

2020년 2분기 사이버 보안 빅데이터 활용 공유 세미나

# 스파이웨어 에이전트 테슬라의 최신 변종 분석 기법 공유

엔키 서명환 주임연구원



# 목 차

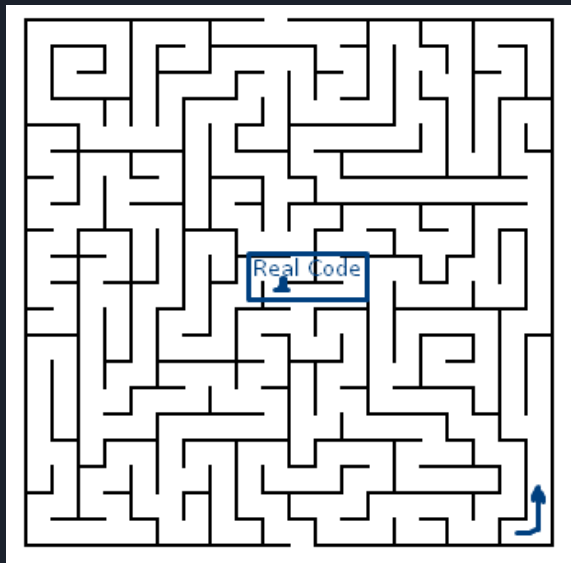
에이전트 테슬라

에이전트 테슬라 변종

에이전트 테슬라 변종 : 난독화 분석 및 해제 기법

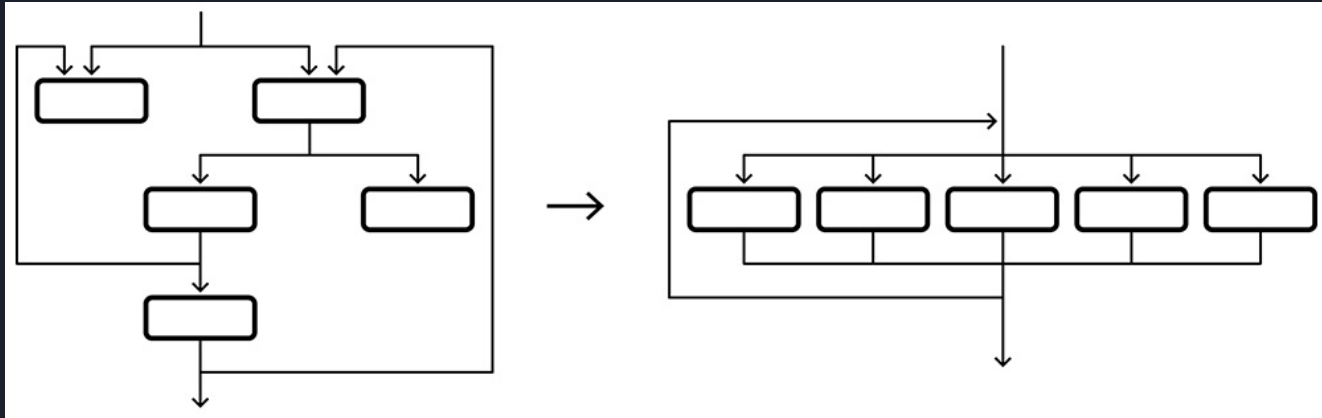
마치며

# 들어가며



들어가며

How To?



## 들어가며

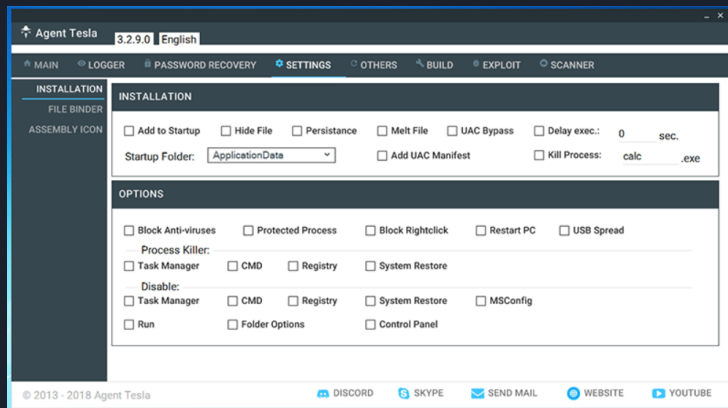


- 지능형 악성코드는 사용자 신뢰 확보를 위해 일반 프로그램과 유사한 형태로 제작 및 유포
- **에이전트 테슬라**는 상용 Keystroke Logger 프로그램이나, 이메일에 첨부된 문서 형태의 Dropper에 의해 유포되는 방식으로 악용
- 2020년 이집트의 국영 석유회사인 엔피를 사칭하여 말레이시아, 미국, 이란, 남아공, 오만, 터키 등 여러 나라의 석유 및 가스 회사 등 주요 기반시설들을 공격한 이력이 있음



# 에이전트 테슬라 (Agent Tesla)

# 에이전트 테슬라



PC 제어 및 모니터링

Pricing			
BRONZE	SILVER	GOLD MOST POPULAR	PLATINUM
\$15	\$35	\$49	\$69
1 Month License 7/24 Support Web Panel Advanced Keylogger	3 Months License 7/24 Support Web Panel Advanced Keylogger Crypter	6 Months License 7/24 Support Web Panel Advanced Keylogger Crypter doc/xls Converter	1 Year License 7/24 Support Web Panel Advanced Keylogger Crypter doc/xls Converter
1 Month Updates 1 Month Builds	3 Months Updates 3 Months Builds	6 Months Updates 6 Months Builds	1 Year Updates 1 Year Builds
Buy Now	Buy Now	Buy Now	Buy Now

구독형 요금제



# 에이전트 테슬라의 주요 기능

01	패스워드 탈취	<ul style="list-style-type: none"><li>• 브라우저 로그인 정보</li><li>• 서버 계정 정보</li><li>• 메일앱 계정 정보</li></ul>
02	화면 캡처 / 웹캠 녹화	<ul style="list-style-type: none"><li>• 사용자 화면 캡처</li><li>• 웹캠 녹화 기능</li></ul>
03	키로깅	<ul style="list-style-type: none"><li>• 클립보드 탈취</li><li>• 키보드 입력 탈취</li></ul>
04	백신 우회, 자가 보호	<ul style="list-style-type: none"><li>• 프로세스 보호 기능</li><li>• 백신 우회 기능</li></ul>
05	프로세스 정보 탈취 및 조작	<ul style="list-style-type: none"><li>• 프로세스 목록 탈취</li><li>• 프로세스 강제 종료 (최신 버전은 제거됨)</li></ul>



# 에이전트 테슬라에서 수집된 데이터를 전송하는 방식

Agent Tesla 3.2.7.0 English

MAIN | **LOGGER** | PASSWORD RECOVERY | SETTINGS | OTHERS | BUILD | EXPLOIT

**SEND OPTIONS**

LOG OPTIONS

☐ Web Panel ☒ SMTP ☐ FTP

Web Panel

Panel Link: \_\_\_\_\_ DEFAULT **BROWSE**

SMTP

E-mail: Example@gmail.com Example@gmail.com

Password: \*\*\*\*\* smtp.gmail.com

Port: 587 ☒ SSL **TEST E-MAIL**

FTP

Host: \_\_\_\_\_

Username: \_\_\_\_\_

Password: \_\_\_\_\_ **TEST FTP**

© 2013 - 2017 Agent Tesla

DISCORD SKYPE SEND MAIL WEBSITE YOUTUBE

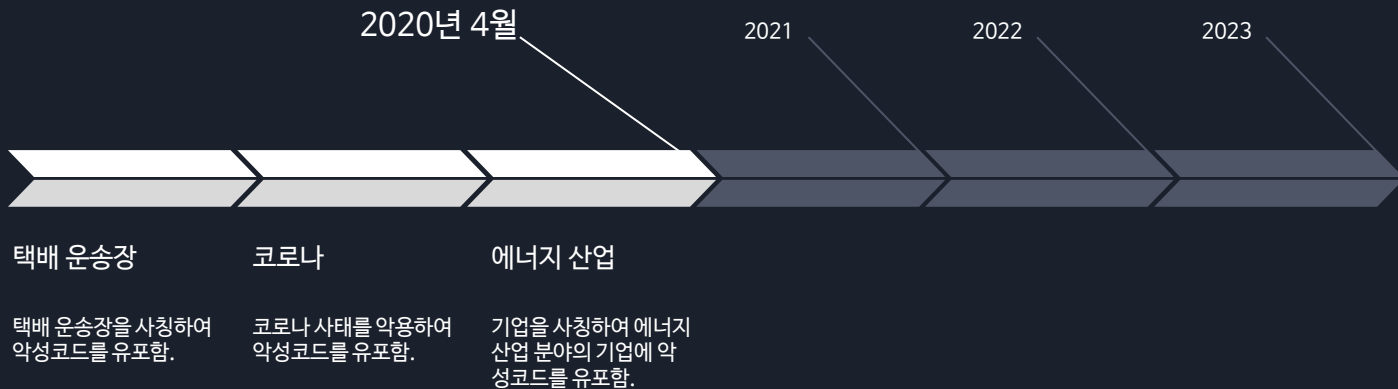


# 에이전트 테슬라 변종

# 에이전트 테슬라 변종

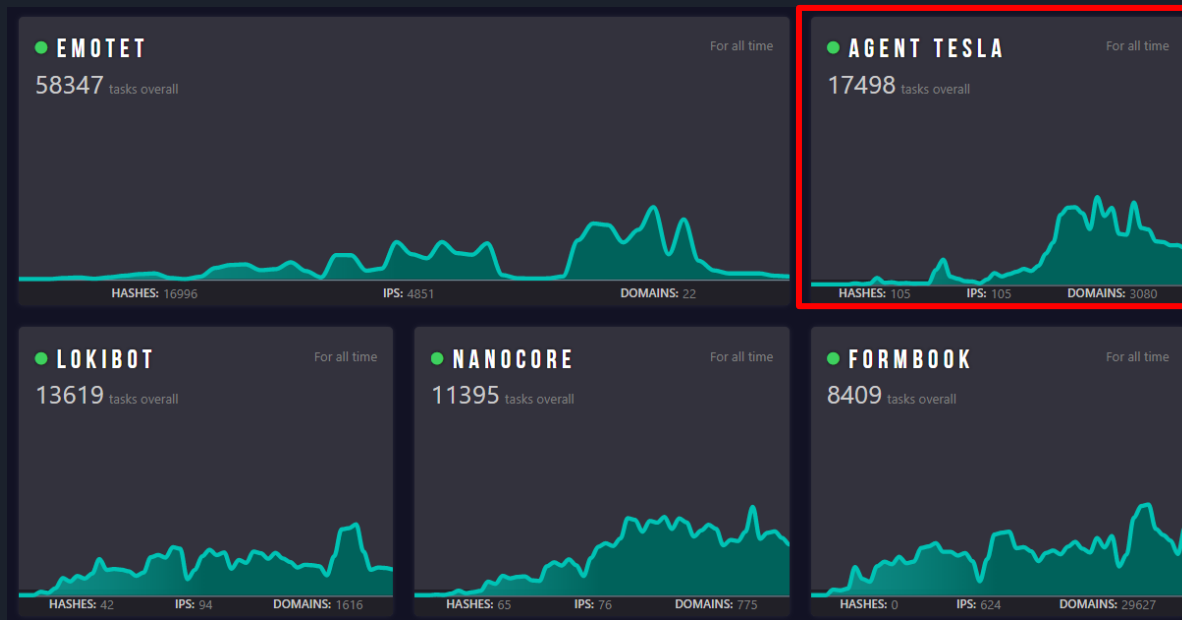
## 2020년 4월 대규모 공격 이행

❖ 화제가 되는 키워드를 이용한 지능형 스피어 피싱 공격



# 에이전트 테슬라 변종 주요 악성코드의 시기별 공격 동향

❖ (2020.6.17) ANY.RUN 통계 결과



전체 공격 순위: **2위**      분석 요청 횟수: **17498회**

에이전트 테슬라 변종

# 에이전트 테슬라 변종의 공격 과정

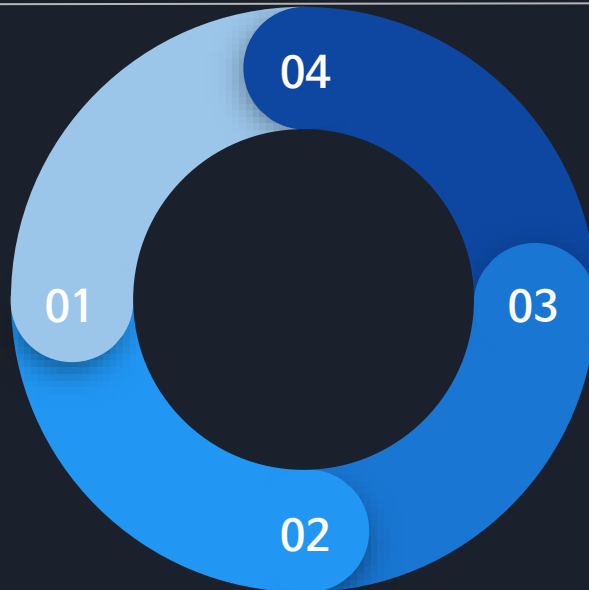
❖ 이메일 등을 통한 최초 유입 이후 백도어 실행, 정보 탈취

## 1단계. 스피어피싱

이메일을 통한 유입  
문서, 실행 파일 형태로 배포

## 2단계. AutoIt 런처

Autoit Script로 작성된 드로퍼를 통  
해 암호화된 .Net 스파이웨어를 실행



## 4단계. 정보 전송

과정 3에서 탈취한 데이터를  
압축하여 공격자의 메일로 전송

## 3단계. .NET 스파이웨어

지속적인 정보 탈취  
탈취 목록은 브라우저 정보,  
화면 캡처, 키로깅으로 확인 가능

## VirusTotal 탐지 결과

[illegible][illegible]

MD5 : a53b3f6a7e651f79bf3e651393db66c8  
최초 업로드 : 2020-04-06 07:54:46

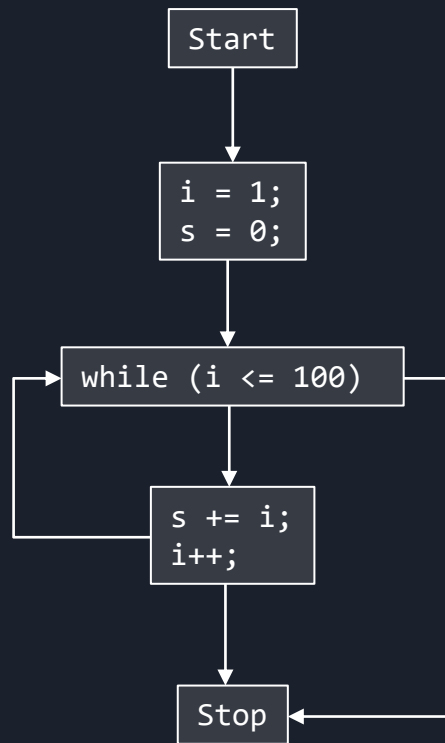
[illegible]

## 2단계. Autolt 런처

## Autoit Script로 작성된 드로퍼를 통해 암호화된 .Net 스파이웨어 실행

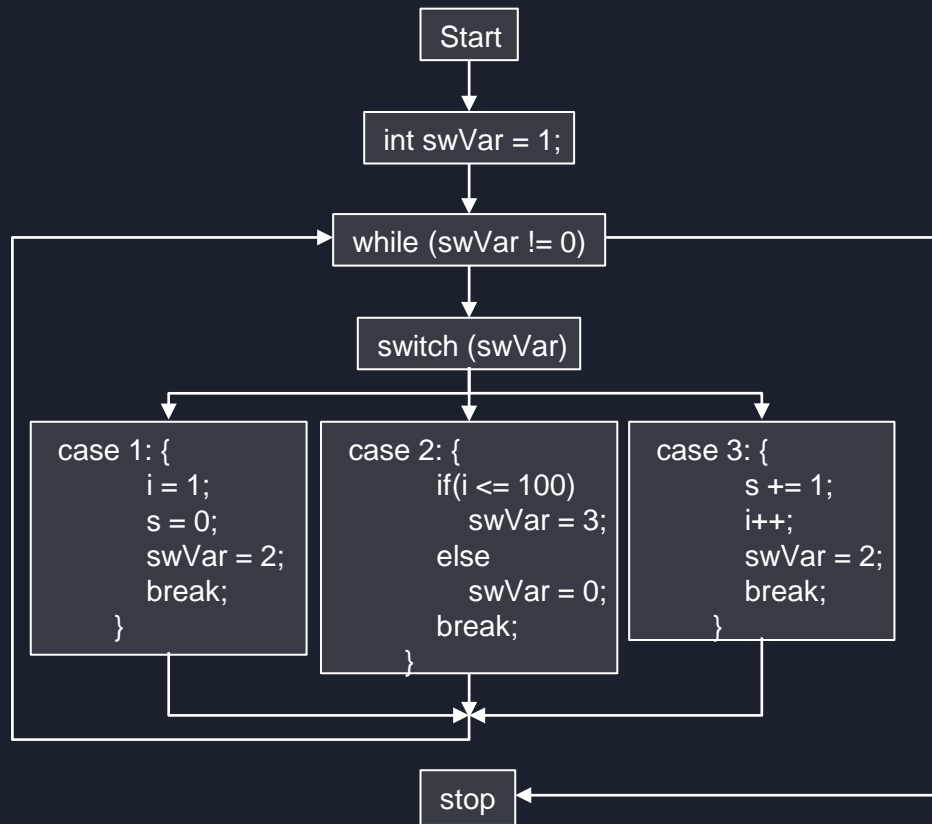
# Control Flow Flattening 적용 전

```
i = 1;  
s = 0;  
  
while(i <= 100) {  
    s += i;  
    i++;  
}
```



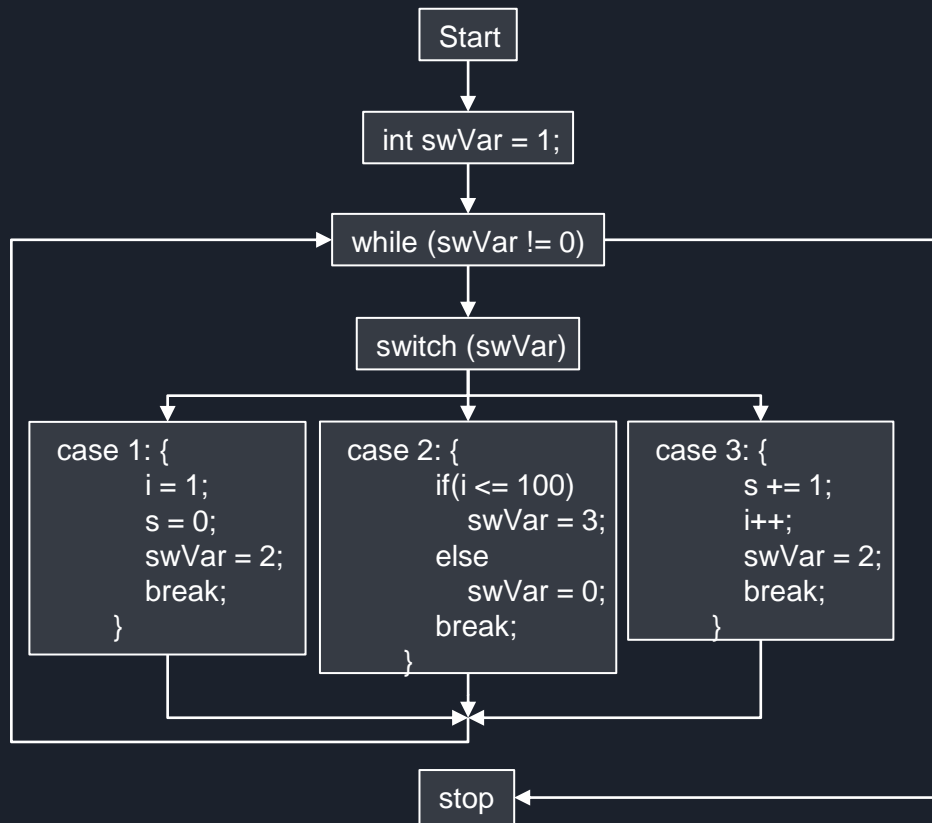
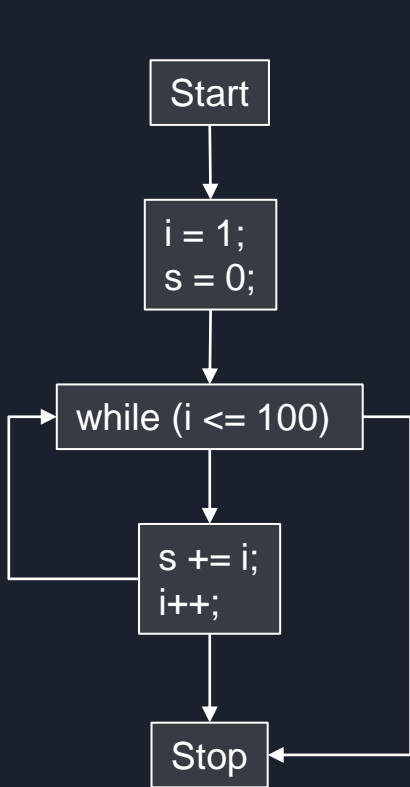
# Control Flow Flattening 적용 후

```
int swVar = 1;
while (swVar != 0) {
    switch(swVar){
        case 1: {
            i = 1;
            s = 0;
            swVar = 2;
            break;
        }
        case 2: {
            if(i <= 100)
                swVar = 3;
            else
                swVar = 0;
            break;
        }
        case 3: {
            s += 1;
            i++;
            swVar = 2;
            break;
        }
    }
}
```





# Control Flow Flattening 적용 비교



## 3단계 .NET 에이전트 테슬라 샘플

Assembly Explorer

- mscorlib (4.0.0.0)
- System (4.0.0.0)
- System.Core (4.0.0.0)
- System.Xml (4.0.0.0)
- System.Xaml (4.0.0.0)
- WindowsBase (4.0.0.0)
- PresentationCore (4.0.0.0)
- PresentationFramework (4.0.0.0)
- dnlib (3.2.0.0)
- dnSpy (6.0.5.0)
- jFssqLDrNpOAdPEoqCGINhUBcCptM
  - JFssqLDrNpOAdPEoqCGINhUBcC
    - PE
    - References
    - { }
      - <Module> @02000001
        - amt @02000034
        - bve @02000040
        - cve @0200003C
        - fgx @0200002E
        - fvg @02000027
        - jkp @02000010
        - jrlb @0200004D
        - jrlc @0200004C
        - jrlj @02000048
        - jrlh @02000048
        - jrlm @02000047
        - jrls @02000046
        - jrlu @02000049
        - jrlv @0200004E
        - jrlw @0200004A
        - jroc @02000045
        - kpf @02000022
        - lsm @02000024
        - luv @02000025
        - lvt @02000026

jkp X

```
171 // Token: 0x06000048 RID: 72 RVA: 0x00019F7C File Offset: 0x0001817C
172 [STAThread]
173 public static void jkm()
174 {
175     jkp.rfz = rub.ruv();
176     jkp.rfh = SystemInformation.UserName + "/" + SystemInformation.ComputerName;
177     jkp.rqc = Environment.GetEnvironmentVariable(<Module>.Wu202E(124540)) + <Module>.Wu202E(124520);
178     for (;;)
179     {
180         IL_46:
181         uint num = 4184331332u;
182         for (;;)
183         {
184             uint num2;
185             switch ((num2 = (num ^ 3815713949u)) % 5u)
186             {
187             case 0u:
188                 if (!Directory.Exists(Environment.GetEnvironmentVariable(<Module>.Wu202E(124500)) + <Module>.Wu202E(124608)))
189                 {
190                     num = (num2 + 2241215261u ^ 1062541242u);
191                     continue;
192                 }
193                 goto IL_139;
194             case 2u:
195                 if (Operators.CompareString(jkp.rqx, jkp.rqc, false) != 0)
196                 {
197                     num = (num2 + 2372208955u ^ 393575035u);
198                     continue;
199                 }
200                 goto IL_4E5;
201             case 3u:
202                 goto IL_46;
203             case 4u:
204                 jkp.jyn();
205                 jkp.rax = Assembly.GetExecutingAssembly().Location;
206                 if (jkp.rqb)
207                 {
208                     num = (num2 + 280000109u ^ 21795683u);
209                     continue;
210                 }
211             }
```

MD5 : 57A39AD30C112D377B936F0C7BEA3125

컴파일 시간 : 2020-03-03 05:20:05

## Control Flow Flattening 적용 모습

```
switch ((num2 = (num ^ 3815713949u)) % 5u)
{
    case 0u:
        if (!Directory.Exists(Environment.GetEnvironmentVariable(<Module>.Wu202E(124500)) + <Module>.Wu202E(124608)))
        {
            num = (num2 + 2241215261u ^ 1062541242u);
            continue;
        }
        goto IL_139;
    case 2u:
        if (Operators.CompareString(jkp.rqx, jkp.rqc, false) != 0)
        {
            num = (num2 + 2372208955u ^ 393575035u);
            continue;
        }
        goto IL_4E5;
    case 3u:
        goto IL_46;
    case 4u:
        jkp.jyn();
        jkp.rqx = Assembly.GetExecutingAssembly().Location;
        if (jkp.rqb)
        {
            num = (num2 + 280000109u ^ 21795683u);
            continue;
        }
}
```

### 3단계 .NET 에이전트 테슬라 샘플

## 문자열 난독화 적용 모습

```
jkp.rqc = Environment.GetEnvironmentVariable(<Module>.\u202E(124540)) + <Module>.\u202E(124520);  
for (;;)   
{  
    IL_46:  
    uint num = 4184331332u;  
    for (;;)   
    {  
        uint num2;  
        switch ((num2 = (num ^ 3815713949u)) % 5u)  
        {  
            case 0u:  
                if (!Directory.Exists(Environment.GetEnvironmentVariable(<Module>.\u202E(124500)) + <Module>.\u202E(124608)))  
                {  
                    num = (num2 * 2241215261u ^ 1062541242u);  
                    continue;  
                }  
                goto IL_139;  
            case 2u:  
                if (Operators.CompareString(jkp.rqx, jkp.rqc, false) != 0)  
                {  
                    num = (num2 * 2372208955u ^ 393575035u);  
                    continue;  
                }  
                goto IL_4E5;  
        }  
    }  
}
```

# de4dot 도구 이용 : CFF 난독화 해제 시도

```
C:\Users\Asus-Agent17\Desktop>de4dot.exe sample1 --only-cflow-deob --strtyp emulate --strtok 06000002 -v
```

```
de4dot v3.1.41592.3405 Copyright (C) 2011-2015 de4dot@gmail.com  
Latest version and source code: https://github.com/0xd4d/de4dot
```

```
1: Unknown  
0: Agile.NET  
0: Babel .NET  
0: CodeFort  
0: CodeVeil  
0: CodeWall  
Deobfuscating control flow: System.Void <Module>::.cctor() (06000001)  
0: Confuser  
0: Crypto Obfuscator  
Deobfuscating control flow: System.Void <Module>::.cctor() (06000001)  
Deobfuscating control flow: System.Void jkp::jyn() (0600004B)  
Deobfuscating control flow: System.Void jkp::jlt() (06000056)  
Deobfuscating control flow: System.Void jkp::jbg() (06000061)  
0: DeepSea  
0: Dotfuscator  
0: .NET Reactor 3.x  
Deobfuscating control flow: System.Void jkp::jkm() (06000048)  
Deobfuscating control flow: System.Void jkp::cctor() (06000045)  
0: .NET Reactor 4.x  
0: Eazfuscator.NET  
0: Goliath.NET  
0: ILProtector  
0: MaxtoCode  
0: MPRESS  
0: Rummage  
0: Skater .NET  
0: SmartAssembly  
0: Spices.Net  
0: Xenocode  
Detected Unknown Obfuscator (C:\Users\Asus-Agent17\Desktop\sample1)
```

# de4dot 도구 이용 : CFF 난독화 해제 실패

```
C:\Users\Asus-Agent17\Desktop>de4dot.exe sample1 --only-cflow-deob --strtyp emulate --strtok 06000002 -v
```

```
de4dot v3.1.41592.3405 Copyright (C) 2011-2015 de4dot@gmail.com  
Latest version and source code: https://github.com/0xd4d/de4dot
```

```
1: Unknown  
0: Agile.NET  
0: Babel .NET  
0: CodeFort  
0: CodeVeil  
0: CodeWall
```

Confuser 패턴 매칭 실패

de4dot을 통한 CFF 난독화 해제 옵션  
--only-cflow-deob

```
Deobfuscating control flow: System.Void <Module>::.cctor() (06000001)  
0: Confuser  
0: Crypto Obfuscator  
Deobfuscating control flow: System.Void <Module>::.cctor() (06000001)  
Deobfuscating control flow: System.Void jkp::jyn() (0600004B)  
Deobfuscating control flow: System.Void jkp::jlt() (06000056)  
Deobfuscating control flow: System.Void jkp::jbg() (06000061)  
0: DeepSea  
0: Dotfuscator  
0: .NET Reactor 3.x  
Deobfuscating control flow: System.Void jkp::jkm() (06000048)  
Deobfuscating control flow: System.Void jkp::cctor() (06000045)  
0: .NET Reactor 4.x  
0: Eazfuscator.NET  
0: Goliath.NET  
0: ILProtector  
0: MaxtoCode  
0: MPRESS  
0: Rummage  
0: Skater .NET  
0: SmartAssembly  
0: Spices.Net  
0: Xenocode  
Detected Unknown Obfuscator (C:\Users\Asus-Agent17\Desktop\sample1)
```

3단계 .NET 에이전트 테슬라 샘플

## de4dot 도구 이용 : 문자열 난독화 해제 시도

```
C:\Users\Asus-Agent17\Desktop>de4dot.exe sample1 --only-cflow-deob --strtyp emulate --strtok 06000002 -v
```

```
de4dot v3.1.41592.3405 Copyright (C) 2011-2015 de4dot@gmail.com  
Latest version and source code: https://github.com/0xd4d/de4dot
```

```
1: Unknown  
0: Agile.NET  
0: Babel .NET  
0: CodeFort  
0: CodeVeil  
0: CodeWall
```

```
Deobfuscating control flow: System.Void <Module>::.cctor() (06000001)
```

```
0: Confuser  
0: Crypto Obfuscator
```

```
Deobfuscating control flow: System.Void <Module>::.cctor() (06000001)
```

```
Deobfuscating control flow: System.Void jkp::jyn() (0600004B)
```

```
Deobfuscating control flow: System.Void jkp::jlt() (06000056)
```

```
Deobfuscating control flow: System.Void jkp::jbg() (06000061)
```

```
0: DeepSea  
0: Dotfuscator  
0: .NET Reactor 3.x
```

```
Deobfuscating control flow: System.Void jkp::jkm() (06000048)
```

```
Deobfuscating control flow: System.Void jkp::cctor() (06000045)
```

```
0: .NET Reactor 4.x  
0: Eazfuscator.NET  
0: Goliath.NET  
0: ILProtector  
0: MaxtoCode  
0: MPRESS  
0: Rummage  
0: Skater .NET  
0: SmartAssembly  
0: Spices.Net  
0: Xenocode
```

```
Detected Unknown Obfuscator (C:\Users\Asus-Agent17\Desktop\sample1)
```

de4dot을 통한 리소스 해제 옵션

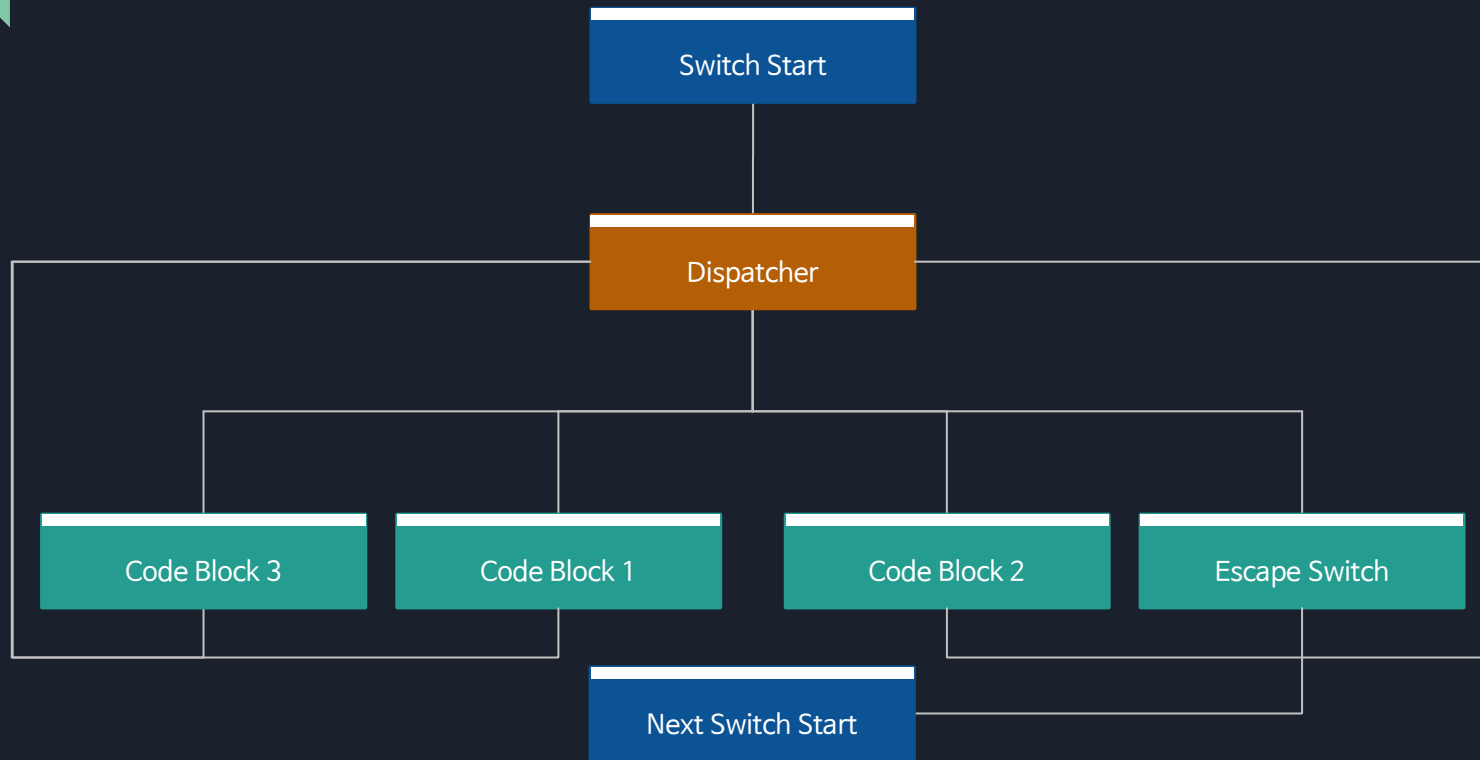
```
--strtyp emulate --strtok [MD token]
```

## de4dot 도구 이용 : 문자열 난독화 해제 결과

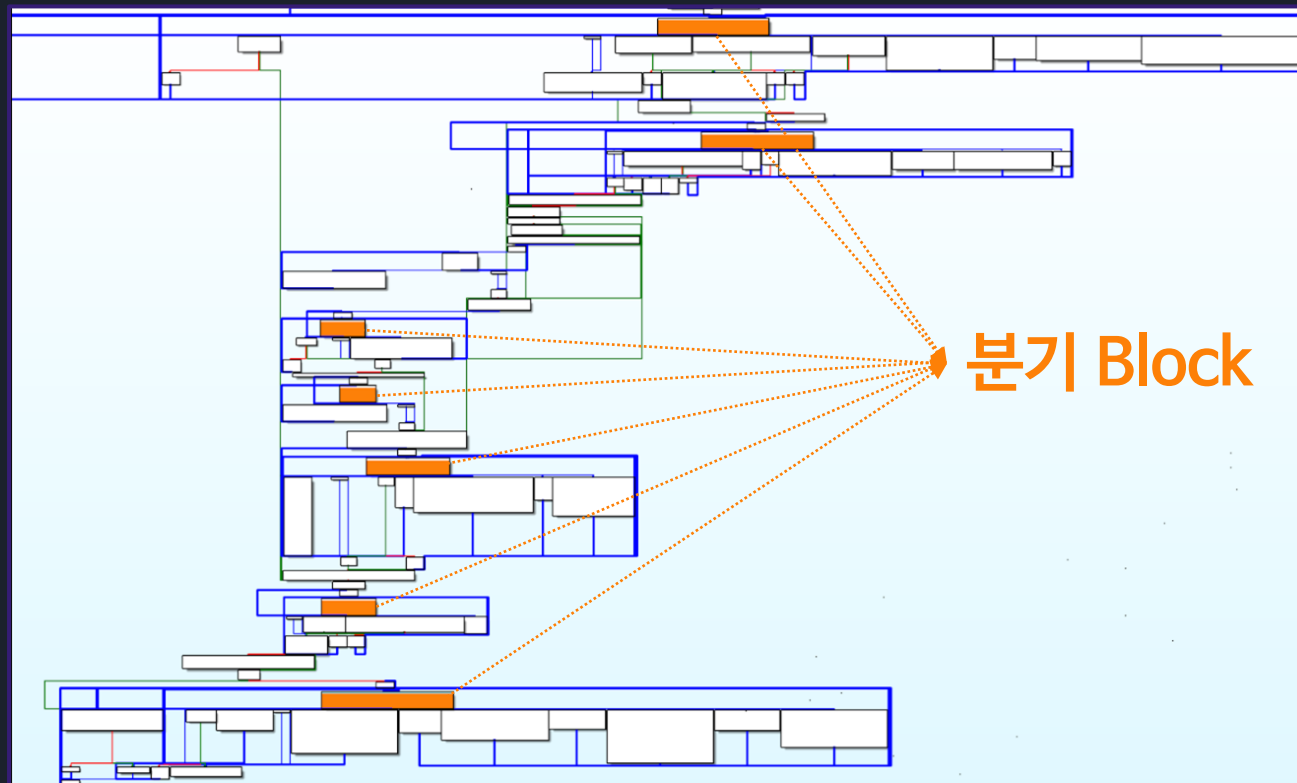
```
public static void jkm()  
{  
    jkp.rfz = rub.ruv();  
    jkp.rfh = SystemInformation.UserName + "/" + SystemInformation.ComputerName;  
    jkp.rqc = Environment.GetEnvironmentVariable("%startupfolder%") + "%%insfolder%%insname%";  
    for (;;) 리소스 영역 복호화 성공  
    {  
        IL_F0:  
        uint num = 4184331332u;  
        for (;;) 리소스 영역 복호화 성공  
        {  
            uint num2;  
            switch ((num2 = (num ^ 3815713949u)) % 5u)  
            {  
            case 0u:  
                if (!Directory.Exists(Environment.GetEnvironmentVariable("%startupfolder%") + "%%insfolder%%"))  
                {  
                    num = (num2 + 2241215261u ^ 1062541242u);  
                    continue;  
                }  
                goto IL_111;  
            }  
        }  
    }  
}
```



# 실제 샘플 CFF Basic Block 구조



## CFF가 적용된 메소드 전체의 Basic Block 그래프



# CFF Switch Block 시작 부분

```
.method private instance void wxt(valuetype mhe& dcb)
{
    .maxstack 6
    .locals init (int64 V0,
                 unsigned int32 V1)
    ldarg.0
    ldflld class [mscorlib]System.IO.Stream mht::dgcx
    callvirt instance int64 [mscorlib]System.IO.Stream::get_Position()
    stloc.0
    ldarg.0
    ldflld class [mscorlib]System.IO.Stream mht::dgcx
    ldarg.1
    ldflld unsigned int32 mhe::HeaderOffset
    conv.u8
    ldc.i4.8
    conv.i8
    add.ovf
    callvirt instance void [mscorlib]System.IO.Stream::set_Position(int64)
```

loc\_295C1:  
ldc.i4 0x9E9627AF

초기값 설정

loc\_295C6:  
ldc.i4 0xC1500A43  
xor  
dup  
stloc 1  
ldc.i4 3  
rem.un  
switch loc\_295C1, loc\_295ED, loc\_29660

# CFF 다음 Switch Block 설정

초기값 설정

```
loc_295C6:  
ldc.i4 0xC1500A43  
xor  
dup  
stloc 1  
ldc.i4 3  
rem.un  
switch loc_295C1, loc_295ED, loc_29660  
  
loc_295ED:  
ldarg.0  
ldfld class [mscorlib]System.IO.Stream mht::dgc  
ldarg.1  
ldfld valuetype mhw mhe::Method  
call unsigned int8[] [mscorlib]System.BitConverter::GetBytes(unsigned int16)  
ldc.i4.0  
ldc.i4.2  
callvirt instance void [mscorlib]System.IO.Stream::Write(unsigned int8[], int32, int32)  
ldarg.0  
ldfld class [mscorlib]System.IO.Stream mht::dgc  
ldarg.1  
ldfld unsigned int32 mhe::HeaderOffset  
conv.u8  
ldc.i4.s 0xE  
conv.i8  
add.ovf  
callvirt instance void [mscorlib]System.IO.Stream::Set_Position(int64)  
ldarg.0  
ldfld class [mscorlib]System.IO.Stream mht::dgc  
ldarg.1  
ldfld unsigned int32 mhe::Crc32  
call unsigned int8[] [mscorlib]System.BitConverter::GetBytes(unsigned int32)  
ldc.i4.0  
ldc.i4.4  
callvirt instance void [mscorlib]System.IO.Stream::Write(unsigned int8[], int32, int32)  
ldarg.0  
ldfld class [mscorlib]System.IO.Stream mht::dgc  
ldarg.1  
ldfld unsigned int32 mhe::CompressedSize  
call unsigned int8[] [mscorlib]System.BitConverter::GetBytes(unsigned int32)  
ldc.i4.0  
ldc.i4.4  
callvirt instance void [mscorlib]System.IO.Stream::Write(unsigned int8[], int32, int32)  
ldloc 1  
ldc.i4 0xB8AA7474  
mul  
ldc.i4 0xB803822B  
xor  
br loc_295C6
```

실제 Code Block 및  
다음 값 설정

## 문자열 난독화에서 사용하는 복호화 메소드 분석

```
internal static string StaticMethod2(int A_0)
{
    object[] staticField = Class1.StaticField1;
    if (Assembly.GetExecutingAssembly() == Assembly.GetExecutingAssembly())
    {
        byte[] array = new byte[32];
        byte[] array2 = new byte[16];
        int num = 32;
        int num2 = 16;
        int num3 = 2;
        int num4 = 5;
        int num5 = 588;
        int num6 = 6019;
        int num7 = A_0 >> num3;
        num7 = num7 - num4 + num5 - 26698;
        num7 = (num7 ^ num5 ^ num6);
        num7 -= 831;
        num7 = (num7 - num5) / num4;
        uint[] array3 = (uint[])staticField[num7];
        byte[] array4 = new byte[array3.Length * 4];
        Buffer.BlockCopy(array3, 0, array4, 0, array3.Length * 4);
        byte[] array5 = array4;
        int num8 = array5.Length - (num + num2);
        byte[] array6 = new byte[num8];
        Buffer.BlockCopy(array5, 0, array, 0, num);
        Buffer.BlockCopy(array5, num, array2, 0, num2);
        Buffer.BlockCopy(array5, num + num2, array6, 0, num8);
        return Encoding.UTF8.GetString(Class1.AESDecrypt(array6, array, array2));
    }
    return "";
}
```

## 문자열 난독화에서 사용하는 복호화 메소드 분석

```
internal static string StaticMethod2(int A_0)
{
    object[] staticField = Class1.StaticField1;
    if (Assembly.GetExecutingAssembly() == Assembly.GetExecutingAssembly())
    {
        byte[] array = new byte[32];
        byte[] array2 = new byte[16];
        int num = 32;
        int num2 = 16;
        int num3 = 2;
        int num4 = 5;
        int num5 = 588;
        int num6 = 6019;
        int num7 = A_0 >> num3;
        num7 = num7 - num4 + num5 - 26698;
        num7 = (num7 ^ num5 ^ num6);
        num7 -= 831;
        num7 = (num7 - num5) / num4;
        uint[] array3 = (uint[])staticField[num7];
        byte[] array4 = new byte[array3.Length * 4];
        Buffer.BlockCopy(array3, 0, array4, 0, array3.Length * 4);
        byte[] array5 = array4;
        int num8 = array5.Length - (num + num2);
        byte[] array6 = new byte[num8];
        Buffer.BlockCopy(array5, 0, array, 0, num);
        Buffer.BlockCopy(array5, num, array2, 0, num2);
        Buffer.BlockCopy(array5, num + num2, array6, 0, num8);
        return Encoding.UTF8.GetString(Class1.AESDecrypt(array6, array, array2));
    }
    return "";
}
```

## 문자열 난독화에서 사용하는 복호화 메소드 분석

```
internal static string StaticMethod2(int A_0)
{
    object[] staticField = Class1.StaticField1;
    if (Assembly.GetExecutingAssembly() == Assembly.GetExecutingAssembly())
    {
        byte[] array = new byte[32];
        byte[] array2 = new byte[16];
        int num = 32;
        int num2 = 16;
        int num3 = 2;
        int num4 = 5;
        int num5 = 588;
        int num6 = 6019;
        int num7 = A_0 >> num3;
        num7 = num7 - num4 + num5 - 26698;
        num7 = (num7 ^ num5 ^ num6);
        num7 -= 831;
        num7 = (num7 - num5) / num4;
        uint[] array3 = (uint[])staticField[num7];
        byte[] array4 = new byte[array3.Length * 4];
        Buffer.BlockCopy(array3, 0, array4, 0, array3.Length * 4);
        byte[] array5 = array4;
        int num8 = array5.Length - (num + num2);
        byte[] array6 = new byte[num8];
        Buffer.BlockCopy(array5, 0, array, 0, num);
        Buffer.BlockCopy(array5, num, array2, 0, num2);
        Buffer.BlockCopy(array5, num + num2, array6, 0, num8);
        return Encoding.UTF8.GetString(Class1.AESDecrypt(array6, array, array2));
    }
    return "";
}
```

## 문자열 난독화에서 사용하는 복호화 메소드 분석

```
internal static string StaticMethod2(int A_0)
{
    object[] staticField = Class1.StaticField1;
    if (Assembly.GetExecutingAssembly() == Assembly.GetExecutingAssembly())
    {
        byte[] array = new byte[32];
        byte[] array2 = new byte[16];
        int num = 32;
        int num2 = 16;
        int num3 = 2;
        int num4 = 5;
        int num5 = 588;
        int num6 = 6019;
        int num7 = A_0 >> num3;
        num7 = num7 - num4 + num5 - 26698;
        num7 = (num7 ^ num5 ^ num6);
        num7 -= 831;
        num7 = (num7 - num5) / num4;
        uint[] array3 = (uint[])staticField[num7];
        byte[] array4 = new byte[array3.Length + 4];
        Buffer.BlockCopy(array3, 0, array4, 0, array3.Length + 4);
        byte[] array5 = array4;
        int num8 = array5.Length - (num + num2);
        byte[] array6 = new byte[num8];
        Buffer.BlockCopy(array5, 0, array, 0, num);
        Buffer.BlockCopy(array5, num, array2, 0, num2);
        Buffer.BlockCopy(array5, num + num2, array6, 0, num8);
        return Encoding.UTF8.GetString(Class1.AESDecrypt(array6, array, array2));
    }
    return "";
}
```



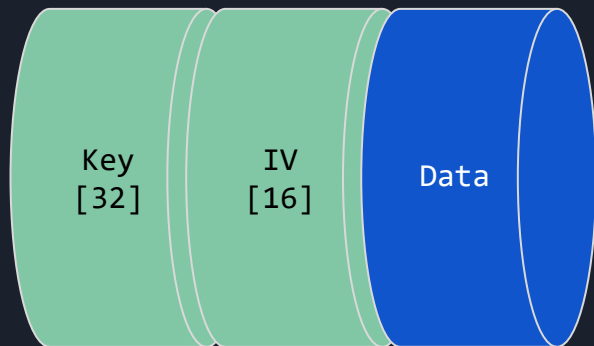
## 문자열 난독화에서 사용하는 복호화 메소드 분석

```
internal static string StaticMethod2(int A_0)
{
    object[] staticField = Class1.StaticField1;
    if (Assembly.GetExecutingAssembly() == Assembly.GetExecutingAssembly())
    {
        byte[] array = new byte[32];
        byte[] array2 = new byte[16];
        int num = 32;
        int num2 = 16;
        int num3 = 2;
        int num4 = 5;
        int num5 = 588;
        int num6 = 6019;
        int num7 = A_0 >> num3;
        num7 = num7 - num4 + num5 - 26698;
        num7 = (num7 ^ num5 ^ num6);
        num7 -= 831;
        num7 = (num7 - num5) / num4;
        uint[] array3 = (uint[])staticField[num7];
        byte[] array4 = new byte[array3.Length * 4];
        Buffer.BlockCopy(array3, 0, array4, 0, array3.Length * 4);
        byte[] array5 = array4;
        int num8 = array5.Length - (num + num2);
        byte[] array6 = new byte[num8];
        Buffer.BlockCopy(array5, 0, array, 0, num);
        Buffer.BlockCopy(array5, num, array2, 0, num2);
        Buffer.BlockCopy(array5, num + num2, array6, 0, num8);
        return Encoding.UTF8.GetString(Class1.AESDecrypt(array6, array, array2));
    }
    return "";
}
```

CFF 난독화 해제 상태

## 문자열 난독화에서 사용하는 복호화 메소드 분석

```
buffer.BlockCopy(array5, 0, array, 0, num);  
Buffer.BlockCopy(array5, num, array2, 0, num2);  
Buffer.BlockCopy(array5, num + num2, array6, 0, num8);  
return Encoding.UTF8.GetString(Class1.AESDecrypt(array6, array, array2));  
}  
return "";
```

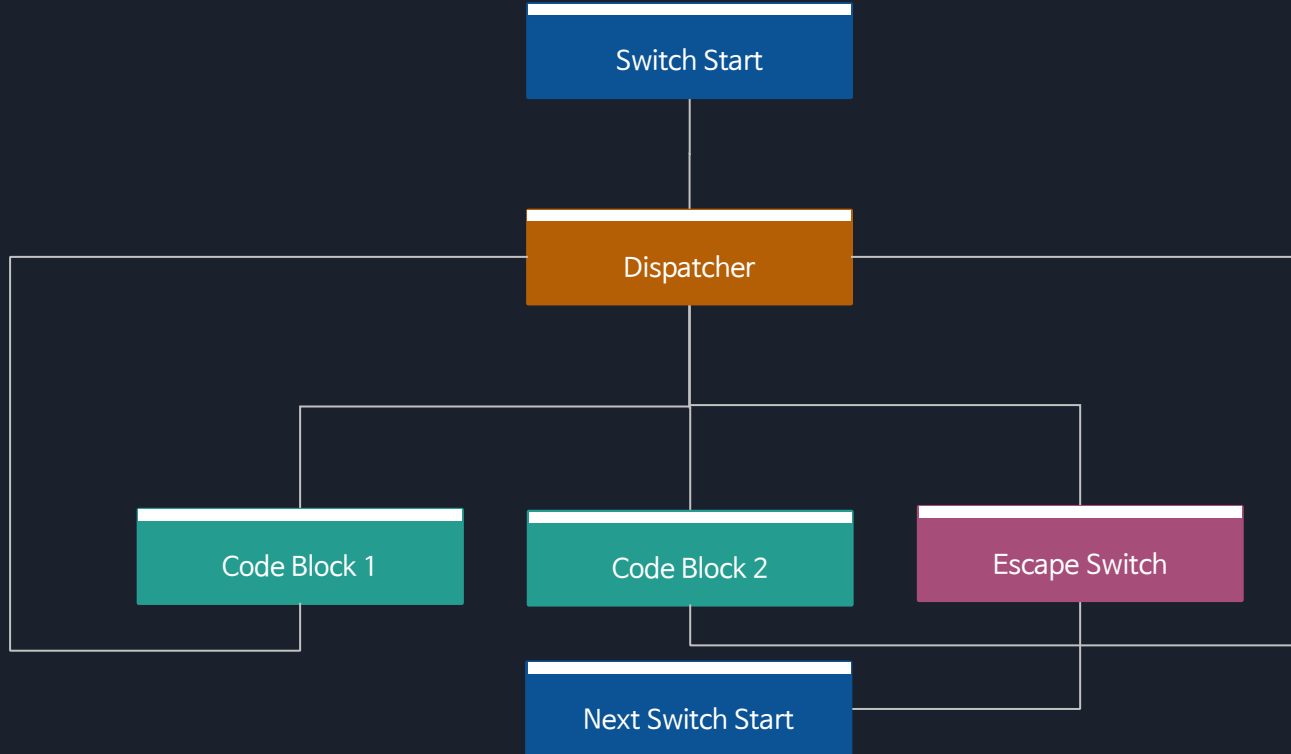


암호화된 데이터 배열 : Key[32] + IV[16] + DATA[??]

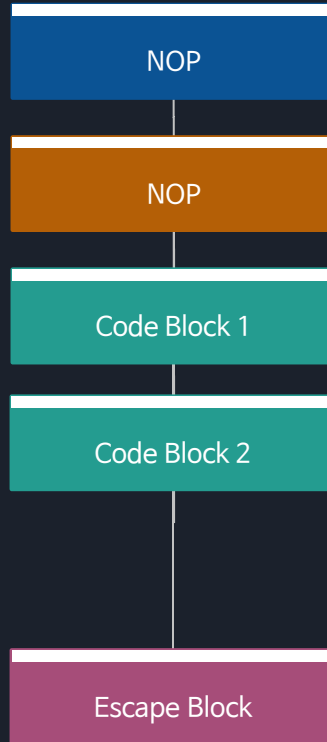


# 에이전트 테슬라 변종 : 난독화 분석 및 해제 기법

# CFF 난독화 분석 : 난독화 적용 상태



## CFF 난독화 분석 : 난독화 해제 상태



## 초기 값 식별

```
for (;;)
{
    IL_260:
    uint num5 = 3818295590u;
    for (;;)
    {
        uint num2;
        switch ((num2 = (num5 ^ 3815713949u)) % 3u)
        {
            case 0u:
                goto IL_260;
            case 1u:
                num5 = (num2 * 3896289847u ^ 354163349u);
                continue;
        }
        goto Block_27;
    }
}
Block_27::
}
```

NOP

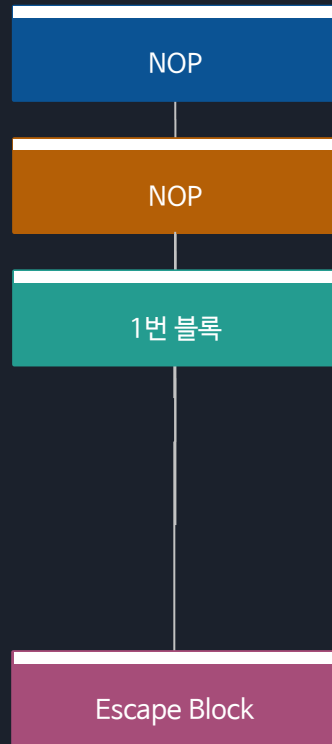
NOP

Escape Block

## 최초 실행 Block 식별

```
for (;;)
{
    IL_260:
    uint num5 = 3818295590u;
    for (;;)
    {
        uint num2;
        switch ((num2 = (num5 ^ 3815713949u)) % 3u)
        {
            case 0u:
                goto IL_260;
            case 1u:
                num5 = (num2 * 3896289847u ^ 354163349u);
                continue;
        }
        goto Block_27;
    }
}
Block_27::
}
```

연산 결과 : 1  
num5 : 3818295590



# Switch 인자 변경

```
for (;;)
{
    IL_260:
    uint num5 = 3818295590u;
    for (;;)
    {
        uint num2;
        switch ((num2 = (num5 ^ 3815713949u)) % 3u)
        {
            case 0u:
                goto IL_260;
            case 1u:
                num5 = (num2 * 3896289847u ^ 354163349u);
                continue;
        }
        goto Block_27;
    }
}
Block_27::
}
```

연산 결과 : 1  
num5 : 3919325671





## 다음 Block 식별

```
for (;;)
{
    IL_260:
    uint num5 = 3818295590u;
    for (;;)
    {
        uint num2;
        switch ((num2 = (num5 ^ 3815713949u)) % 3u)
        {
            case 0u:
                goto IL_260;
            case 1u:
                num5 = (num2 * 3896289847u ^ 354163349u);
                continue;
        }
        goto Block_27;
    }
}
Block_27::
}
```

연산 결과 : 2  
num5 : 3919325671



# Block Escape & 코드 실행

```
for (;;)
{
    IL_260:
    uint num5 = 3818295590u;
    for (;;)
    {
        uint num2;
        switch ((num2 = (num5 ^ 3815713949u)) % 3u)
        {
            case 0u:
                goto IL_260;
            case 1u:
                num5 = (num2 * 3896289847u ^ 354163349u);
                continue;
        }
        goto Block_27;
    }
}
Block_27::
}
```

연산 결과 : 2  
num5 : 3919325671



## CFF 난독화 분석

# CFF 난독화 해제 전/후 비교

```
uint num = 1463824866u;
for (;;)
{
    uint num2;
    switch ((num2 = (num ^ 1079817129u)) % 11u)
    {
        case 0u:
            goto IL_18A;
        case 1u:
        {
            int num4;
            int num5;
            int num3 = num3 ^ num4 ^ num5;
            num3 -= 831;
            num = (num2 * 1441864503u ^ 3634836150u);
            continue;
        }
        case 2u:
        {
            int num4;
            int num6;
            int num3 = num3 - num6 + num4 - 26698;
            num = (num2 * 75269128u ^ 519288486u);
            continue;
        }
        case 3u:
        {
            uint[] array;
            Buffer.BlockCopy(array, 0, array2, 0, array.Length * 4);
            num = (num2 * 3037318715u ^ 1504203551u);
        }
    }
}
```

난독화 해제 전

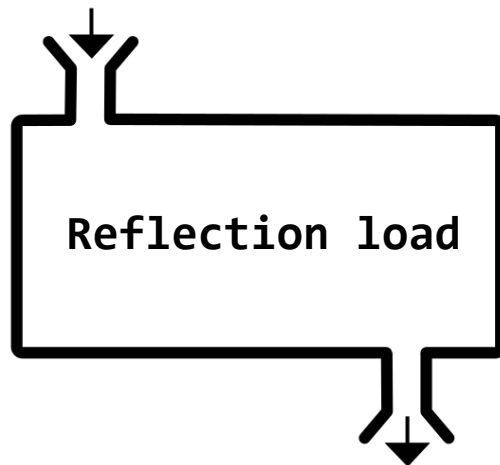
```
internal static string StaticMethod2(int A_0)
{
    object[] staticField = Class1.StaticField1;
    if (Assembly.GetExecutingAssembly() == Assembly.GetExecutingAssembly())
    {
        byte[] array = new byte[32];
        byte[] array2 = new byte[16];
        int num = 32;
        int num2 = 16;
        int num3 = 2;
        int num4 = 5;
        int num5 = 588;
        int num6 = 6019;
        int num7 = A_0 >> num3;
        num7 = num7 - num4 + num5 - 26698;
        num7 = (num7 ^ num5 ^ num6);
        num7 -= 831;
        num7 = (num7 - num5) / num4;
        uint[] array3 = (uint[])staticField[num7];
        byte[] array4 = new byte[array3.Length * 4];
        Buffer.BlockCopy(array3, 0, array4, 0, array3.Length * 4);
        byte[] array5 = array4;
        int num8 = array5.Length - (num + num2);
        byte[] array6 = new byte[num8];
        Buffer.BlockCopy(array5, 0, array, 0, num);
        Buffer.BlockCopy(array5, num, array2, 0, num2);
        Buffer.BlockCopy(array5, num + num2, array6, 0, num8);
        return Encoding.UTF8.GetString(Class1.AESDecrypt(array6, array, array2));
    }
}
```

난독화 해제 후

# 에이전트 테슬라 변종 문자열 난독화 해제

```
internal static string StaticMethod2(int A_0)
{
    object[] staticField = Class1.StaticField1;
    if (Assembly.GetExecutingAssembly() == Assembly.GetExecutingAssembly())
    {
        byte[] array = new byte[32];
        byte[] array2 = new byte[16];
        int num = 32;
        int num2 = 16;
        int num3 = 2;
        int num4 = 5;
        int num5 = 588;
        int num6 = 6019;
        int num7 = A_0 >> num3;
        num7 = num7 - num4 + num5 - 26698;
        num7 = (num7 ^ num5 ^ num6);
        num7 -= 831;
        num7 = (num7 - num5) / num4;
        uint[] array3 = (uint[])staticField[num7];
        byte[] array4 = new byte[array3.Length * 4];
        Buffer.BlockCopy(array3, 0, array4, 0, array3.Length * 4);
        byte[] array5 = array4;
        int num8 = array5.Length - (num + num2);
        byte[] array6 = new byte[num8];
        Buffer.BlockCopy(array5, 0, array, 0, num);
        Buffer.BlockCopy(array5, num, array2, 0, num2);
        Buffer.BlockCopy(array5, num + num2, array6, 0, num8);
        return Encoding.UTF8.GetString(Class1.AESDecrypt(array6, array, array2));
    }
    return "";
}
```

INPUT : 12311



OUTPUT : "Hello World"

## 에이전트 테슬라 변종 문자열 난독화 해제

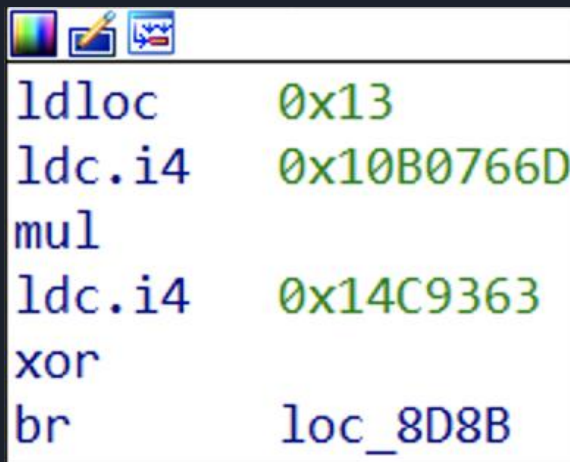
```
if (Assembly.GetExecutingAssembly() == Assembly.GetCallingAssembly())
{
    byte[] array2;
    int num8;
    byte[] array3;
    byte[] array4;
    int num9;
    for (;;)
    {
        IL_18A:
        uint num = 1463824866u;
        for (;;)
        {
            uint num2;
            switch ((num2 = (num ^ 1079817129u)) % 11u)
            {
                case 0u:
                    goto IL_18A;
                case 1u:
                {
                    int num4;
                    int num5;
                    int num3 = num3 ^ num4 ^ num5;
                    num3 -= 831;
                    num = (num2 * 1441864503u ^ 3634836150u);
                    continue;
                }
                case 2u:
```

난독화 해제 전

```
internal static string StaticMethod2(int A_0)
{
    object[] staticField = Class1.StaticField1;
    if (Assembly.GetExecutingAssembly() == Assembly.GetExecutingAssembly())
    {
        byte[] array = new byte[32];
        byte[] array2 = new byte[16];
        int num = 32;
        int num2 = 16;
        int num3 = 2;
        int num4 = 5;
        int num5 = 588;
        int num6 = 6019;
        int num7 = A_0 >> num3;
        num7 = num7 - num4 + num5 - 26698;
        num7 = (num7 ^ num5 ^ num6);
        num7 -= 831;
        num7 = (num7 - num5) / num4;
        uint[] array3 = (uint[])staticField[num7];
        byte[] array4 = new byte[array3.Length * 4];
        Buffer.BlockCopy(array3, 0, array4, 0, array3.Length * 4);
        byte[] array5 = array4;
        int num8 = array5.Length - (num + num2);
        byte[] array6 = new byte[num8];
        Buffer.BlockCopy(array5, 0, array, 0, num);
        Buffer.BlockCopy(array5, num, array2, 0, num2);
        Buffer.BlockCopy(array5, num + num2, array6, 0, num8);
        return Encoding.UTF8.GetString(Class1.AESDecrypt(array6, array, array2));
    }
}
```

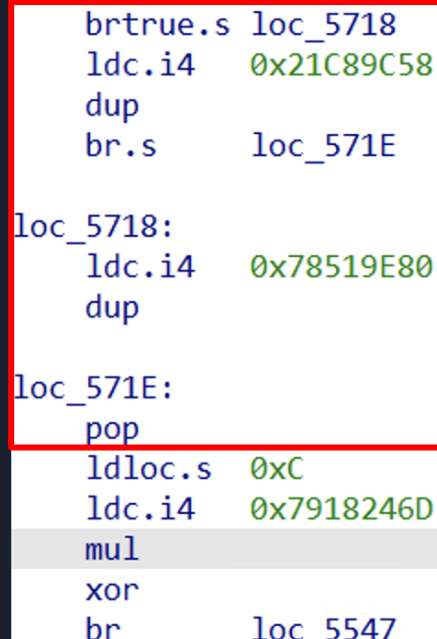
난독화 해제 후

## 유사 샘플에 추가된 패턴



```
ldloc      0x13
ldc.i4     0x10B0766D
mul
ldc.i4     0x14C9363
xor
br         loc_8D8B
```

기존 샘플



```
brtrue.s   loc_5718
ldc.i4     0x21C89C58
dup
br.s       loc_571E

loc_5718:
ldc.i4     0x78519E80
dup

loc_571E:
pop
ldloc.s    0xC
ldc.i4     0x7918246D
mul
xor
br         loc_5547
```

추가 발견 샘플



마치며

# 에이전트 테슬라 변종 다양한 난독화 도구



**PreEmptive** PROTECTION | **Dotfuscator** for .NET

The #1 .NET obfuscation and in-app protection product for 17 years. Community version in Visual Studio since 2003.

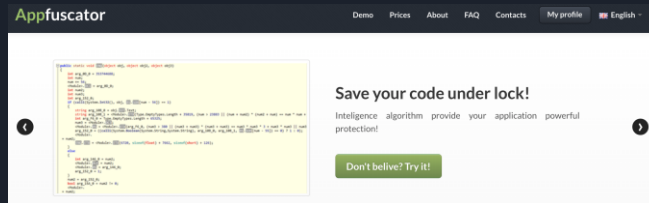
## obfuscator

Obfuscator, The Open Source Obfuscation Tool for .NET Assemblies

[download archive](#)

This project migrated to <https://www.obfuscator.com>

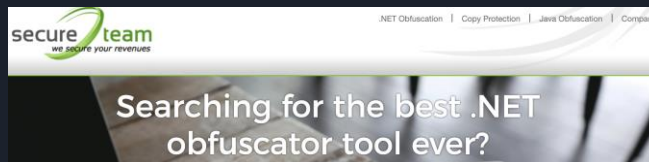
Obfuscator is an open source .NET obfuscator released under MIT license. It provides basic obfuscation features that help secure secrets in a .NET assembly.



**Appfuscator**

Save your code under lock!  
Intelligence algorithm provide your application powerful protection!

[Don't believe? Try it!](#)



**secure team**  
we secure your revenues

Searching for the best .NET obfuscator tool ever?

## Confuser

Confuser

[download archive](#)

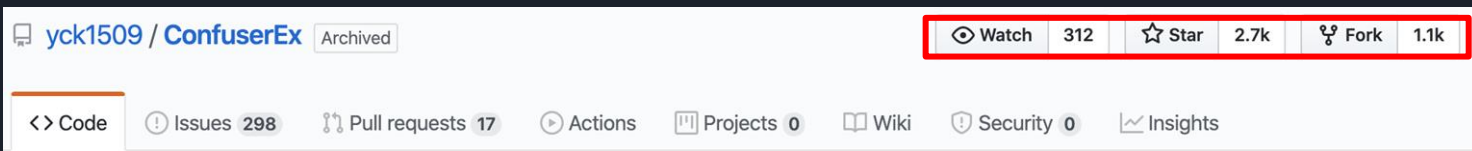
Confuser is a obfuscator for .NET. It is developed in C# and using Mono.Cecil for assembly manipulation.

[home](#) [issues](#) [discussions](#)



# 마치며

- ❖ de4dot의 난독화 해제 코드는 다양한 변종 악성코드 분석에 활용 가능
- ❖ 그러나 CFF 난독화의 경우 다양한 패턴으로 여러 변종이 제작될 수 있어 빠른 대응이 어려움
- ❖ 에이전트 테슬라 샘플에 적용된 CFF는 오픈 소스 난독화 도구 Confuser의 CFF 패턴과 유사함





# 감사합니다

## Q&A

<https://github.com/kookmin-cat/AgentTeslaDeob>

