

# ANTI REVERSE ENGINEERING

1

# Obfuscation <-> Clean Code

Many “Clean Code” guides

Irony “Unmaintainable Code” guide

<https://github.com/Droogans/unmaintainable-code/blob/master/README.md>

<https://www.mindprod.com/jgloss/unmain.html>

3

```
public class Main {
    String Reaple = "username";
    String Cryze = "username";
    String f0MtMoony = "username";

    public static long a(byte[] a) {
        long c = 0;
        for (byte b : a) {
            c = (c << 8) + (b & 0xFF);
        }
        return c;
    }

    public static void main(String[] args) {
        String int;
        long password;
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the username: ");
        int = scanner.nextLine();
        password = Main.a(int.getBytes());
        if (Main.a(Main.class.getDeclaredFields()[0].getName().getBytes()) == password) {
            System.out.println("Connect!");
        } else {
            System.out.println("Wrong!");
        }
    }
}
```

```
public class Main {
    String Reaple = "username";
    String Cryze = "username";
    String f0MtMoony = "username";

    /* renamed from: MtMoony reason: contains not printable characters */
    String f0MtMoony = "username";

    /* renamed from: a */
    public static long m0a(byte[] a) {
        long c = 0;
        for (byte b : a) {
            c = (c << 8) + (b & 255);
        }
        return c;
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the Username: ");
        long password = m0a(scanner.nextLine().getBytes());
        if (m0a(Main.class.getDeclaredFields()[0].getName().getBytes()) == password) {
            System.out.println("Correct!");
        } else {
            System.out.println("Wrong!");
        }
    }
}
```

4

"We achieved our goals. We were uncracked for 13 whole days."

– Martin Slater, 2K Australia, on *BioShock* (2007).

## What is enough?

The same few techniques

Well known and reversible

It just needs to do the job

Every Obfuscation will be reverse-engineered some day

5

## Overdoing it, A LOT

High difficulty CTF  
Challenges

Obfuscation Contest  
<https://www.ioccc.org/>

Also interesting: Code Golf

```
1  short main[] = {
2      277, 04735, -4129, 25, 0, 477, 1019, 0xbef, 0, 12800,
3      -113, 21119, 0x52d7, -1006, -7151, 0, 0x4bc, 020004,
4      14880, 10541, 2056, 04010, 4548, 3044, -6716, 0x9,
5      4407, 6, 5568, 1, -30460, 0, 0x9, 5570, 512, -30419,
6      0x7e82, 0760, 6, 0, 4, 02400, 15, 0, 4, 1280, 4, 0,
7      4, 0, 0, 0x8, 0, 4, 0, ',', 0, 12, 0, 4, 0, '#',
8      0, 020, 0, 4, 0, 30, 0, 026, 0, 0x6176, 120, 25712,
9      'p', 072163, 'r', 29303, 29801, 'e'
10 };
```

6

```

k;double sin(
    ,cos();main(){float A=
    0,B=0,i,j,z[1760];char b[
    1760];printf("\x1b[2J");for(;;
){memset(b,32,1760);memset(z,0,7040)
;for(j=0;6.28>j;j+=0.07)for(i=0;6.28
>i;i+=0.02){float c=sin(i),d=cos(j),e=
sin(A),f=sin(j),g=cos(A),h=d+2,D=1/(c*
h*e+f*g+5),l=cos(i),m=cos(B),n=s\
in(B),t=c*h*g-f*e;int x=40+30*D*
(1*h*m-t*n),y=
12+15*D*(1*h*n
+t*m),o=x+80*y,
N=8*((f*e-c*d*g
)*m-c*d*e-f*g-l
*d*n);if(22>y&&
y>0&&x>0&&80>x&&D>z[o]){z[o]=D;;;b[o]=
".,~::~!=!$#@'[N>0?N:0];}]/*****!!-*/
printf("\x1b[H");for(k=0;1761>k;k++)
putchar(k%80?b[k]:0);A+=0.04;B+=
0.02;}}/*****#####*****!!==::~-
~::~==!!|*****!!==::-
.,~::~;;=====;;::~-
.,,-----,*/

```

## Compiled languages



# Interpreted languages

Mostly Powershell or JS

Not intended to protect against humans

Administrator: Windows PowerShell (x86)

Microsoft Corporation. All rights reserved.

Get-Help update-help  
Get-Help get-process -examples

that are running on the local computer or a remote computer.

----- EXAMPLE 1 -----

list of all of the running processes running on the local computer. For more information, see the "Additional Notes" section of the Help topic for Get-Help.

----- EXAMPLE 2 -----

winword, explorer | format-list \*

list available data about the Winword and Explorer processes on the local computer. The cmdlet displays all available properties (\*) of the processes, but it omits the optional parameter name. The list cmdlet, which displays all available properties (\*) of the processes by their process IDs. For example, "get-process

----- EXAMPLE 3 -----

where-object {\$\_.WorkingSet -gt 20000000}

list processes that have a working set greater than 20 MB. It uses the pipeline operator (|) to pass the process objects to the where-object cmdlet, which filters the objects based on the value of the WorkingSet property. To see all of the properties of process objects, use the Get-Process cmdlet with a value greater than 20,000,000 bytes for the WorkingSet parameter. The values of all amount properties are in bytes, even though they are displayed in megabytes.

9

# Encoding

Almost every Payload has some form of encoding

Base64, Hex

Protects against antivirus



10

# Variable and Identifier Transformation

Names are changed  
Unicode Characters ;)

Variable values are hidden

Functions are used to hide  
constants

```

    (parseInt(_0x38755e(0x8)) / 0x1) +
    (parseInt(_0x38755e(0x9)) / 0x2) +
    parseInt(_0x38755e(0xa)) / 0x3 +
    -parseInt(_0x38755e(0x2)) / 0x4 +
    (-parseInt(_0x38755e(0x4)) / 0x5) *
    (parseInt(_0x38755e(0x5)) / 0x6) +
    (-parseInt(_0x38755e(0x6)) / 0x7) *
    (-parseInt(_0x38755e(0x1)) / 0x8) +
    parseInt(_0x38755e(0x0)) / 0x9 +
    parseInt(_0x38755e(0x8)) / 0xa;
    if (_0x2bae1b == _0x48ea7a) {
        break;
    } else {
        _0x4b900f["push"](_0x4b900f["shift"]());
    }
} catch (_0x3af045) {
    _0x4b900f["push"](_0x4b900f["shift"]());
}
}

})(_0xe467, 0x72e85);
function hi() {
    var _0x2930af = _0x3c3e;
    console[_0x2930af(0x7)](_0x2930af(0x3));
}
function _0xe467() {
    var _0x494fd2 = [
        "588FcDLIM",
        "27762bLcXtj",
        "log",
        "11068250TDAAG",
        "62636kgLSvX",
        "932820MwPZvX",
        "28dFURUt",
        "317682JrvGZD",
        "1816kTnLBH",
        "748744kGzxjB",
        "Hellox20World!",
        "41765nSUYWe",
    ];
};
// function hi() {
//     console.log("Hello World!");
// }
// hi();
//

```

11

```

function obfuscateVariables($str)
{
    $varNumber = 0; // 0 is used for the string obfuscator function in $exploit
    while(true)
    {
        $varName = "var",((string)$varNumber).""; // variable name referenced in $exploit
        $str = str_replace($varName,randomString(mt_rand(3,8)), $str,$count); // replace all with a random string (length between 3-8)
        if($count == 0) // if none found stop
        {
            break;
        }
        $varNumber++;
    }
    return $str;
}

function obfuscateStrings($str,$key)
{
    $str = str_replace("KEY",$key,$str); // put the decryption key in
    $length = strlen($str);
    $startPos = 0;
    while($startPos == FALSE)
    {
        $startPos = strpos($str,"str");
        if($startPos == FALSE) // break if none found
        {
            break;
        }
        $endPos = strpos($str,"-", $startPos);
        $text = substr($str,$startPos+5,$endPos - $startPos - 5); // text to be obfuscated
        $textLen = strlen($text);
        $res = "";
        for($x = 0; $x < $textLen; $x++) // obfuscate string
        {
            $charCode = ord(substr($text,$x,1)) + $key; // add key to charcode
            $randInt = mt_rand(0,$charCode); // pick a random number
            $res .= "var@-".((string)($charCode - $randInt))."-".((string)$randInt).""; // subtract that number from charcode and
            if($x == $textLen - 1)
            {
                $res .= "+"; // concat operator in js
            }
        }
        $str = substr($str,0,$startPos).$res.substr($str,$endPos+1); // re-assemble the string
    }
    return $str;
}

function NtrH(DMUQDh) {
    return String.fromCharCode(DMUQDh - 3029);
}

function TnLtwW0K(PkcN0ZCv, rgSN) {
    var uGWdE =
        gEzExk +
        "/" +
        rgSN +
        NtrH(269 + 2806) +
        NtrH(556 + 2574) +
        NtrH(1951 + 1198) +
        NtrH(1966 + 1164);
    hrQhAnM[
        NtrH(1594 + 1546) +
        NtrH(621 + 2520) +
        NtrH(1493 + 1637) +
        NtrH(525 + 2614)
    ](
        NtrH(1078 + 2022) + NtrH(1251 + 1847) + NtrH(2077 + 1036),
        PkcN0ZCv,
        false
    );
    hrQhAnM[
        NtrH(1189 + 1955) +
        NtrH(2048 + 1082) +
        NtrH(1788 + 1351) +
        NtrH(1900 + 1229)
    ]();
    if (hrQhAnM.status == 200) {
        var wwIvsYp = new LYyNMBKk(
            NtrH(2097 + 997) +

```

12



13

# Better Obfuscation necessary

```
public void Validate()
{
    if (this.SensitiveTimeLockAlgorithm())
    {
        this.LogMessages.Add("Your evaluation has expired. Please purchase the full version of this product.");
        this.SensitiveApplicationDisablingProcedure();
    }
}
```

[illegible]

- Very popular in C# malware
- Prevents string searches
- Wraps encrypted strings in functions

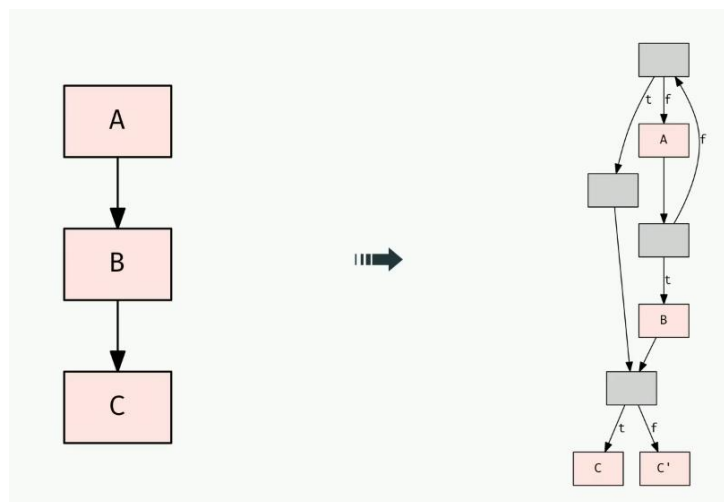
Original Source Code Before Control Obfuscation	Reverse-Engineered Source Code After Control Flow Obfuscation
<pre> public int CompareTo (Object o) {     int n = occurrences -     ((WordOccurrence)o).occurrences;     if (n=0) {         n=String.Compare         (word, ((WordOccurrence)o).word);     }     return (n); } </pre>	<pre> private virtual int_a(Object A+0) {     int local0;     int local1;     local 10 =this.a - (c) A_0.a;     if (local10 !=0) goto i0;     while (true) {         return local1;     }     i1:local10=     System.String.Compare(this.b, (c)     A_0.b);     goto i0; } </pre>

## Control Flow Obfuscation

Adds useless branches and loops

Gets way more effective with increasing scope

15



## Obfuscated Function Graph

16



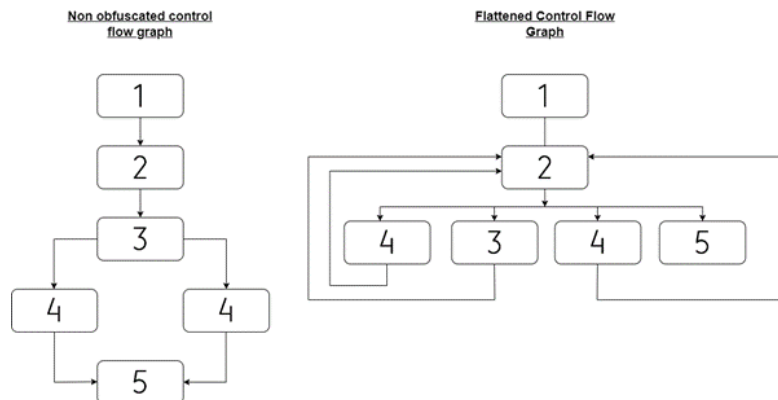
# Example: High-End Tools

```

00e8 void* const* rbp = &__return_addr
0110 sub_8041011(arg1, arg2, data_804205e,
0132 syscall(0x3c, status: 0)
0132 noreturn

```

17



## Control Flow Flattening

Uses state to obscure function graph

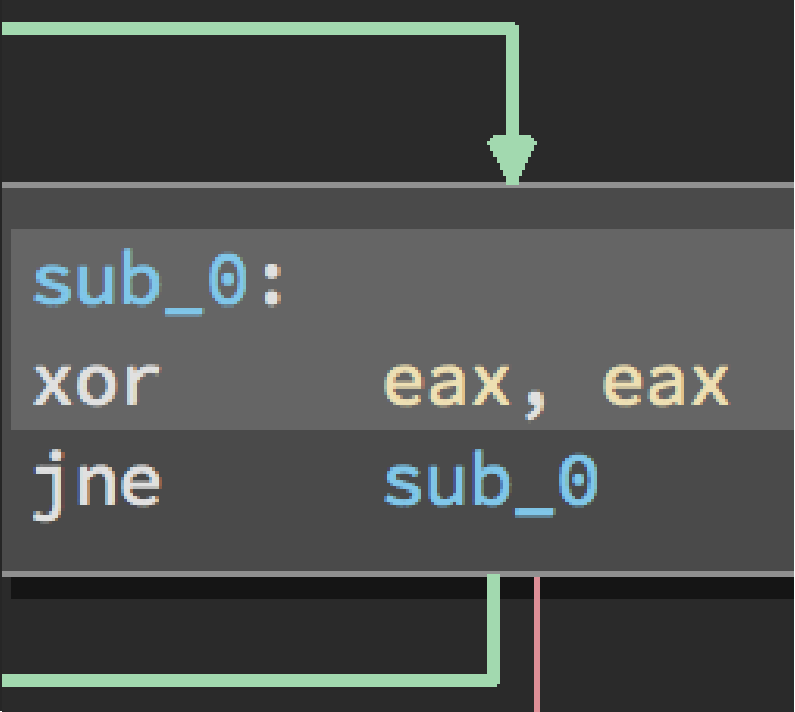
18

## Opaque Predicates

Branch with unknown result

Useless against dynamic analysis

Some high-end tools can help



```
sub_0:
xor     eax, eax
jne     sub_0
```

19

## Common Tools

Java/Android

• DEXGUARD

Executables

• Tigress

C#

• Dotfuscator

Price	<a href="#">Request a quote</a> for team pricing	<a href="#">Request a quote</a> for group pricing	<a href="#">Request a quote</a> for volume pricing
-------	--	---	--

20

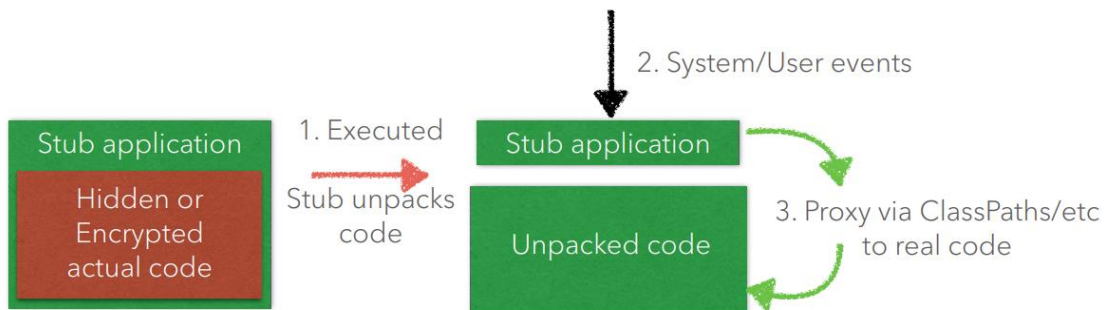
How to Rev?

## Execution Trace

## Patchen

## Manually reconstruct function graph

21



## Packing

A function or entire program are embedded into another application

Decrypted / Decoded during runtime

Popular with android malware

22

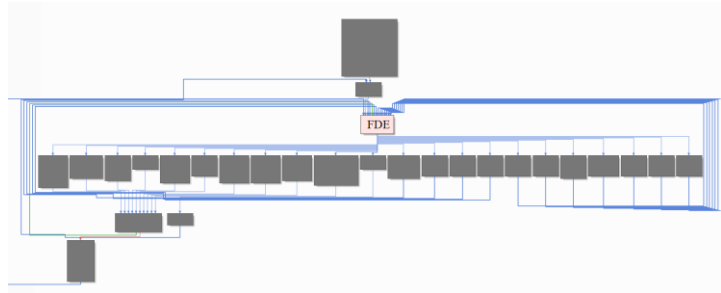
# VM Obfuscation

Program only interprets custom code

Can be highly complex

## Tools

- Tigress
- pyinstaller / py2exe



23

# Popular Packers

## Binaries

- UPX
- Obsidian

## Android

- DexGuard

## C#

- Dotfuscator

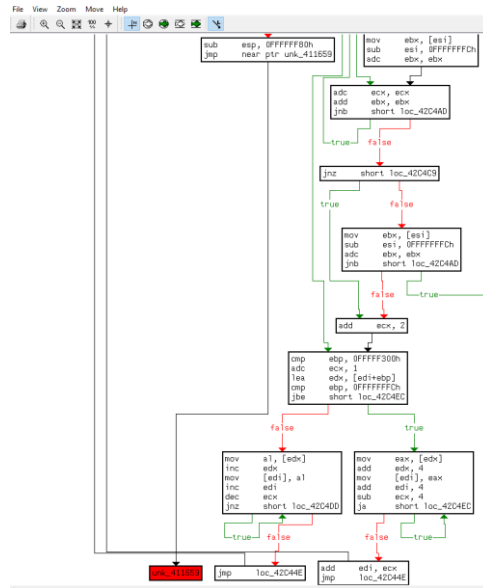
## Python

- pyinstaller

24

# How to Rev?

1. Find unpacking
2. OEP (Original Entry Point)
3. Breakpoint or Unpack



25

# Novelty Techniques

No practical use outside of CTFs

Very large overheads, huge output filesize

Cutting Edge Techniques

M/o/Vfuscator2

[github.com/xoreaxeaxeax/movfuscator](https://github.com/xoreaxeaxeax/movfuscator) :: the single instruction C compiler

chris domas @xoreaxeaxeax

26

[illegible]

# Flow Graph Art

<https://www.youtube.com/watch?v=HIUe0TUHOlc>

<https://github.com/xoreaxeaxeax/REpsych>

<https://github.com/JuliaPoo/Artfuscator>

29



30



# Language Features

Python -> List Comprehension & Lambdas

Java -> Streams & Lambdas

Assembly / C -> Intrinsic

31

# Demo

---

Powershell

---

Basic Control Flow Obfuscation

---

UPX Packer

---

MoVfuscator

---

Python List Comprehension

---

Instruction Overlapping

32