Hijacking client software for fuzz and profit
- 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

CENSORED

F**K THAT SH*T
OF NORWAY
PRIME MINISTER
LET'S BE CLEAR

THIS WORK IS SHIT
LET'S BE CLEAR

THIS WORK IS SHIT

"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!

Look for the ChocoQuiz Icon
DISCLAIMER

FOR EDUCATIONAL PURPOSES ONLY...
Fuzzing 101
Fuzzing is like violence: if it doesn’t solve your problems, you are not using enough of it.
BRUTE FORCE
from sulley import *
from requests import ftp # this is our ftp.py file

sess = sessions.session(session_filename="audits/freeftpc/pftp.session")
target = sessions.target("192.168.1.1", 21)
target.netmon = pedrpc.client("192.168.1.1", 26001) # NetMonitor (packets)
target.procmon = pedrpc.client("192.168.1.1", 26002) # ProcMonitor (crashes)
target.procmon_options = { "proc_name" : "FTPServer.exe" }

tess.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("mdtr"))
sess.connect(s_get("pass"), s_get("mkcd"))
sess.fuzz()

ftp_session.py
And Now

I Wait
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:

fffffffff -> kernel32.dll:7c839ad8 push ebp

PWNED!
There's always a but

That's all very nice.

But what if I don't know the protocol?
There is NO documentation at all. :((((((

I CAN ALWAYS TRY DUMB FUZZING!
Think about checksums...

Packet

Data

Checksum

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

Dumb Fuzzing...
F*CK!
Is everything lost?

FAIL HARDER

THINK WRONG
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

Host Plasmid

Site of cleavage

Clavage by Restriction Endonucleases

Sticky ends

Annealing

Specified Oligos

Recombinant Plasmid DNA

At the ribosome, the RNA’s message is translated into a specific protein.
In a nutshell...

WE ARE SECURE. NOBODY KNOWS OUR PROTOCOL.

THE CLIENT DOES...
; int __stdcall cgP_ArithmeticSender01(char *buf, int len, int)
cgp_ArithmeticSender01 proc near

var_FE= word ptr -0F Eh
var_FC= qword ptr -0F Ch
Dest= dword ptr -0F 4h
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
Detours
= userland hooking
= amazing stuff

= dynamic binary instrumentation
= AWESOME stuff !!!
- 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?

Developer: “Can’t touch this!”
What can possibly **go wrong**?

"Hmm... Can't touch this?"
WHAT CAN POSSIBLY **GO WRONG?**
Long story short...

Dance, Puppet!

Dance!
PLUMBING TIME
OVERVIEW (FROM A MILLION MILES AWAY)

Server
- Server software
- Debugger
- Communications module

Client
- Client software
- Debugger
- Communications module

Protocol
- Event info.
- Feedback
OVERVIEW (FROM A THOUSAND MILES AWAY)

Client software

Save state

hooking

send()

f1

f2

f3

f4

f5

f6

Restore state

(*) This step works like so... so... right now
NOT SURE IF THIS MAKES SENSE
OR I'M STARTING TO BELIEVE MY OWN BULLSHIT
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
Anyone getting dizzy?
BRACE YOURSELVES, SHAMELESS AUTOPROMOTION IS COMING.
M*LF & PIN TRACER
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

28 blocks from a total of 13292 blocks matched with the above settings.

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

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
GUI_stuff
Windows_stuff
Function_2
Thread_sync
Function_3
[...]

Function_1
GUI_stuff
Windows_stuff
Function_1
Login_stuff
Thread_sync
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 | Function_1 |
| GUI_stuff  | GUI_stuff  | GUI_stuff  |
| Windows_stuff | Windows_stuff | Windows_stuff |
| Function_2 | Login_stuff | Login_stuff |
| Thread_sync | Thread_sync | Encryption_stuff |
| Function_3 | Encryption_stuff | Encryption_stuff |

[...]

[...]

[...]

[...]
BUILD YOUR WEAPON
EPIC ASS KICKING

WAKE UP!
You're gonna miss the good stuff!!!
Finding possible weak spots
Finding possible weak spots

Keep walking.

Nothing to see here...
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?
You can [lulz](https://github.com/carlosgprado/Boyka) 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