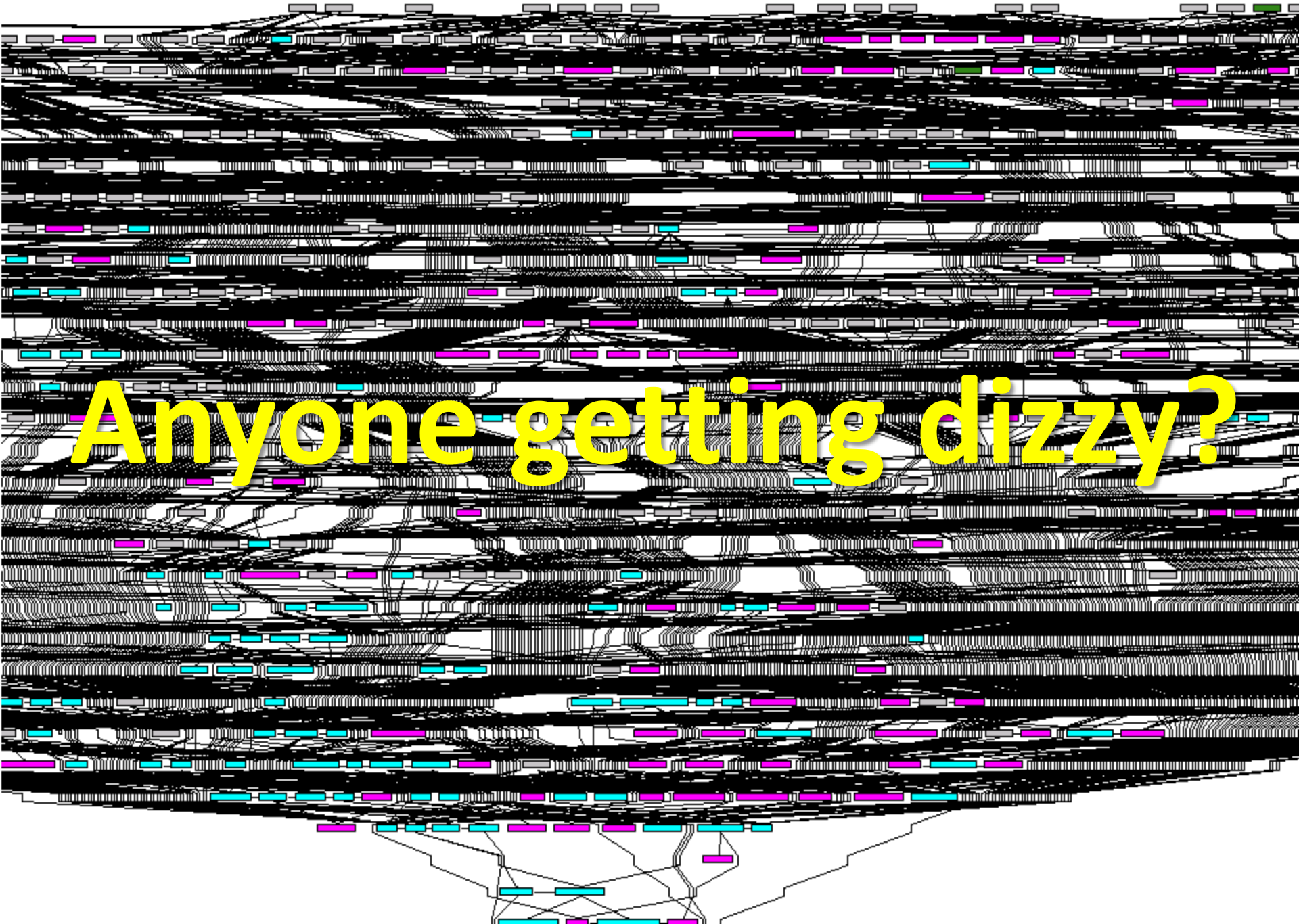

NOT SURE IF THIS MAKES SENSE
OR I'M STARTING TO BELIEVE MY OWN BULLSHIT
quickmeme.com

THE CHALLENGE

- I can "inject" some data into the server
- By hijacking client execution at certain points
- …
- … aha…

- **Which. Points. Do. I. Use. ?!?!?!?**


Carlos, help me!
There are TOO MANY Addresses
!!!!!!!!!!!
SNICKERS


Anyone getting dizzy?


BRACE YOURSELVES,
SHAMELESS
AUTOPROMOTION IS
COMING.

M*LF & PIN TRACER

beta

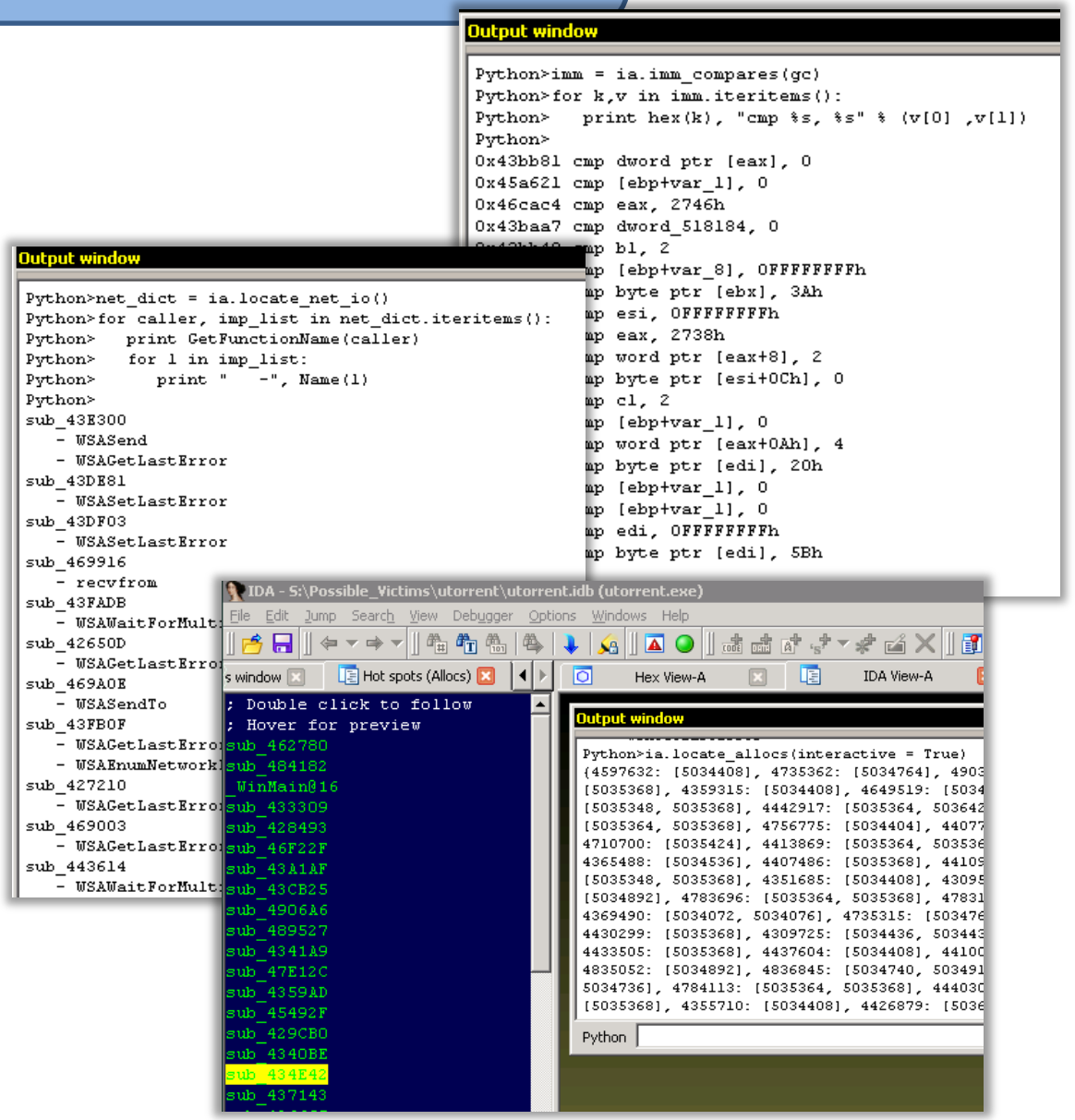
SOME COOL FEATURES

- Mark dangerous functions
- Find immediate compares
- Mark switches
- Show paths between functions
- Find File IO
- Find Network IO
- Find Allocations
- Find dangerous „size params“

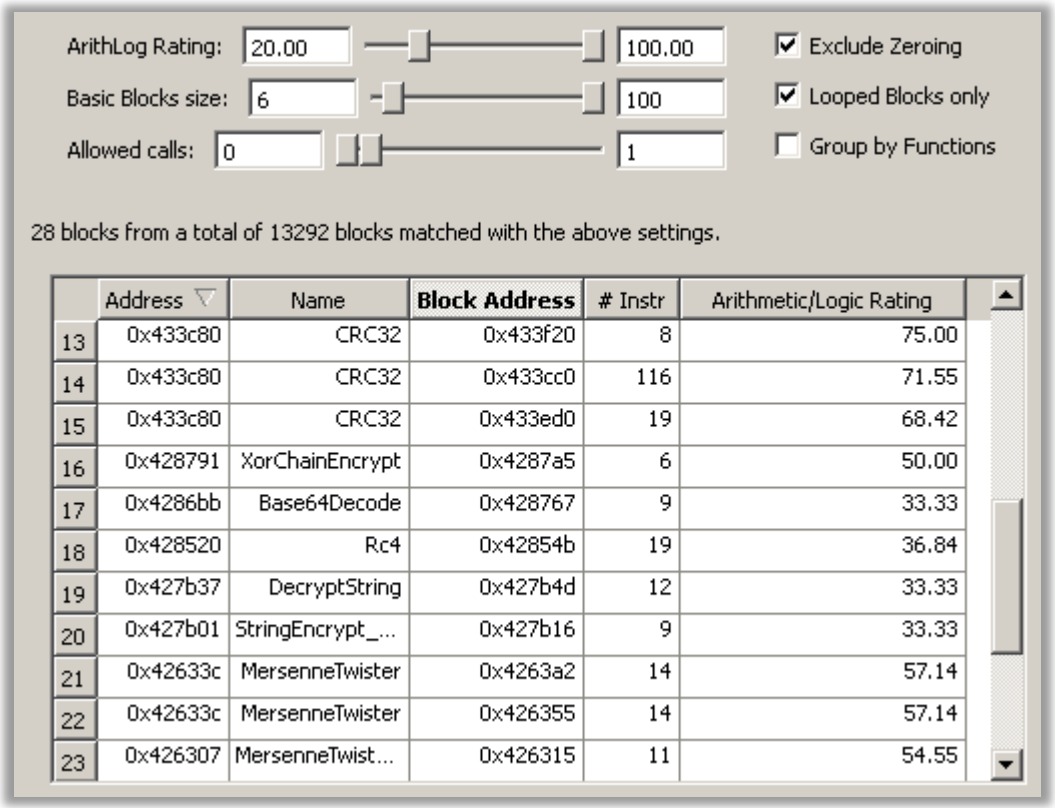
- Create IDA (connection) graphs

- Create „custom viewers“

- etc.



IDAScope



	Address	Name	Block Address	# Instr	Arithmetic/Logic Rating
13	0x433c80	CRC32	0x433f20	8	75.00
14	0x433c80	CRC32	0x433cc0	116	71.55
15	0x433c80	CRC32	0x433ed0	19	68.42
16	0x428791	XorChainEncrypt	0x4287a5	6	50.00
17	0x4286bb	Base64Decode	0x428767	9	33.33
18	0x428520	Rc4	0x42854b	19	36.84
19	0x427b37	DecryptString	0x427b4d	12	33.33
20	0x427b01	StringEncrypt_...	0x427b16	9	33.33
21	0x42633c	MersenneTwister	0x4263a2	14	57.14
22	0x42633c	MersenneTwister	0x426355	14	57.14
23	0x426307	MersenneTwist...	0x426315	11	54.55

http://pnx-tf.blogspot.com/

DIFFERENTIAL DEBUGGING

- Hook every function -> log hits.
- 1st run. Exercise as many functionality as possible.
- 2nd run. Focus on the interesting feature.
- Compare both -> filter out.

Function_1	Function_1
GUI_stuff	GUI_stuff
Windows_stuff	Windows_stuff
Function_2	Login_stuff
Thread_sync	Thread_sync
Function_3	Encryption_stuff
[…]	[…]

DIFFERENTIAL DEBUGGING

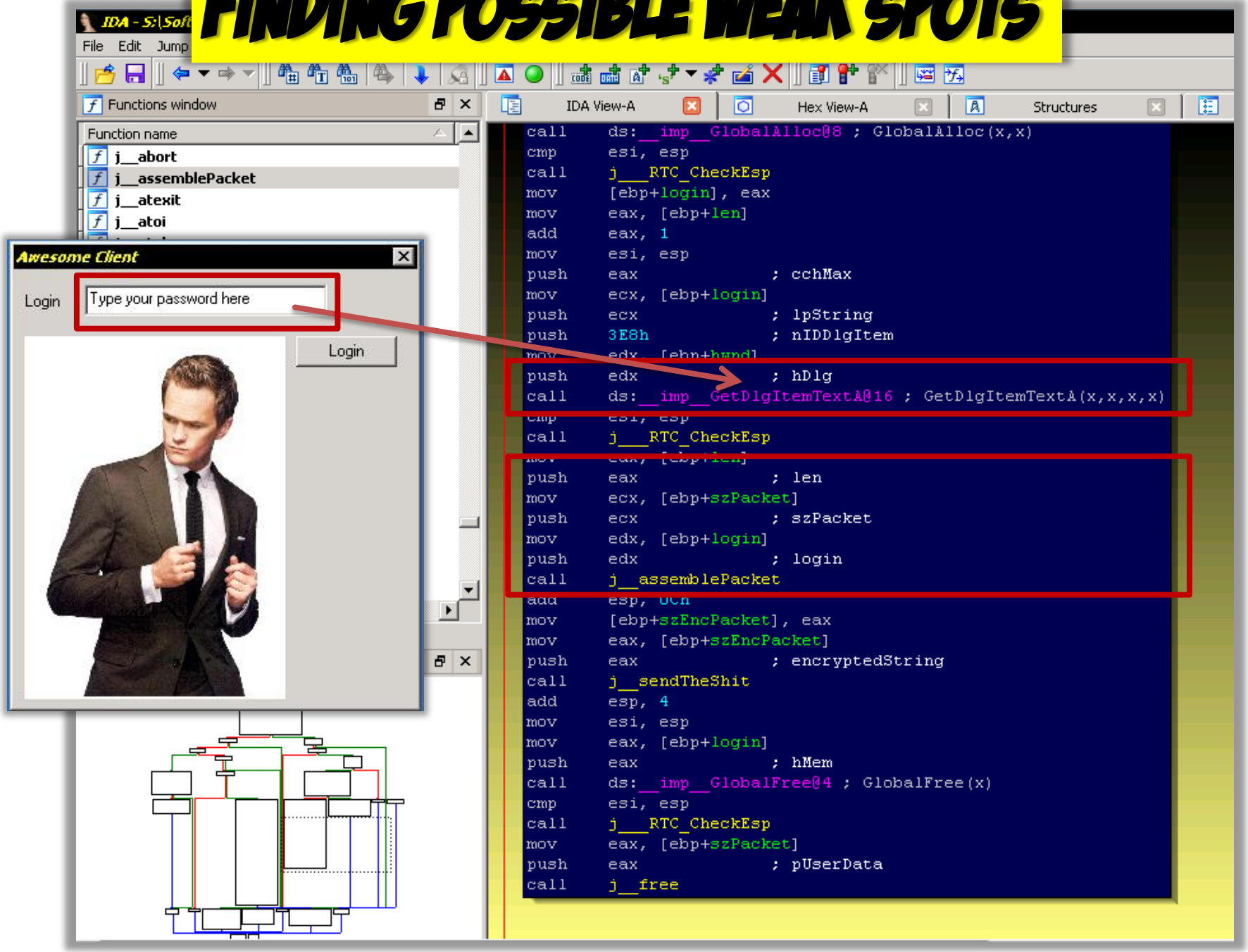
- Hook every function -> log hits.
- 1st run. Exercise as many functionality as possible.
- 2nd run. Focus on the interesting feature.
- Compare both -> filter out.

Function_1	Function_1	
GUI_stuff	GUI_stuff	
Windows_stuff	Windows_stuff	
Function_2	Login_stuff	Login_stuff
Thread_sync	Thread_sync	
Function_3	Encryption_stuff	Encryption_stuff
[…]	[…]	[…]

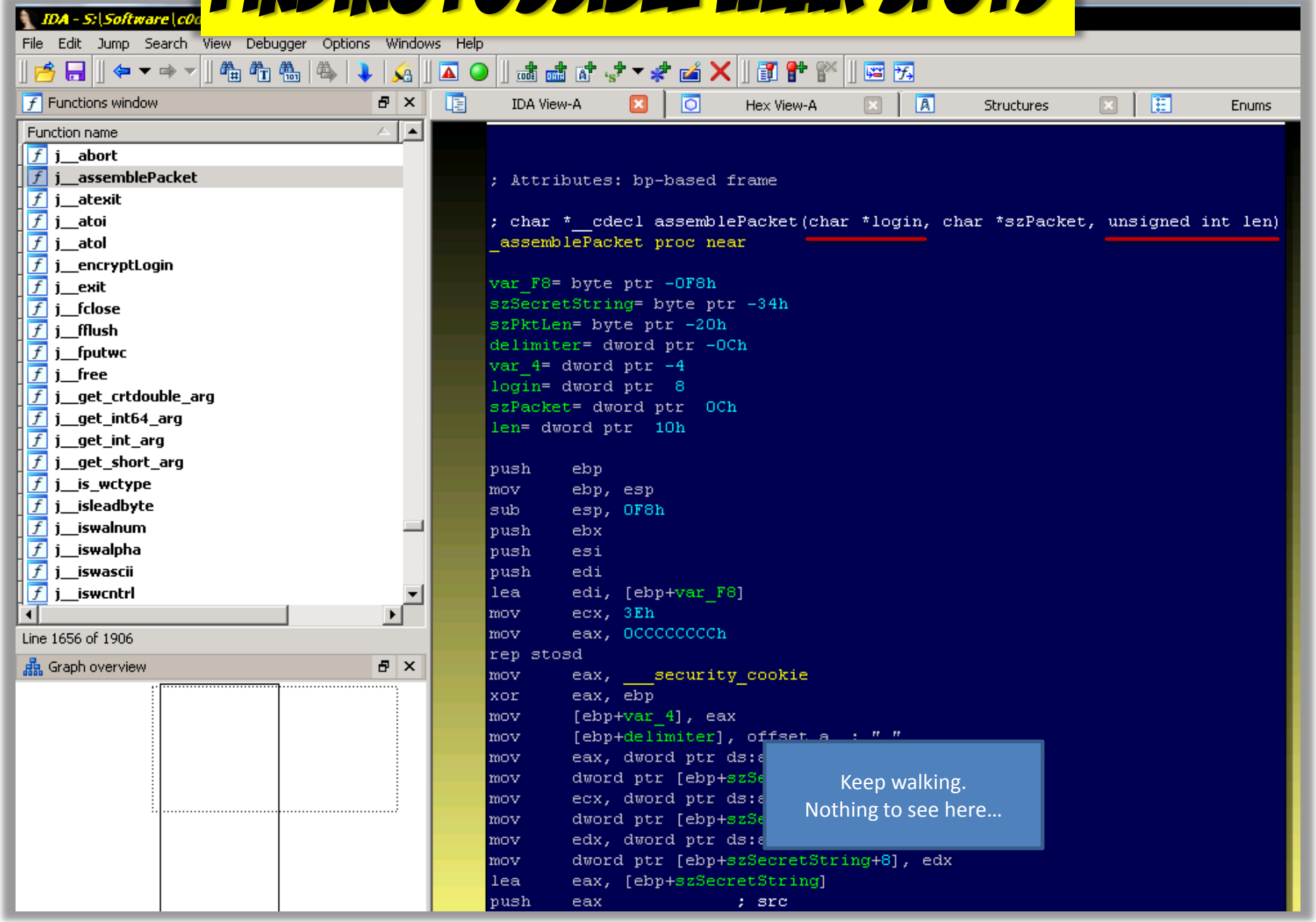

BUILD YOUR WEAPON

EPIC ASS KICKING

FINDING POSSIBLE WEAK SPOTS



FINDING POSSIBLE WEAK SPOTS



Cheating...

Client:

- Calculates login length
- Appends the length (in ASCII) to the login string.
- Appends a “custom“ string
- *Encrypts* everything

Server: Length value (from client)
used to *malloc()* & *strcpy()*

STAND BACK

I'M GOING TO TRY

SCIENCE

Where to go from here

- Better static / dynamic analysis
  - Automatization
  - Heuristic based
- Save / restore snapshot
  - Full emulation (Thx @pleed_ !)
  - Qemu-dbi?

Everything is Online

You can lulz at my code at:

https://github.com/carlosgprado/Boyka

@m0n0sapiens

carlos.g.prado@gmail.com


IT'S
BEEN
LOVELY
BUT
I HAVE TO
SCREAM
NOW