Misères des antivirus et splendeurs des malwares

Guillaume Bonfante
LORIA
Université de Lorraine

Guillaume.Bonfante@loria.fr

Séminaire « confiance numérique » - Clermont-Ferrand, juin 2015

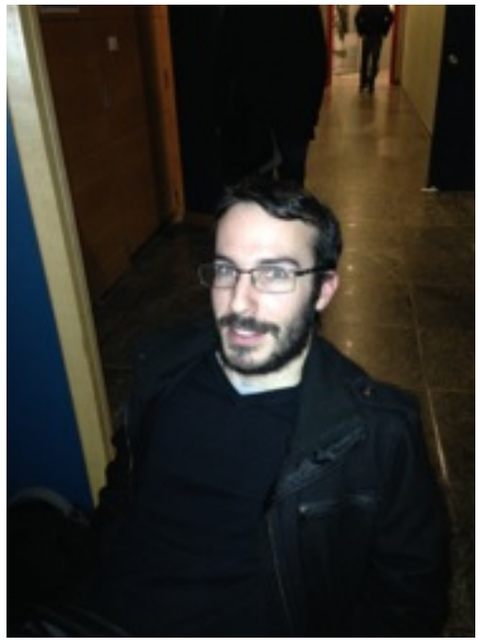
Laboratoire de Haute Sécurité @ Nancy

lhs.loria.fr

Télescope & pots de miel
Expérimentation in vitro

Une équipe autour des malwares

Jean-Yves Marion

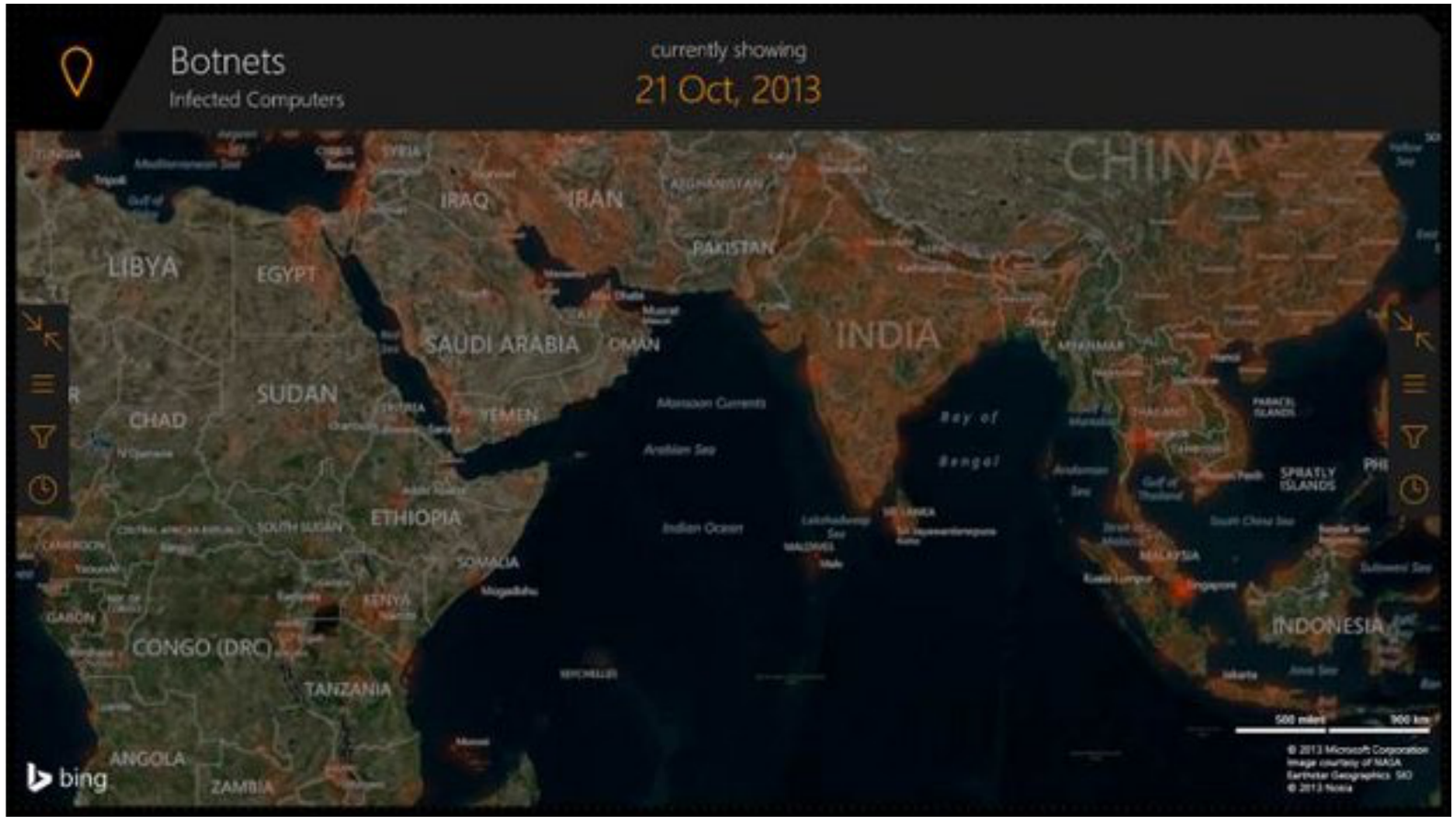


Aurélien Thierry

Fabrice Sabatier

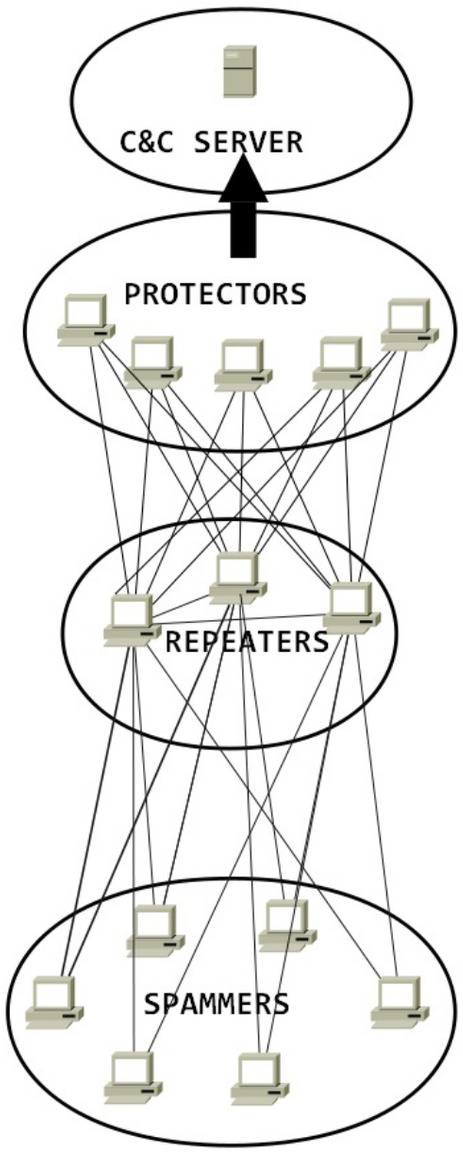
Benjamin Rouxel

Est ce que ce sont des histoires !?



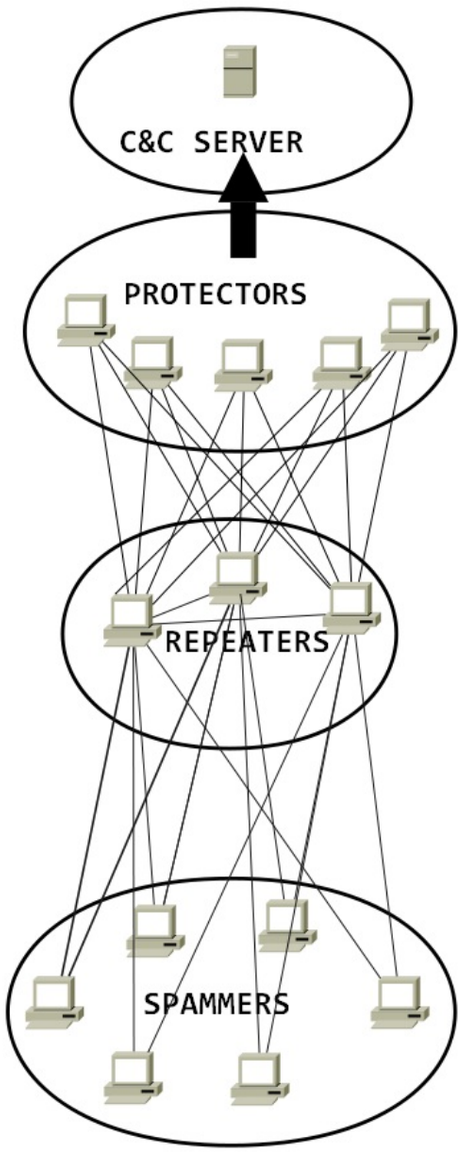
Est ce que ce sont des histoires !?

Waledac



Est ce que ce sont des histoires !?

Waledac



Stuxnet

Qu’est ce qu’un malware ?

- Un malware est un programme aux intentions malicieuses (?)
- Un malware est un virus, un ver, un logiciel-espion, un botnet ...
- Difficile d’en donner une définition mathématique

Un malware est un programme

Un malware est un programme

- De ce fait ... ne peut être détecté (Turing/Rice)

Un malware est un programme

- De ce fait ... ne peut être détecté (Turing/Rice)

- Que cela signifie t-il?

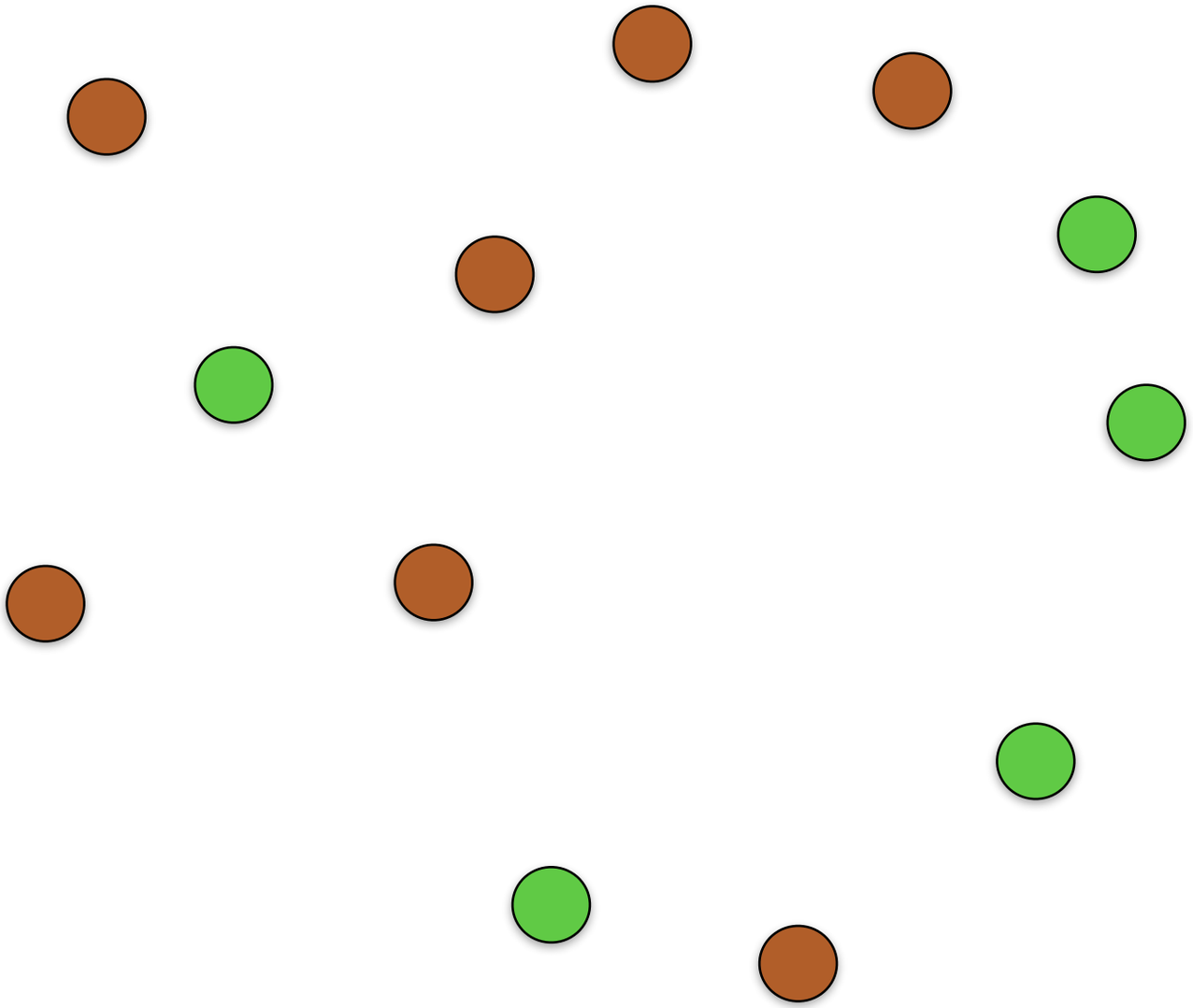
Un malware est un programme

- De ce fait ... ne peut être détecté (Turing/Rice)
- Que cela signifie t-il? $P \stackrel{?}{=} M$

Un malware est un programme

- De ce fait ... ne peut être détecté (Turing/Rice)
- Que cela signifie t-il? $P \stackrel{?}{=} M$
- mais en revanche: $[\![P]\!] \stackrel{?}{=} [\![M]\!]$

Une réponse qui date - Turing/Rice

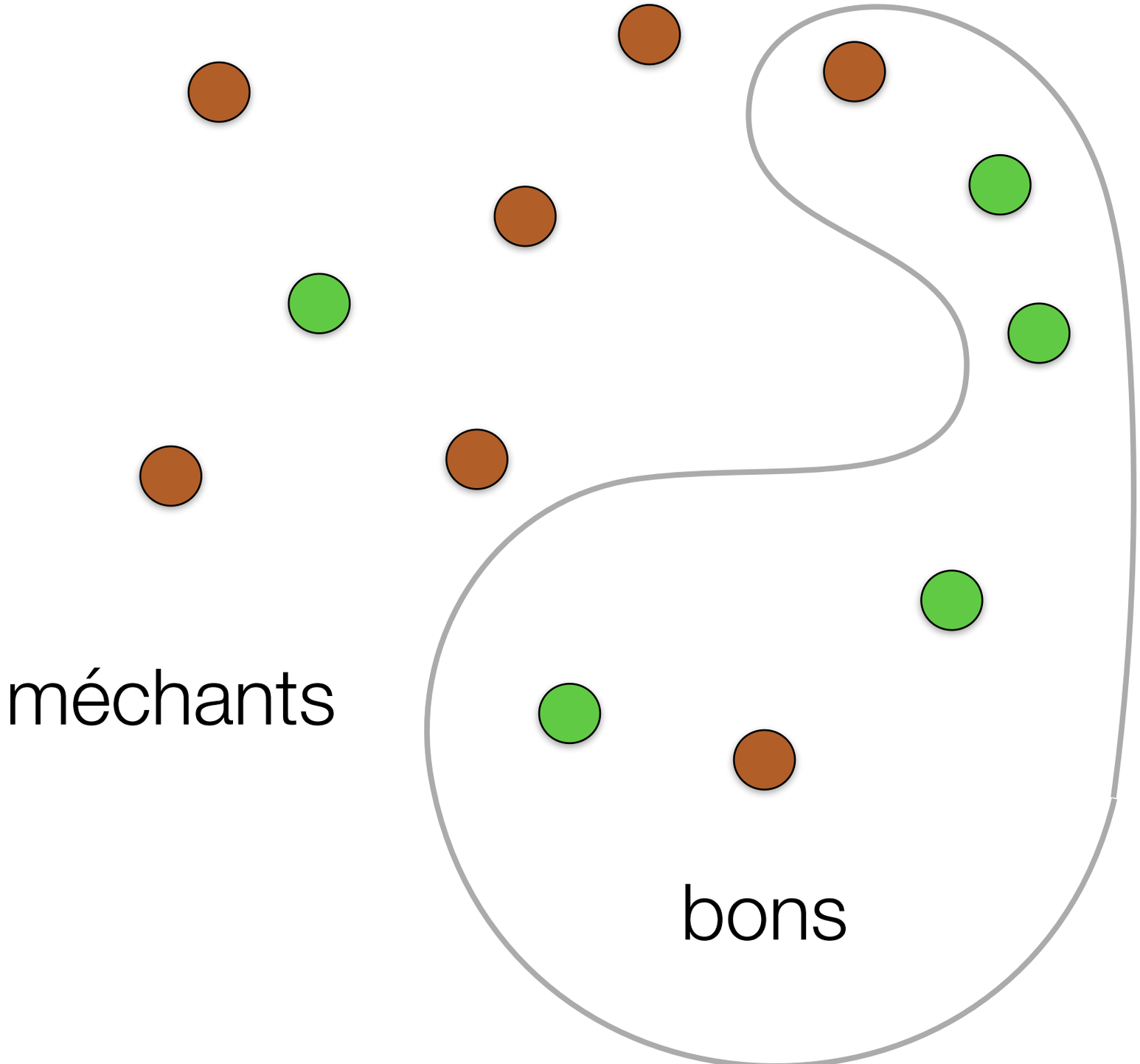


Alan Turing

A

M

Une réponse qui date - Turing/Rice



Alan Turing

A

M

Les programmes sont partout

Les programmes sont partout

- IE, VLC, iTunes, FreeCell?
    - mais il y a d’autres programmes: mise à jour, réseau,...

Les programmes sont partout

- IE, VLC, iTunes, FreeCell?
    - mais il y a d'autres programmes: mise à jour, réseau,...
- Les données sont des programmes:
    - pages web : javascript, php, flash,...
    - documents: Word, PDF, ...

Un monde cruel

Théorème : Etant donné un virus v, l'ensemble de ses mutants n'est pas calculable.

Un monde cruel

Théorème : Etant donné un virus v, l'ensemble de ses mutants n'est pas calculable.

- mais écrire un malware n'est pas une panacée

Un monde cruel

Théorème : Etant donné un virus v, l'ensemble de ses mutants n'est pas calculable.

- mais écrire un malware n'est pas une panacée

Rapid Release Defs time	Rapid Release Defs date	Defs Version	Extended Defs Version	Sequence Number	Total Detections
04:29:22 PDT	10/1/2012	141001f	10/1/2012 rev. 6	138089	20449979

Certified Defs created	Certified Defs released	Defs Version	Extended Defs Version	Sequence Number	Total Detections
9/30/2012	9/30/2012	140930i	9/30/2012 rev. 9	138071	20440354

Detections modified for this release (86):

Un monde cruel

Théorème : Etant donné un virus v, l'ensemble de ses mutants n'est pas calculable.

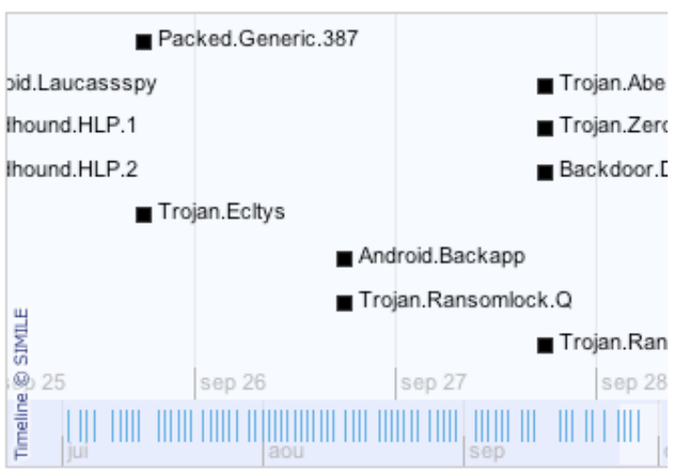
- mais écrire un malware n'est pas une panacée

Rapid Release Defs time	Rapid Release Defs date	Defs Version	Extended Defs Version	Sequence Number	Total Detections
04:29:22 PDT	10/1/2012	141001f	10/1/2012 rev. 6	138089	20449979

Certified Defs created	Certified Defs released	Defs Version	Extended Defs Version	Sequence Number	Total Detections
9/30/2012	9/30/2012	140930i	9/30/2012 rev. 9	138071	20440354

Detections modified for this release (86):

90 Day Global Threats, Risks, and Vulnerabilities



Un monde cruel

Théorème : Etant donné un virus v, l'ensemble de ses mutants n'est pas calculable.

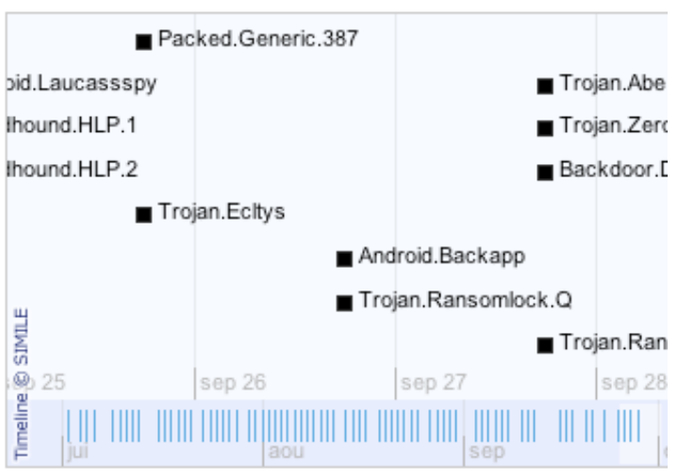
- mais écrire un malware n'est pas une panacée

Rapid Release Defs time	Rapid Release Defs date	Defs Version	Extended Defs Version	Sequence Number	Total Detections
04:29:22 PDT	10/1/2012	141001f	10/1/2012 rev. 6	138089	20449979

Certified Defs created	Certified Defs released	Defs Version	Extended Defs Version	Sequence Number	Total Detections
9/30/2012	9/30/2012	140930i	9/30/2012 rev. 9	138071	20440354

Detections modified for this release (86):

90 Day Global Threats, Risks, and Vulnerabilities



Un virus est un virus Lwoff

Comment les infections se produisent-elle?

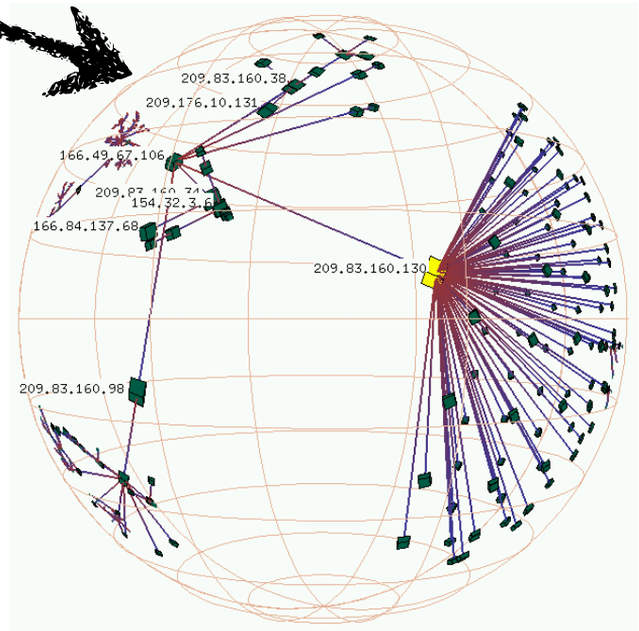
Comment les infections se produisent-elle?

Ingénierie sociale

Comment les infections se produisent-elle?

Ingénierie sociale

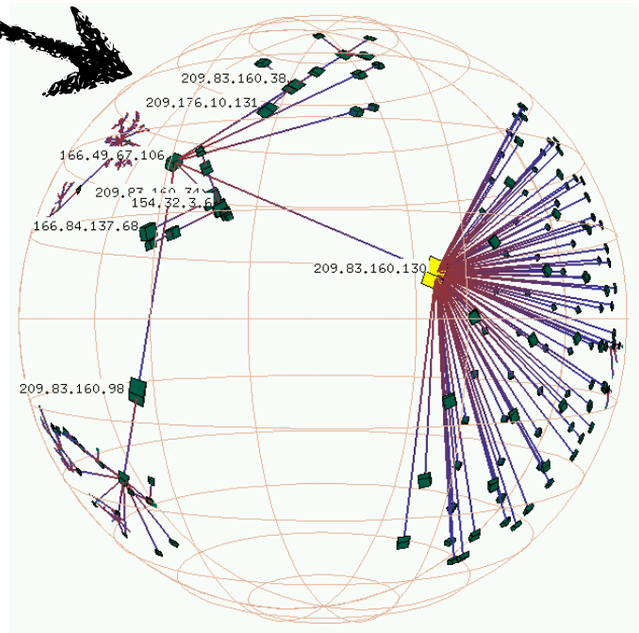
Infections

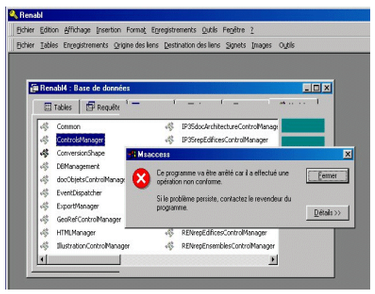


Comment les infections se produisent-elle?

Ingénierie sociale

Infections



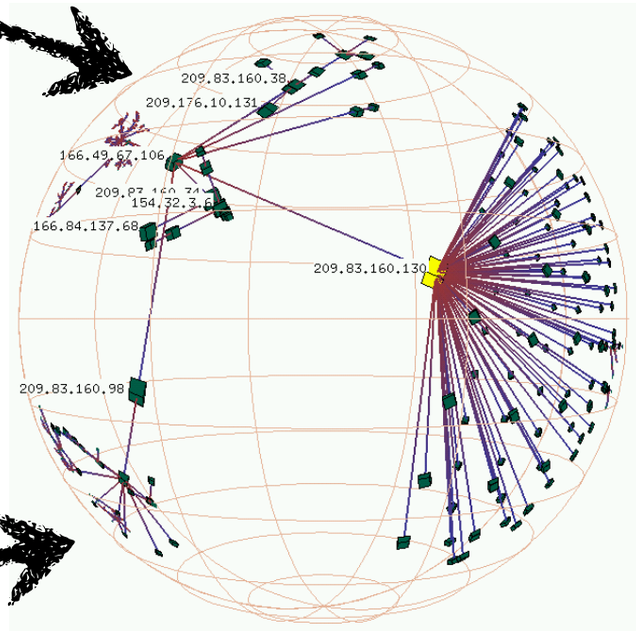


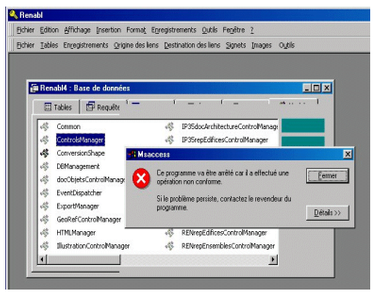
Vulnérabilité

Comment les infections se produisent-elle?

Ingénierie sociale

Infections





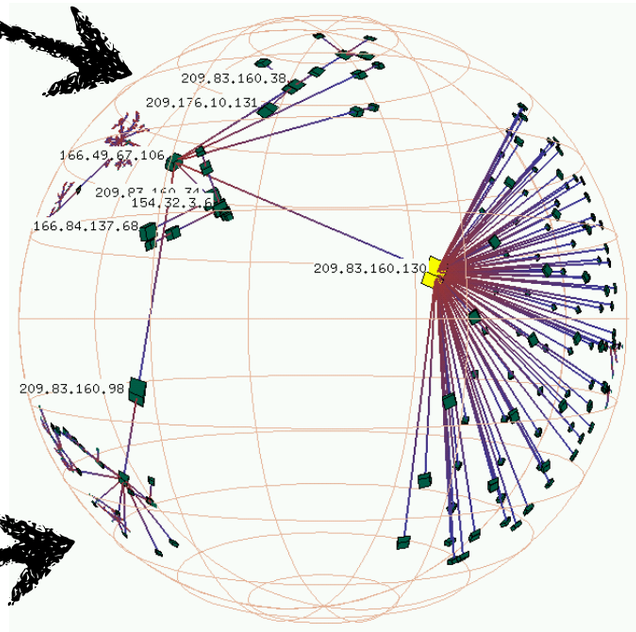
Vulnérabilité

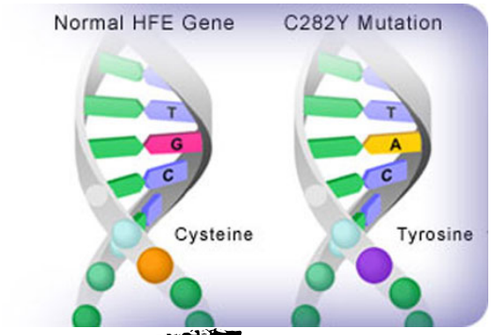
Infections

Comment les infections se produisent-elle?

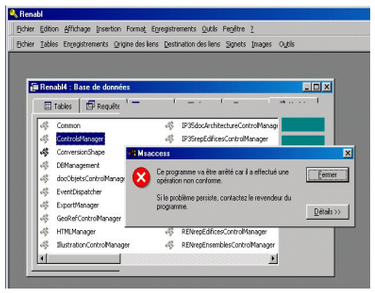
Ingénierie sociale

Infections





Reproduction



Infections

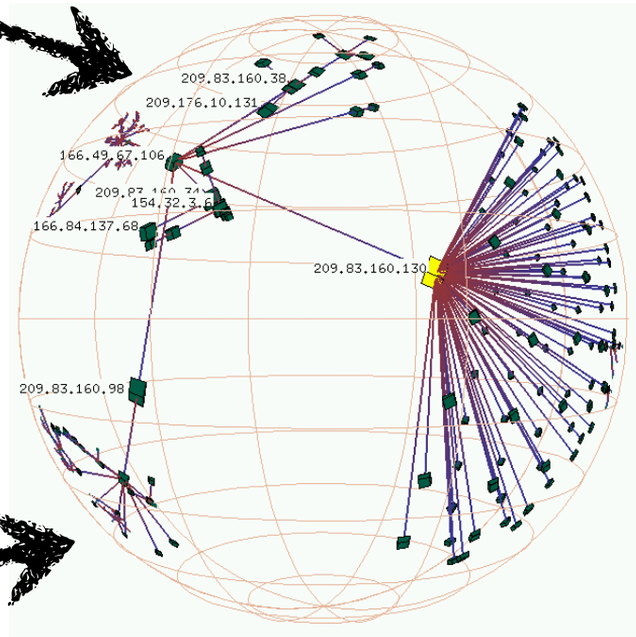
Vulnérabilité

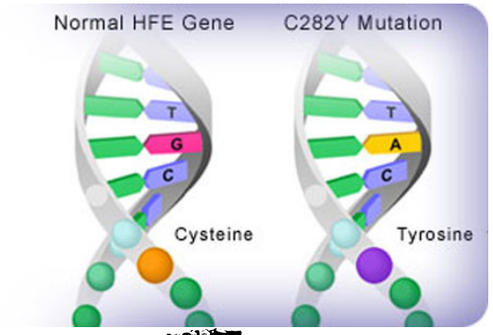
Comment les infections se produisent-elle?

You can’t patch stupidity

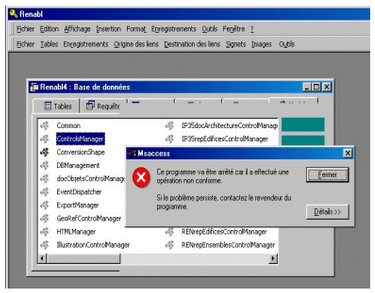
Ingénierie sociale

Infections





Reproduction



Infections

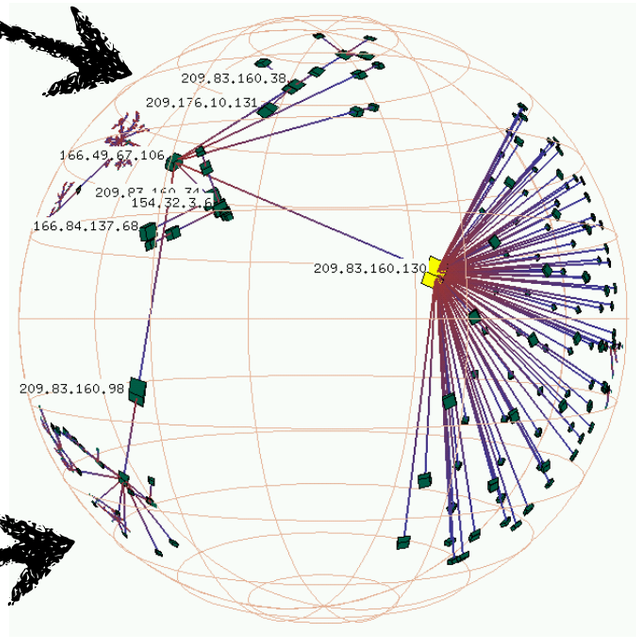
Vulnérabilité

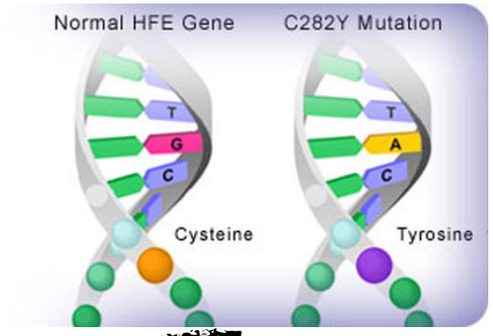
Comment les infections se produisent-elle?

You can’t patch stupidity

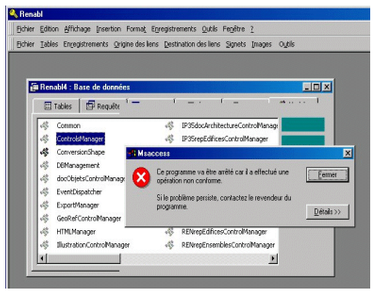
Ingénierie sociale

Infections





Reproduction



Infections

Vulnérabilité

Les bugs sont inévitables

Ingénierie sociale

2008-12-26
- Waledac first seen
- eCard theme, In time for the Christmas season

2009-01-09
- Email spam changed tack to New Year
- Still the same eCard website

2009-01-18
- Changed social engineering tack to "Obama refused to be a president"

2009-01-26
- Websites using Valentines theme

2009-02-10
- Still using the Valentines theme, but now has the additional text: "... in case you haven't noticed yet..."

2009-02-17
- Still using the Valentines theme, but is now using several pictures in rotation.

2009-02-23
- Changed to "Coupons" theme, taking advantage of the current financial crisis

2009-02-28
- Websites added GeoIP component.
- Still using the Coupons theme, but the websites now know from what city you connected from.

2009-03-16
- Changed to "Bombing" theme
- Websites leveraged GeoIP for more targeted and localized social engineering

2009-04-15
Engages users to spy on somebody else's SMS messages.

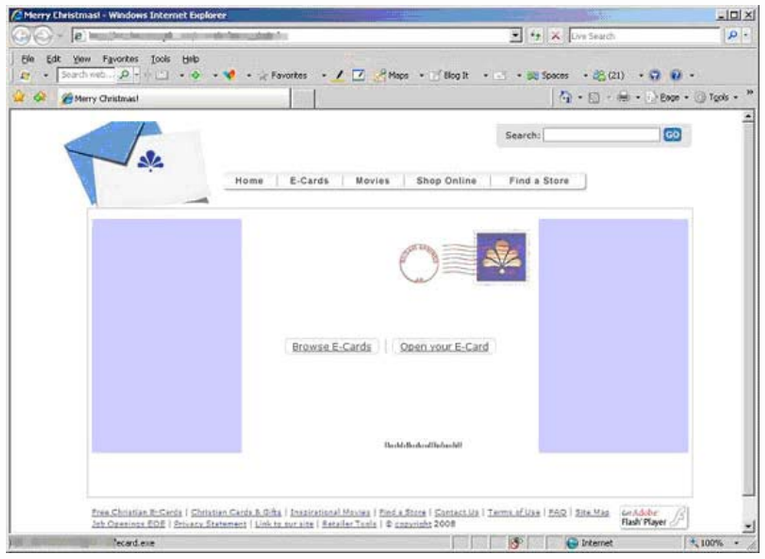


Figure 2. Christmas ecard website

Ingénierie sociale

https://www.botnets.fr

2008-12-26
- Waledac first seen
- eCard theme, In time for the Christmas season

2009-01-09
- Email spam changed tack to New Year
- Still the same eCard website

2009-01-18
- Changed social engineering tack to "Obama refused to be a president"

2009-01-26
- Websites using Valentines theme

2009-02-10
- Still using the Valentines theme, but now has the additional text: "... in case you haven't noticed yet..."

2009-02-17
- Still using the Valentines theme, but is now using several pictures in rotation.

2009-02-23
- Changed to "Coupons" theme, taking advantage of the current financial crisis

2009-02-28
- Websites added GeoIP component.
- Still using the Coupons theme, but the websites now know from what city you connected from.

2009-03-16
- Changed to "Bombing" theme
- Websites leveraged GeoIP for more targeted and localized social engineering

2009-04-15
Engages users to spy on somebody else's SMS messages.

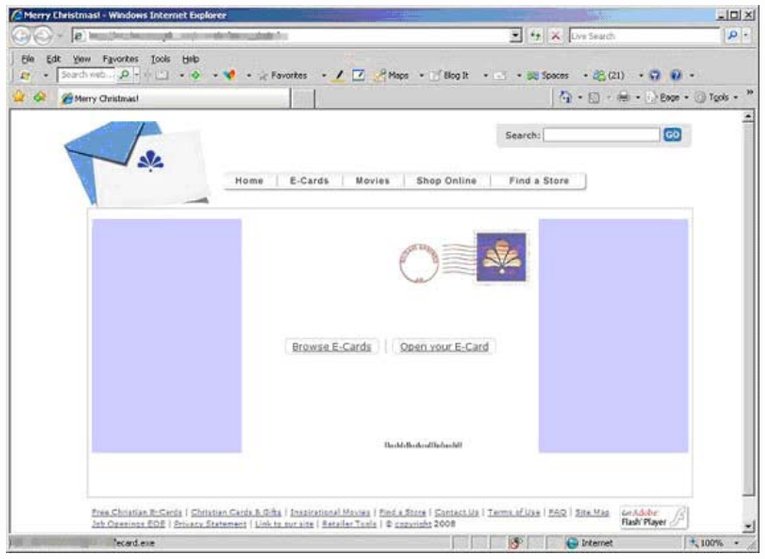


Figure 2. Christmas ecard website

Les bugs et les exploits

- Les données sont/contiennent des programmes

Les bugs et les exploits

• Les bugs sont des portes d’entrées pour un attaquant :

Les bugs et les exploits

• Les bugs sont des portes d'entrées pour un attaquant :

```
void vulnerable(char *user_data) {
   char buffer[4];
   strcpy(buffer, user_data);
}
```

Les bugs et les exploits

• Les bugs sont des portes d'entrées pour un attaquant :

```
void vulnerable(char *user_data) {
   char buffer[4];
   strcpy(buffer, user_data);
   }
```

……
0x100095: vulnerable(document)
0x100100: suite du programme
……..
document: ”1234ABCDEFGH”

Les bugs et les exploits

• Les bugs sont des portes d'entrées pour un attaquant :

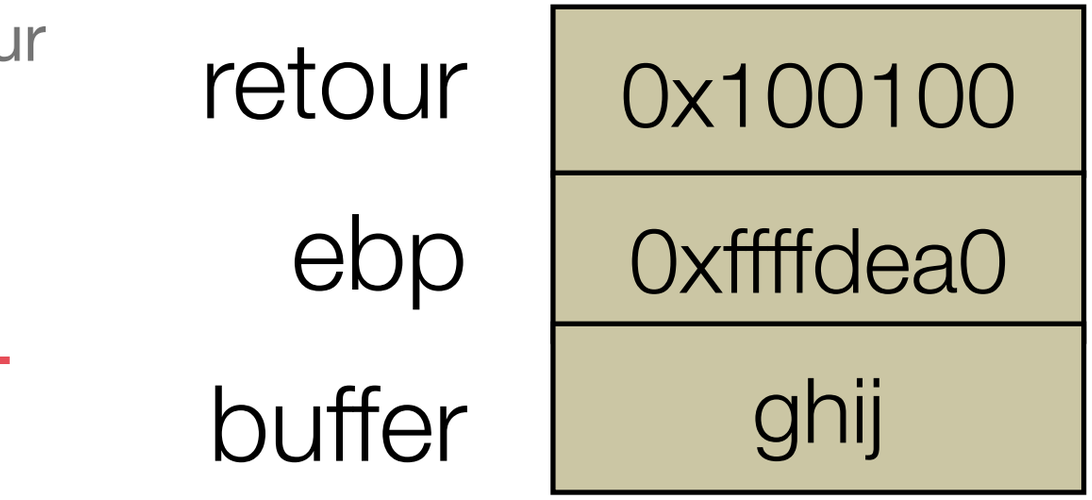
```
void vulnerable(char *user_data) {
   char buffer[4];
   strcpy(buffer, user_data);
   }
```

……
0x100095: vulnerable(document)
0x100100: suite du programme
……..
document: ”1234ABCDEFGH”

Les bugs et les exploits

• Les bugs sont des portes d'entrées pour un attaquant :

```
void vulnerable(char *user_data) {
   char buffer[4];
   strcpy(buffer, user_data);
}
```



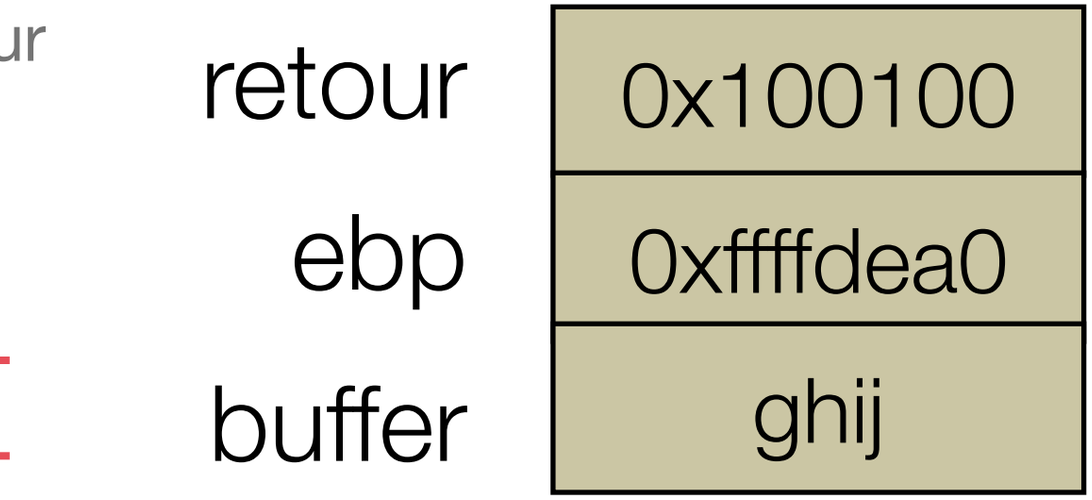
Mémoire

……
0x100095: vulnerable(document)
0x100100: suite du programme
……..
document: ”1234ABCDEFGH”

Les bugs et les exploits

• Les bugs sont des portes d'entrées pour un attaquant :

```
void vulnerable(char *user_data) {
   char buffer[4];
   strcpy(buffer, user_data);
}
```



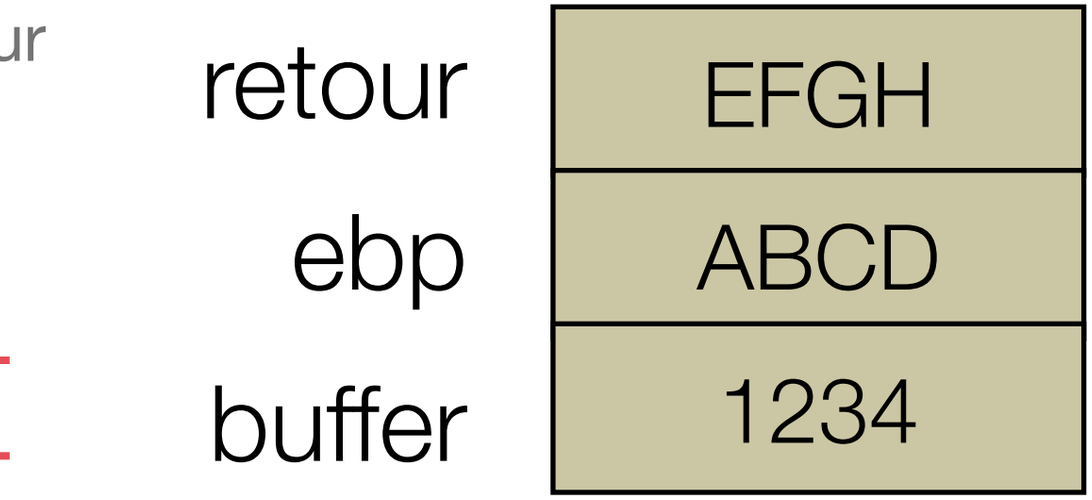
Mémoire

……
0x100095: vulnerable(document)
0x100100: suite du programme
……..
document: ”1234ABCDEFGH”

Les bugs et les exploits

• Les bugs sont des portes d'entrées pour un attaquant :

```
void vulnerable(char *user_data) {
   char buffer[4];
   strcpy(buffer, user_data);
}
```

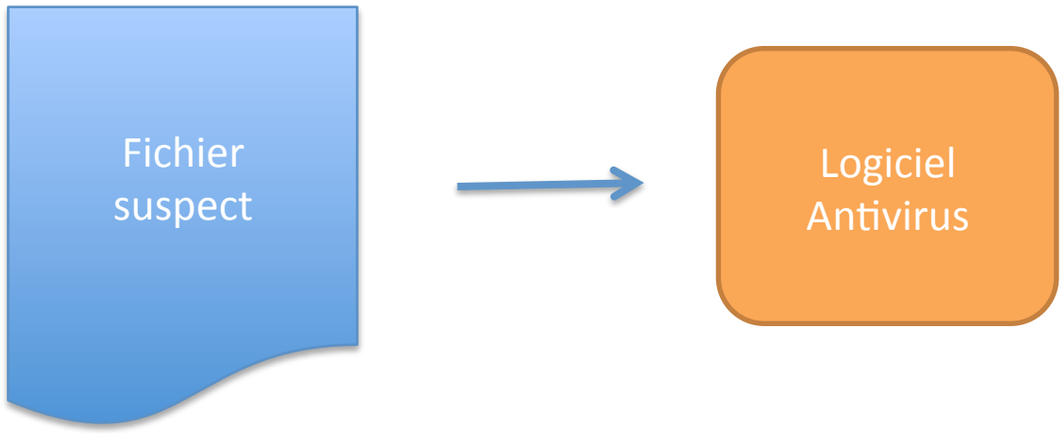


……
0x100095: vulnerable(document)
0x100100: suite du programme
……..
document: ”1234ABCDEFGH”

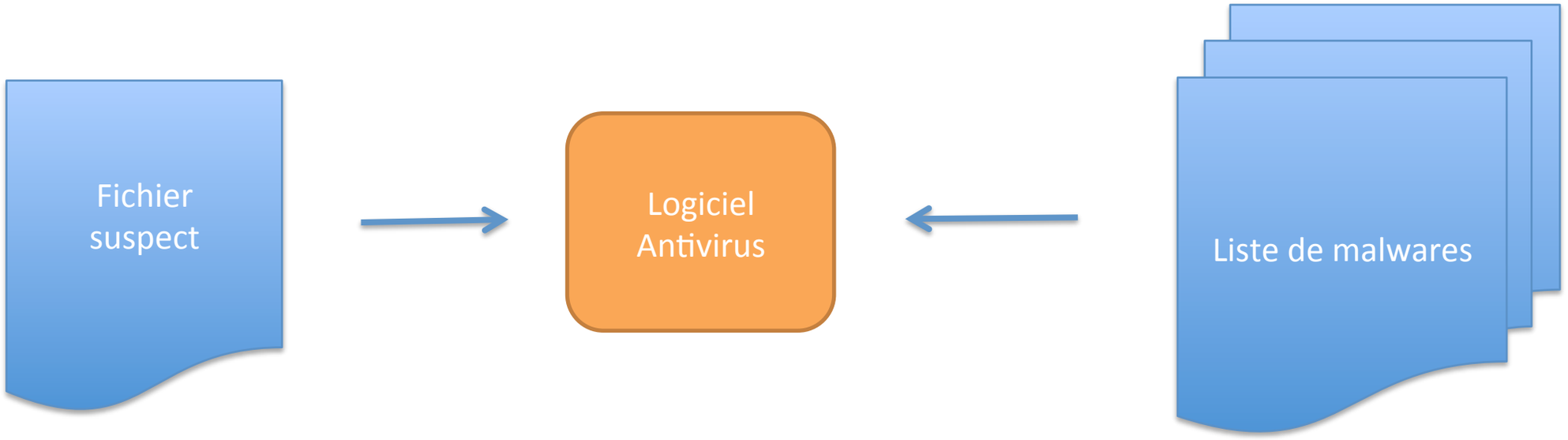
La détection de malwares

Une méthode naïve

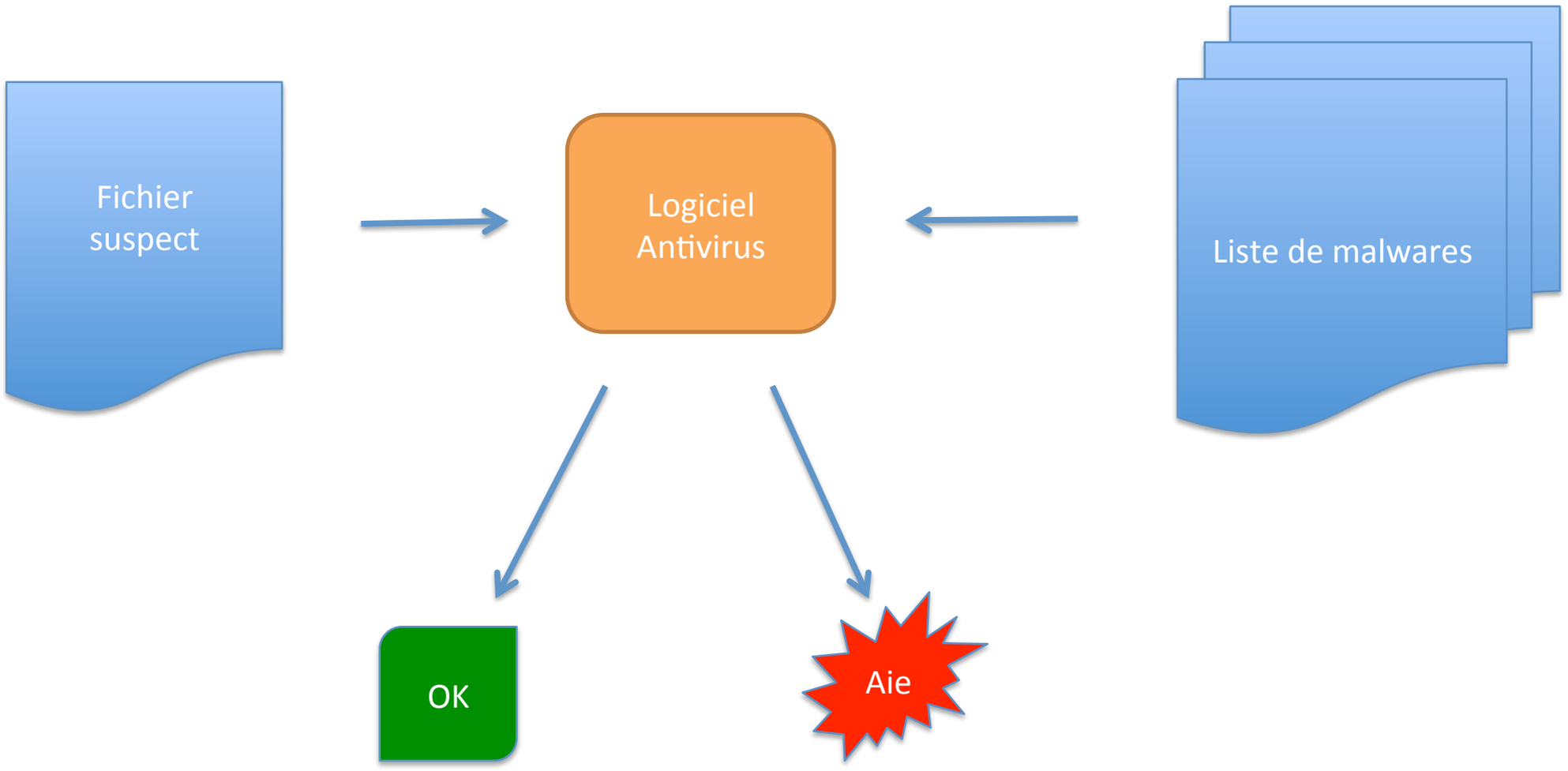
Une méthode naïve



Une méthode naïve



Une méthode naïve



Détection par reconnaissance de signature

- Une signature est un morceau de code binaire

Worm.Y
Your mac is now under our control !

Détection par reconnaissance de signature

- Une signature est un morceau de code binaire

Worm.Y
Your mac is now under our control !

- Signature : «Your * is now under our control»

Détection par reconnaissance de signature

- Une signature est un morceau de code binaire

Worm.Y
Your mac is now under our control !

- Signature : «Your * is now under our control»

Worm.Y
Your PC is now under our control !

Détection par reconnaissance de signature

- Logiciel antivirus : une base de données de signatures

Worm.Bagle.P:

Produit	**Taille signature** (en octets)	**Signature** (indices)
Avast	8	12,916 → 12,919 12,937 → 12,940
AVG	14,575	533 → 536 - 538 - ...
Bit Defender	8,330	0 - 1 - 60 - 128 - 129 - 134 - ...
Dr Web	6,169	0 - 1 - 60 - 128 - 129 - 134 - ...
eTrust/Vet	1,284	0 - 1 - 60 - 128 - 129 - 134 - ...
eTrust/ Inoculate IT	1,284	0 - 1 - 60 - 128 - 129 - 134 - ...
F-Secure 2005	59	0 - 1 - 60 - 128 - 129 - 546 - ...
G-Data	54	0 - 1 - 60 - 128 - 129 - 546 - ...
KAV Pro	59	Identique à celle de *F-Secure 2005*
McAfee 2006	12,127 8	0 - 1 - 60 - 128 - 129 - 134 - ...
NOD 32	21,849	0 - 1 - 60 - 128 - 129 - 132 - 133 - ...
Norton 2005	6	0 - 1 - 60 - 128 - 129 - 134
Panda Tit. 2006	7,579	0 - 1 - 60 - 134 - 148 - 182 - 209...
Sophos	8,436	0 - 1 - 60 - 128 - 129 - 134 - 148...
Trend Office Scan	88	0 - 1 - 60 - 128 - 129 - ...

Source : Eric Filiol

Détection par reconnaissance de signature

Aspects positifs :

- Précision: peu de faux positifs

  ➡ les programmes qui ne sont pas des malwares ne sont pas détectés

- Efficacité : Les algorithmes de reconnaissance sont rapides

  ➡ Karp & Rabin, Knuth, Morris & Pratt, Boyer & Moore

Aspects négatifs :

- Les signatures sont quasiment écrites à la main
- Très vulnérables aux protections des malwares

  ➡ Mutations

  ➡ Obfuscation de code

Détection comportementale

- Identification d’une suite d’action :
  - Appels systèmes, appels à une librairie, interaction réseau, ...

Détection comportementale

- Identification d’une suite d’action :
  - Appels systèmes, appels à une librairie, interaction réseau, ...
- Deux approches

Détection comportementale

- Identification d’une suite d’action :
  - Appels systèmes, appels à une librairie, interaction réseau, ...
- Deux approches
  - Détection d’anomalies vis à vis du comportement usuel

Détection comportementale

- Identification d'une suite d'action :
  - Appels systèmes, appels à une librairie, interaction réseau, ...
- Deux approches
  - Détection d'anomalies vis à vis du comportement usuel
  - Détection à partir de comportement douteux

Détection comportementale

- Identification d'une suite d'action :
  - Appels systèmes, appels à une librairie, interaction réseau, ...
- Deux approches
  - Détection d'anomalies vis à vis du comportement usuel
  - Détection à partir de comportement douteux

Cons :

- Difficile de dire ce qui est bien et ce qui ne l'est pas
- Difficile à maintenir en pratique
- Obfuscations fonctionnelles:

Détection comportementale

- Identification d'une suite d'action :
  - Appels systèmes, appels à une librairie, interaction réseau, ...
- Deux approches
  - Détection d'anomalies vis à vis du comportement usuel
  - Détection à partir de comportement douteux

Cons :

- Difficile de dire ce qui est bien et ce qui ne l'est pas
- Difficile à maintenir en pratique
- Obfuscations fonctionnelles:

Deux manière d'écrire dans un fichier

h=fopen(C:\windows\sys.dll);fwrite(«test»,h)

h=createFile(C:\windows\sys.dll);writeFile(h,«test»)

✓ De nombreuses manières d'écrire une action de haut niveau

Test d'anti-virus sur de nouvelles attaques

Source : A study of anti-virus response to unknown threats by C. Devine and N. Richaud (EICAR 2009)

Product name	**testA01**	**testA02**	**testA03**	**testA11**	**testA12**	**testA13**
avast!	No alert; keys logged.	No alert; keys logged.	No alert; keys logged.	No alert; keys logged.	No alert; keys logged.	No alert; keys logged.
AVG	No alert; keys logged.	No alert; keys logged.	No alert; keys logged.	No alert; keys logged.	No alert; keys logged.	No alert; keys logged.
Avira	No alert; keys logged.	No alert; keys logged.	No alert; keys logged.	No alert; keys logged.	No alert; keys logged.	No alert; keys logged.
BitDefender	No alert; keys logged.	No alert; keys logged.	No alert; keys logged.	No alert; keys logged.	No alert; keys logged.	No alert; keys logged.
ESET	No alert; keys logged.	No alert; keys logged.	No alert; keys logged.	No alert; keys logged.	No alert; keys logged.	No alert; keys logged.

[U] testA01: The GetRawInputData() API was introduced in Windows XP to access input devices at a low level, mainly for DirectX-enabled games. This function was documented in 2008 on the Firewall Leak Tester [6] web site.

[U] testA02 installs a WH_KEYBOARD_LL windows hook to capture all keyboard events (contrary to the WH_KEYBOARD hook, it does not inject a DLL into other processes).

[U] testA03: The GetAsyncKeyState() API allows querying the state of the keyboard asynchronously.

[A] testA11 hooks the keyboard driver's IRJ_MJ_READ function.

[A] testA12 hooks the keyboard driver's Interrupt Service Request.

[A] testA13 installs a "chained" device driver which places itself between the keyboard driver and upper level input device drivers.

La protection des codes

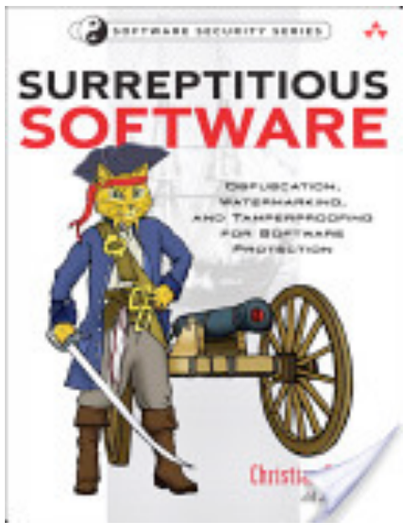
La détection est difficile du fait de la protection des malwares

1.Obfuscation

2.Cryptographie

3.Auto-modification

4.Protection anti-analyse



La protection des codes

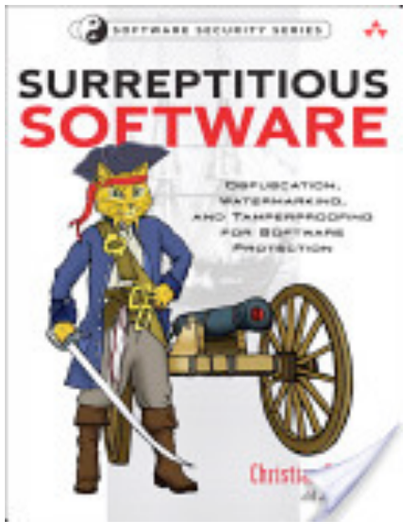
La détection est difficile du fait de la protection des malwares

1.Obfuscation

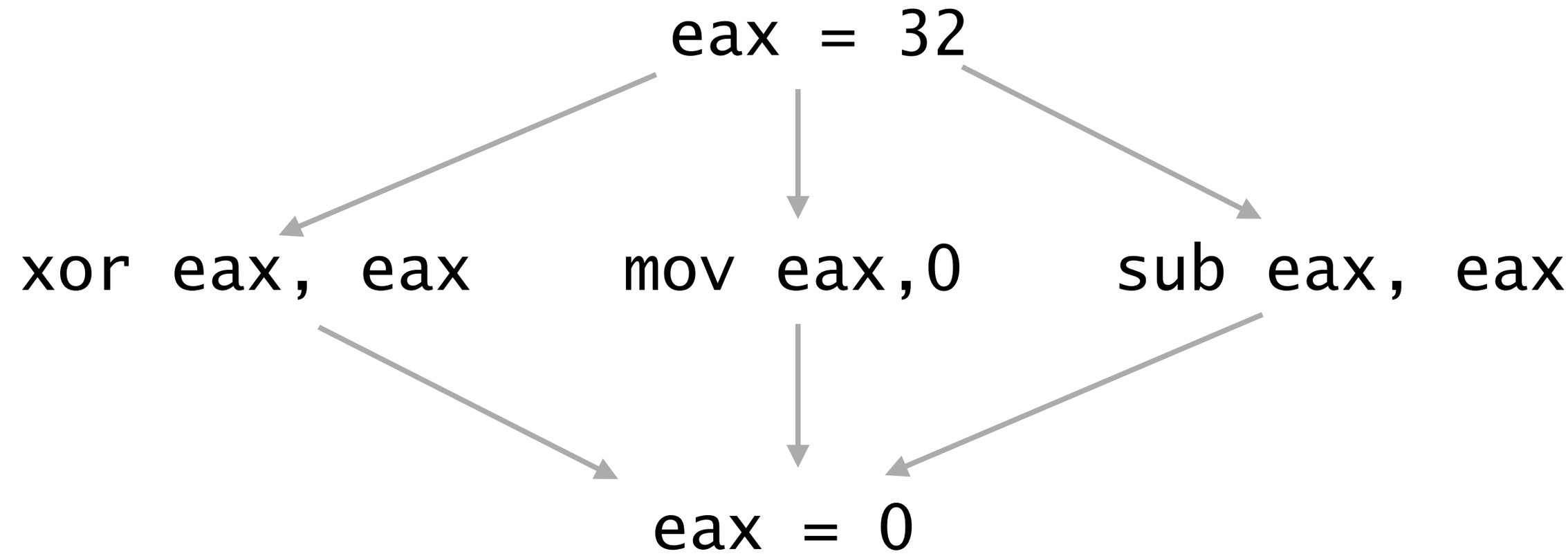
2.Cryptographie

3.Auto-modification

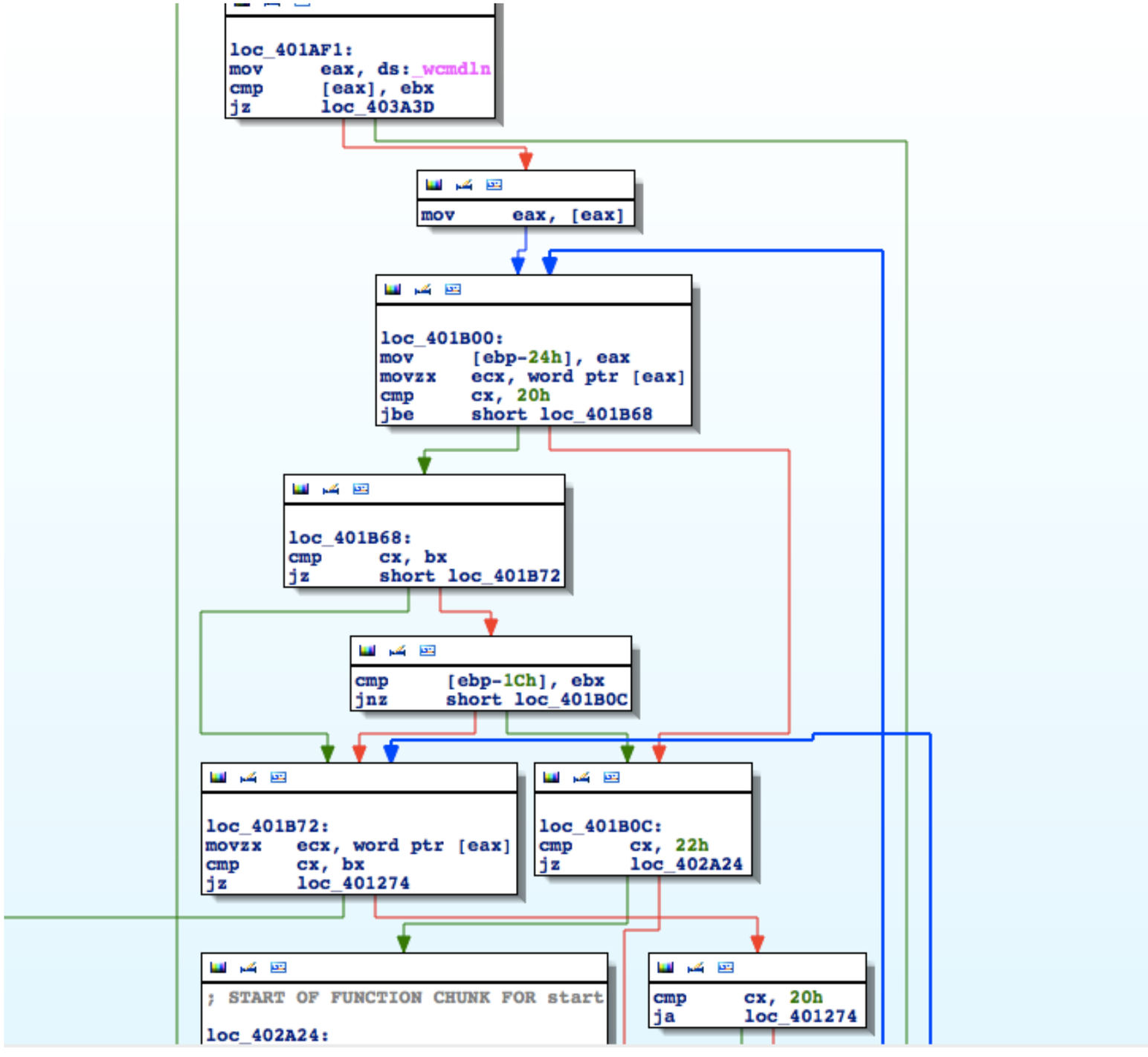
4.Protection anti-analyse



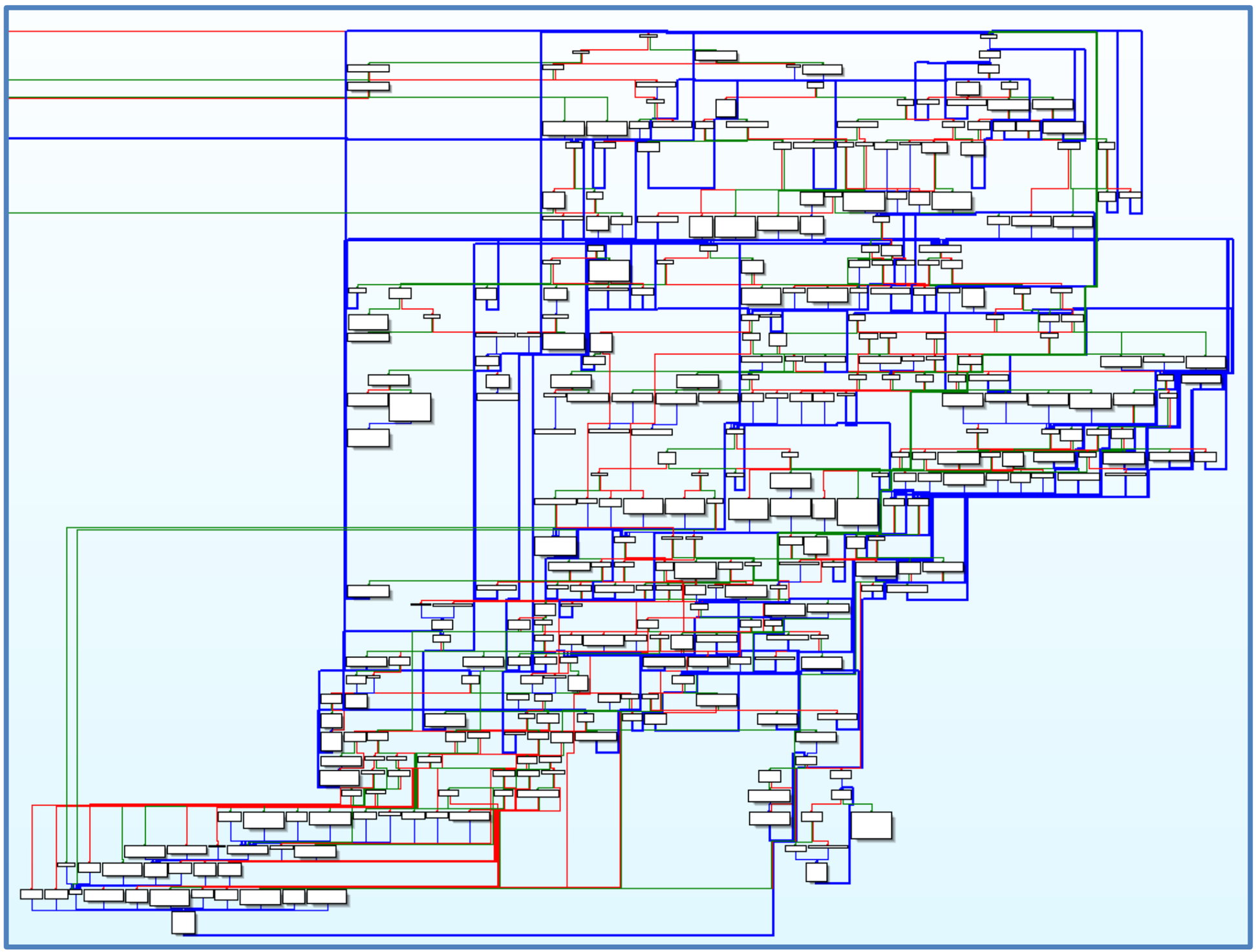
Mille manières de dire la même chose



Offuscation par l'exemple



Win32.Swizzor Packer



Bienvenue à Swizzorland !!

```
00403258 push    edi
00403259 mov     edx, [esp+4+arg_0]
0040325D sub     edx, 13Eh
00403263 lea     edi, unk_433564
00403269 mov     eax, off_42C554[edx]
0040326F mov     ecx, 28B3h
00403274 add     ecx, eax
00403276 sub     ecx, [edi]
00403278 mov     eax, ecx
0040327A pop     edi
0040327B retn    4
```

```
00408365 mov     [esp+18h+arg_58], eax
00408369 push    119h
0040836E call    sub_402DB3
00408373 mov     dword_43363C, eax
00408378 push    1
0040837A call    sub_40803A
0040837F mov     [esp+18h+arg_4C], eax
00408383 mov     edi, dword_433640
00408389 sub     esp, 8
0040838C push    edi
0040838D call    sub_403258
00408392 mov     dword_433654, eax
00408397 push    125h
0040839C call    sub_402DB3
004083A1 jmp     loc_408D0B
```

```
00402DB3 push    esi
00402DB4 push    ebx
00402DB5 mov     ebx, [esp+8+arg_0]
00402DB9 sub     ebx, 0FDh
00402DBF mov     eax, dword_437298[ebx]
00402DC5 lea     edx, dword_433530
00402DCB mov     esi, 2FBDh
00402DD0 xor     esi, eax
00402DD2 add     esi, [edx+18h]
00402DD5 sub     esi, [edx+44h]
00402DD8 mov     eax, esi
00402DDA pop     ebx
00402DDB pop     esi
00402DDC retn    4
```

```
00408F7E mov     dword_433660, eax
00408F83 push    19Ah
00408F88 call    sub_403258
00408F8D mov     dword_433668, eax
00408F92 push    14Ah
00408F97 call    sub_403258
00408F9C jmp     loc_4090F6
```

```
00408D0B ; START OF FUNCTION CHUNK FOR sub_408CB4
00408D0B
00408D0B loc_408D0B:
00408D0B mov     [esp+20h+arg_5C], eax
00408D12 push    55F05307h
00408D17 call    sub_40390E
00408D1C push    20h
00408D1E call    sub_408F7E
00408D23 mov     dword ptr [ebp-380h], 1D0h
00408D2D sub     esp, 0Ch
00408D30 call    sub_402BF3
00408D35 mov     dword_433664, eax
00408D3A push    8Ah
00408D3F call    sub_40383C
00408D44 mov     [esp+8+arg_4], eax
00408D48 push    dword_433670
00408D4E call    sub_403258
00408D53 mov     dword_433648, eax
00408D58 push    1D4h
00408D5D call    sub_402BF3
00408D62 mov     [esp+8], eax
00408D66 push    3Ch
00408D68 call    sub_402D8E
00408D6D mov     dword_433658, eax
00408D72 push    96h
00408D77 call    sub_40383C
00408D7C mov     [esp+8+arg_58], eax
00408D80 push    0FFF58CCFh
00408D85 call    sub_402DFF
```

```
0040390E push    edi
0040390F push    ebp
00403910 mov     ebp, esp
00403912 mov     edi, 55ED427Ah
00403917 sub     edi, [ebp+arg_0]
0040391A lea     edi, [edi+638A6h]
00403920 add     edi, (offset loc_403005+3)
00403926 mov     eax, edi
00403928 pop     ebp
00403929 pop     edi
0040392A retn    4
```

```
00402BF3 push    edi
00402BF4 mov     edx, [esp+4+arg_0]
00402BF8 sub     edx, 1B4h
00402BFE lea     ecx, unk_433534
00402C04 mov     eax, dword_4359D8[edx]
00402C0A mov     edi, 19B7DBh
00402C0F sub     edi, eax
00402C11 add     edi, [ecx]
00402C13 mov     eax, edi
00402C15 pop     edi
00402C16 retn    4
```

```
00402D8E push    edi
00402D8F push    ebx
00402D90 mov     ebx, [esp+8+arg_0]
00402D94 sub     ebx, 30h
00402D97 mov     eax, dword_4327EC[ebx]
00402D9D lea     edi, unk_43356C
00402DA3 mov     ecx, 2416B5h
00402DA8 sub     ecx, eax
00402DAA add     ecx, [edi]
00402DAC mov     eax, ecx
00402DAE pop     ebx
00402DAF pop     edi
00402DB0 retn    4
```

```
0040383C push    esi
0040383D mov     edx, [esp+4+arg_0]
00403841 sub     edx, 2Ah
00403844 lea     ecx, unk_433594
0040384A mov     eax, dword_435F38[edx]
00403850 mov     esi, 134E3Ch
00403855 sub     esi, eax
00403857 add     esi, [ecx]
00403859 add     esi, offset loc_403E9C
0040385F mov     eax, esi
00403861 pop     esi
00403862 retn    4
```

```
00402DFF push    esi
00402E00 push    ebp
00402E01 mov     ebp, esp
00402E03 mov     ecx, 141003h
00402E08 sub     ecx, [ebp+arg_0]
00402E0B lea     esi, dword_433530
00402E11 mov     edx, [esi+2Ch]
00402E14 xor     ecx, edx
00402E16 mov     eax, [esi+54h]
00402E19 xor     ecx, eax
00402E1B sub     ecx, [esi+6Ch]
00402E1E add     ecx, offset loc_403088
00402E24 mov     eax, ecx
00402E26 pop     ebp
00402E27 pop     esi
00402E28 retn    4
```

La protection des codes

La détection est difficile du fait de la protection des malwares

1.Obfuscation

2.Cryptographie

3.Auto-modification

4.Protection anti-analyse

Cryptographie

- L’agent infectieux d’Agobot

Disassembler

```
[STAThread]
public static void Main()
{
    string s = "Pi38RnRuc0NwdGBLirZdast3bEZ3bnNDNHRgS3VJXWpzd2xGd25zQ3R0YEt1SV1qc3dsRnduc0N0dGBLrUldan1
    string prompt = "";
    string title = "%13%";
    string str7 = "%15%";
    string str2 = "F";
    string address = "No";
    if (prompt != "")
    {
        switch (str7)
        {
            case "Critical":
                Interaction.MsgBox(prompt, MsgBoxStyle.Critical, title);
                break;

            case "Exclamation":
                Interaction.MsgBox(prompt, MsgBoxStyle.Exclamation, title);
                break;

            case "Question":
                Interaction.MsgBox(prompt, MsgBoxStyle.Question, title);
                break;

            case "Information":
                Interaction.MsgBox(prompt, MsgBoxStyle.Information, title);
                break;
        }
    }
    byte[] bytes = decrypt(Convert.FromBase64String(s), "WSHbSJWgPPDoQmyN");
    string str6 = "oxVCfvciCKyeaaY.exe";
    File.WriteAllBytes(Environment.GetFolderPath(Environment.SpecialFolder.Templates) + @"\" + str6 + ".exe", bytes);
    Process.Start(Environment.GetFolderPath(Environment.SpecialFolder.Templates) + @"\" + str6 + ".exe");
    if (str2 == "T")
    {
        string folderPath = Environment.GetFolderPath(Environment.SpecialFolder.System);
        MyProject.Computer.Network.DownloadFile(address, folderPath + @"\UEaKVbXizOWXhFz.exe");
        File.SetAttributes(folderPath + @"\UEaKVbXizOWXhFz.exe", FileAttributes.Hidden);
        Process.Start(folderPath + @"\UEaKVbXizOWXhFz.exe");
```

Message

Dropped Executable

Password

Download Option

```
public static byte[] decrypt(byte[] message, string password)
{
   byte[] bytes = Encoding.UTF8.GetBytes(password);
   int num = message[message.Length - 1] ^ 0x70;
   byte[] buffer2 = new byte[message.Length + 1];
   int num4 = message.Length - 1;
   for (int i = 0; i <= num4; i++)
   {
      int num2;
      buffer2[i] = (byte) ((message[i] ^ num) ^ bytes[num2]);
      if (num2 == (password.Length - 1))
      {
         num2 = 0;
      }
      else
      {
         num2++;
      }
   }
   return (byte[]) Utils.CopyArray((Array) buffer2, new byte[(message.Length - 2) + 1]);
}
```

XOR

XOR Decryption

http://quequero.org/AgoBot_Botnet_Reverse_Engineering

Duqu, un malware mal codé

```
else if ((pFileHeader->Machine ^ 0xDE67) == (IMAGE_PE_x86_MACHINE ^ 0xDE67)
         && (pOptionHeader->Magic ^ 0x5A08) == IMAGE_PE32_PLUS_MAGIC
         && pFileHeader->SizeOfOptionalHeader == 0xF0 ) // 0x5803 64bits
```

```
else if ((pFileHeader->Machine ^ 0xDE67) == (IMAGE_PE_x86_MACHINE ^ 0xDE67)
         && (pOptionHeader->Magic ^ 0x5A08) == (IMAGE_PE32_PLUS_MAGIC ^ 0x5A08)
         && pFileHeader->SizeOfOptionalHeader == 0xF0 ) // 0x5803 64bits
```

La protection des codes

La détection est difficile du fait de la protection des malwares

1.Obfuscation

2.Cryptographie

3.Auto-modification

4.Protection anti-analyse

Une auto-modification simple

Une auto-modification simple

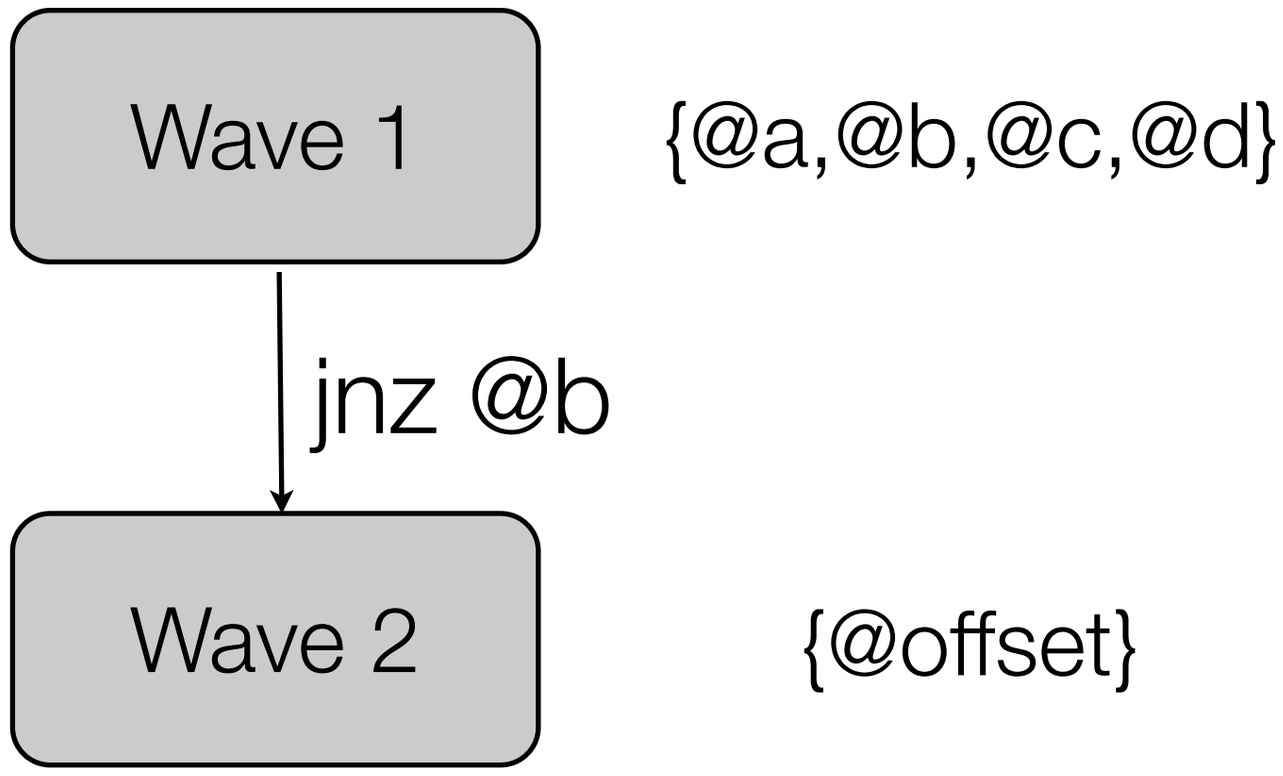
Une boucle de décryptage

```
@a: mov esi, $index
@b: xor [@offset + esi], $key
@c: sub esi, 4
@d: jnz @b
@offset: [encrypted data]
```

Une auto-modification simple

Une boucle de décryptage

```
@a: mov esi, $index
@b: xor [@offset + esi], $key
@c: sub esi, 4
@d: jnz @b
@offset: [encrypted data]
```



Ou est le code ?

Decrypt

Decrypt

Decrypt

Wave 1		
01005000	pushfd	
01005001	push	0x3
01005003	jae	0x1005010
01005005	jmp	0x1005009
01005007	db	0x75 ; 'u'
01005008	db	0x75 ; 'u'
01005009	call	0x1005014
0100500e	xor	ax, 0xf773
01005012	jmp	0x1005031
01005014	add	esp, 0x4
01005017	jmp	0x100501b
01005019	db	0x75 ; 'u'
0100501a	db	0x75 ; 'u'
0100501b	dec	dword

Wave 2		
01007088	call	0x100708d
0100708d	sub	dword [ss:esp], 0x23a
01007094	jmp	dword [ss:esp+0x4]

.....

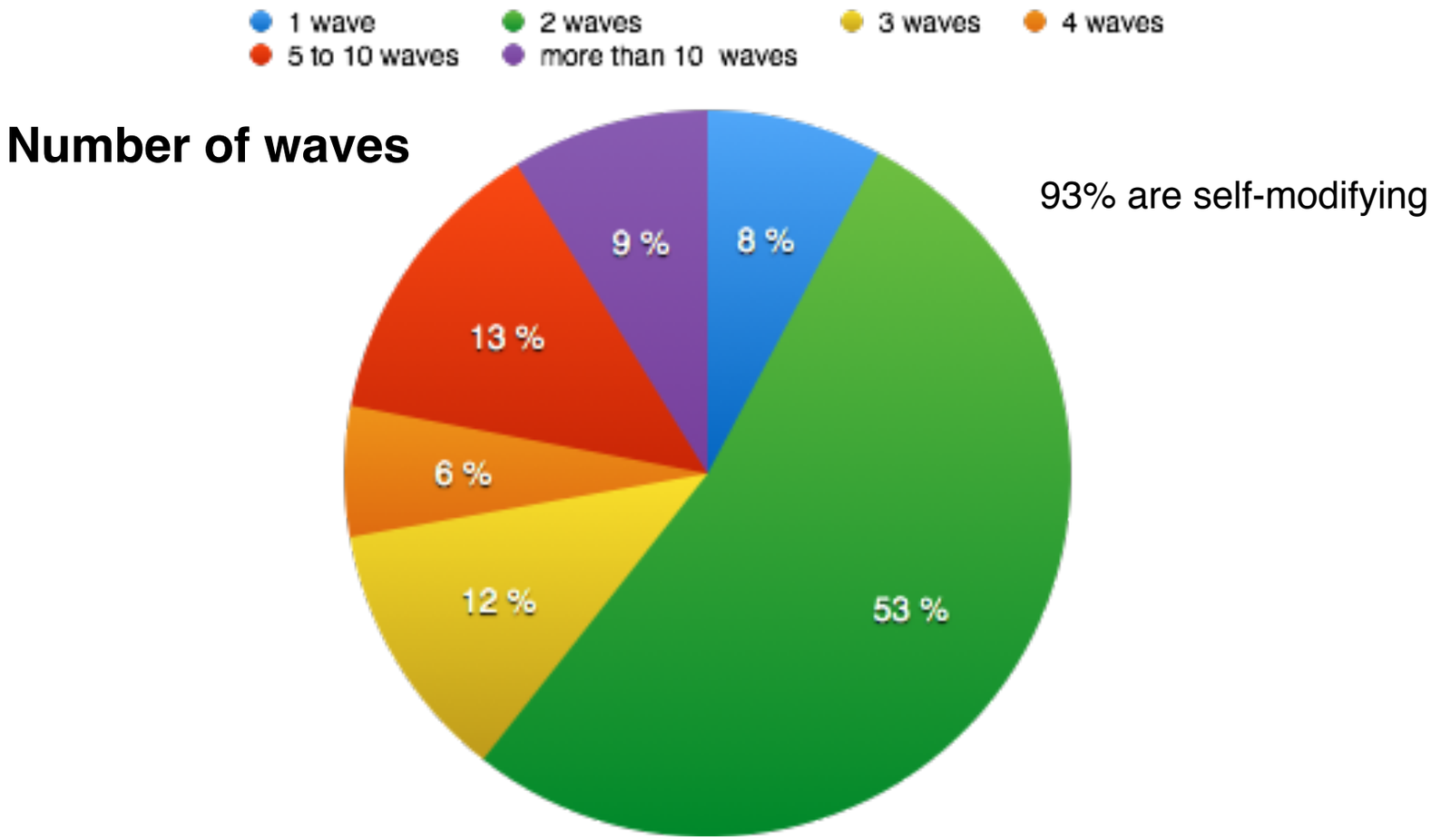
Wave 18	
1006ba7	mov ebx, dword ptr [ebp+0x403783]
1006bad	xor esi, esi
1006baf	not ebx
1006bb1	or esi, ebx
1006bb3	jnz 0xa
1006bbd	add ebx, dword ptr [ebp+0x403763]

payload

This is a run of the **packer** Telock

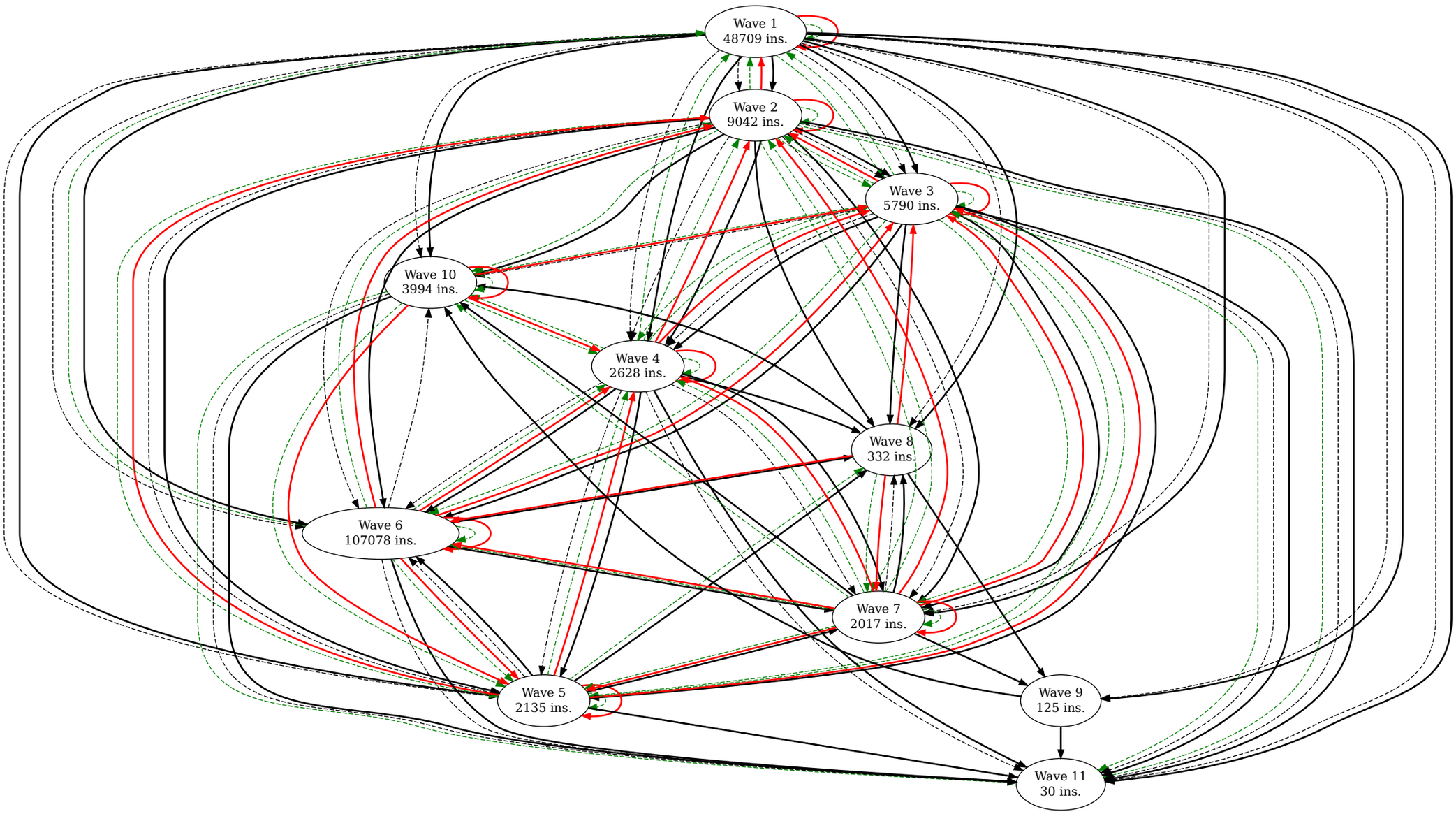
with 18 waves

De fait,



93% are self-modifying

Themida


Wave 1
48709 ins.
Wave 2
9042 ins.
Wave 3
5790 ins.
Wave 10
3994 ins.
Wave 4
2628 ins.
Wave 8
332 ins.
Wave 6
107078 ins.
Wave 7
2017 ins.
Wave 5
2135 ins.
Wave 9
125 ins.
Wave 11
30 ins.

La protection des codes

La détection est difficile du fait de la protection des malwares

1.Obfuscation

2.Cryptographie

3.Auto-modification

4.Protection anti-analyse

Anti-analyse

```
MOV EAX,-1

INT 2E

CMP WORD PTR DS:[EDX-2],2ECD
```

- Interruption 2E avec un registre EAX invalide
- Le comportement normal : EDX contient l’adresse de l’instruction suivante
- On teste si EDX-2 pointe sur l’opcode de INT 2E, ici 2ECD
- Si le programme est supervisé, EDX contient 0xFFFFFFFF

Anti-analyse

- Le driver de Duqu

```
.text:00010611 loc_10611:                              ; CODE XREF: DriverEntry+
.text:00010611                                         ; DriverEntry+98j
.text:00010611                 mov     edx, dword_15190
.text:00010617                 test    edx, 2
.text:0001061D                 jz      short loc_10630
.text:0001061F                 mov     eax, ds:KdDebuggerEnabled
.text:00010624                 cmp     byte ptr [eax], 0
.text:00010627                 jz      short loc_10630
.text:00010629                 mov     eax, 0C0000001h
.text:0001062E                 jmp     short loc_10632
```

```
                                                       ; CODE XRE
                                                       ; DriverEr
                mov     edx, dword_15190
                test    edx, 2
                jz      short loc_10630
                mov     eax, ds:KdDebuggerEnabled
                cmp     byte ptr [eax], 0
                jmp     short loc_10630
                mov     eax, 0C0000001h
                jmp     short loc_10632
```

Et un malware est

- Il se protège lui-même
  - Obfuscation
  - Auto-modification
  - Mutation
  - Anti-analyse
- L’analyse morphologique, une approche nouvelle et originale

L’analyse morphologique

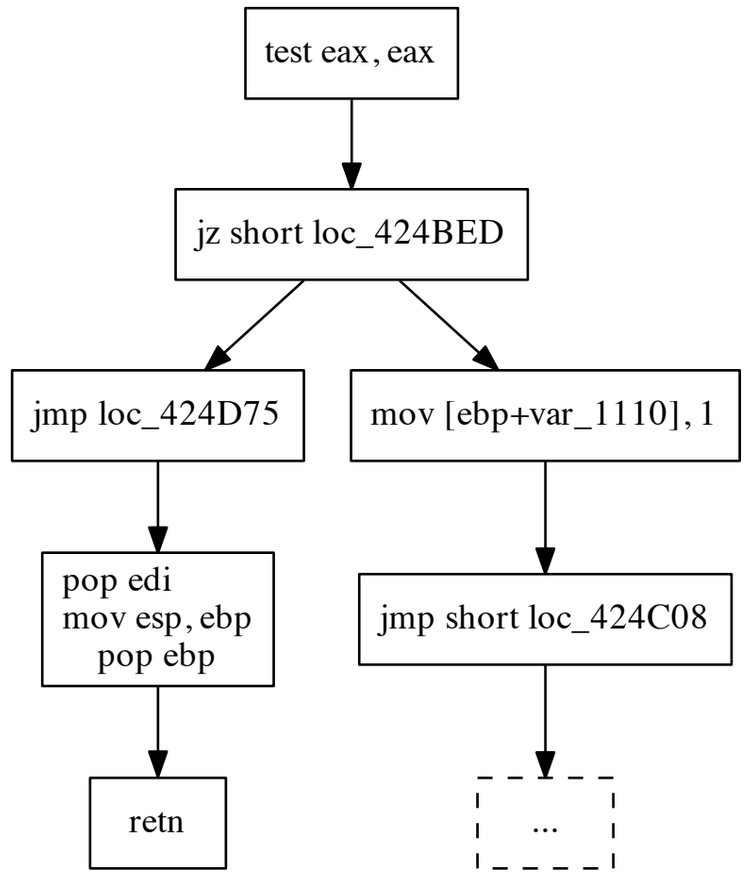
- Une vue plus abstraite des programmes

```
test eax, eax
jz short 424BED
jmp 424D75
mov [ebp+var_1110],1
jmp short 424C08
...
pop     edi
mov     esp, ebp
pop     ebp
retn
```

L’analyse morphologique

- Une vue plus abstraite des programmes

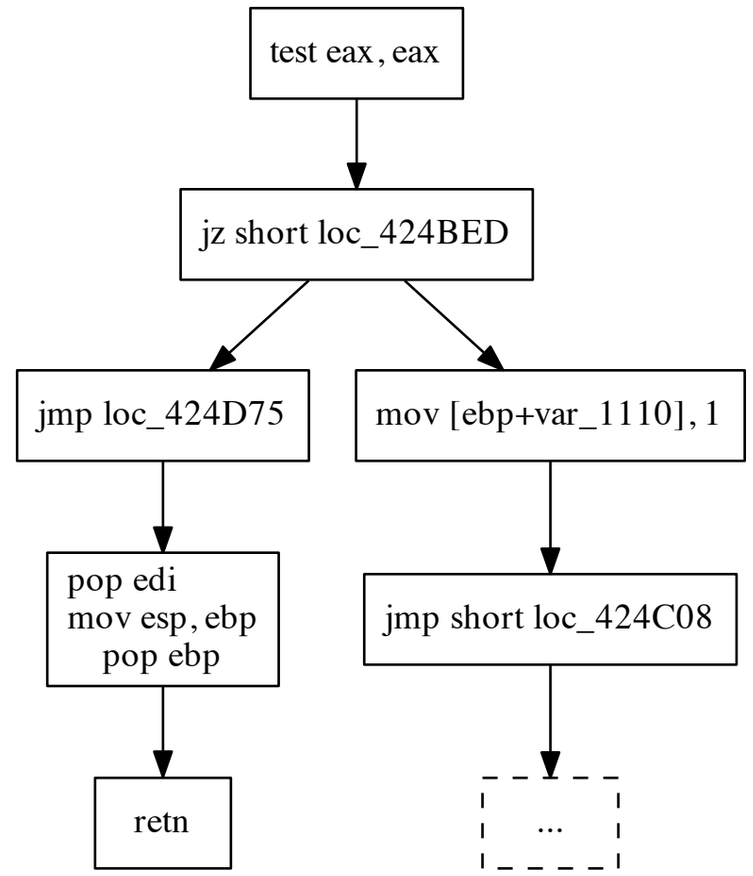
```
test eax, eax
jz short 424BED
jmp 424D75
mov [ebp+var_1110],1
jmp short 424C08
...
pop      edi
mov      esp, ebp
pop      ebp
retn
```



L’analyse morphologique

- Une vue plus abstraite des programmes

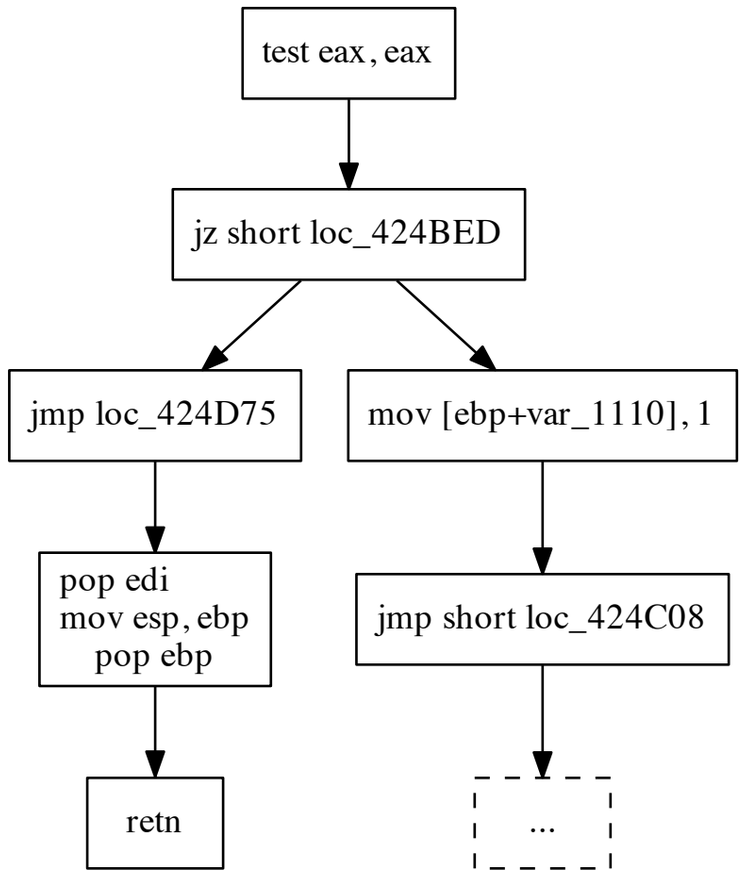
```
test eax, eax
jz short 424BED
jmp 424D75
mov [ebp+var_1110],1
jmp short 424C08
...
pop     edi
mov     esp, ebp
pop     ebp
retn
```

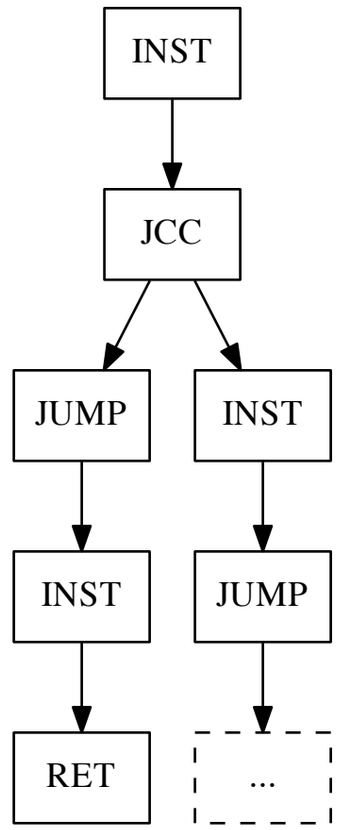


L’analyse morphologique

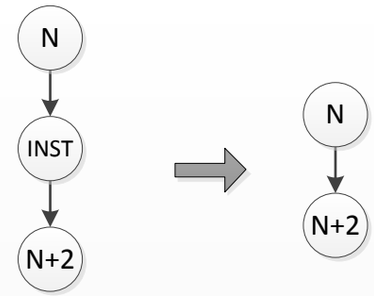
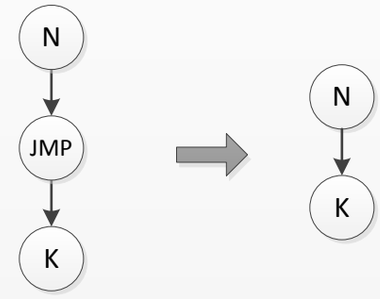
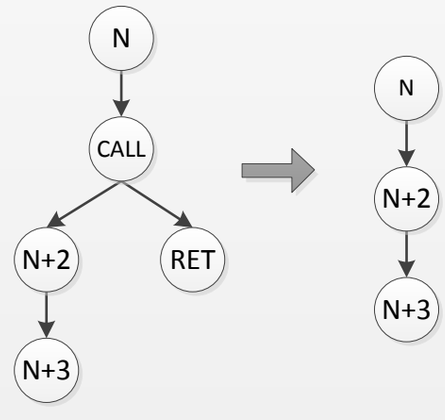
- Une vue plus abstraite des programmes

```
test eax, eax
jz short 424BED
jmp 424D75
mov [ebp+var_1110],1
jmp short 424C08
...
pop     edi
mov     esp, ebp
pop     ebp
retn
```



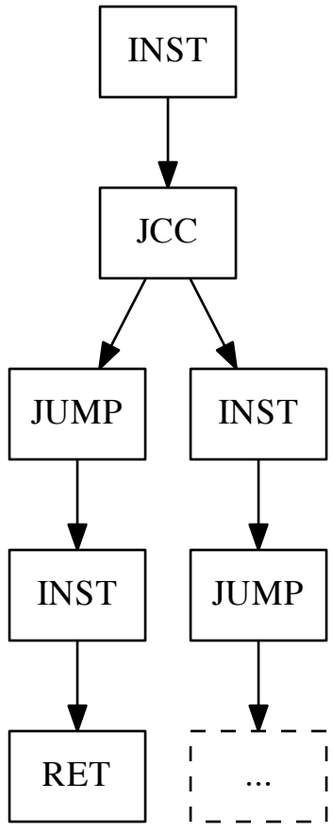


Réduction des signatures par réécriture de graphe

Remove sequential instructions	
Realign jumps	
Remove false calls	

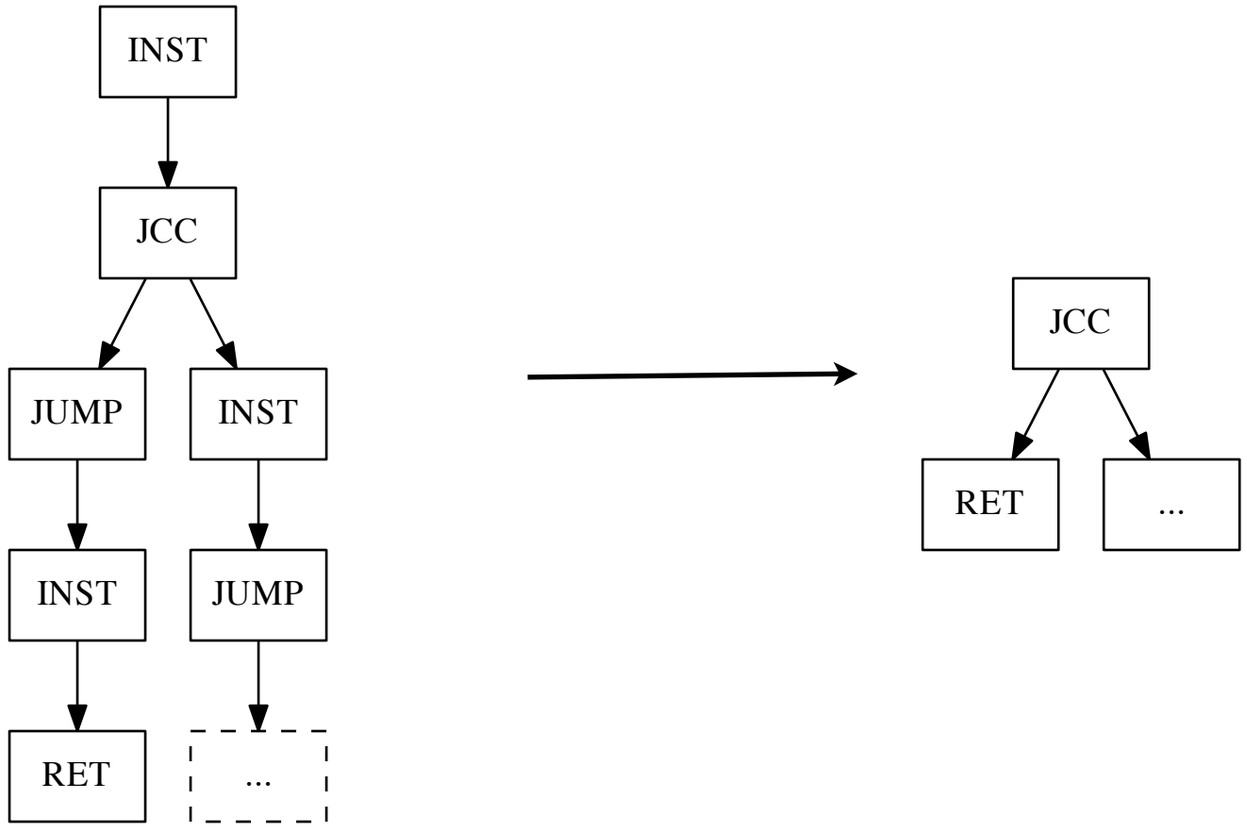
Analyse morphologique

- Une vue plus abstraite sur les programmes



Analyse morphologique

- Une vue plus abstraite sur les programmes

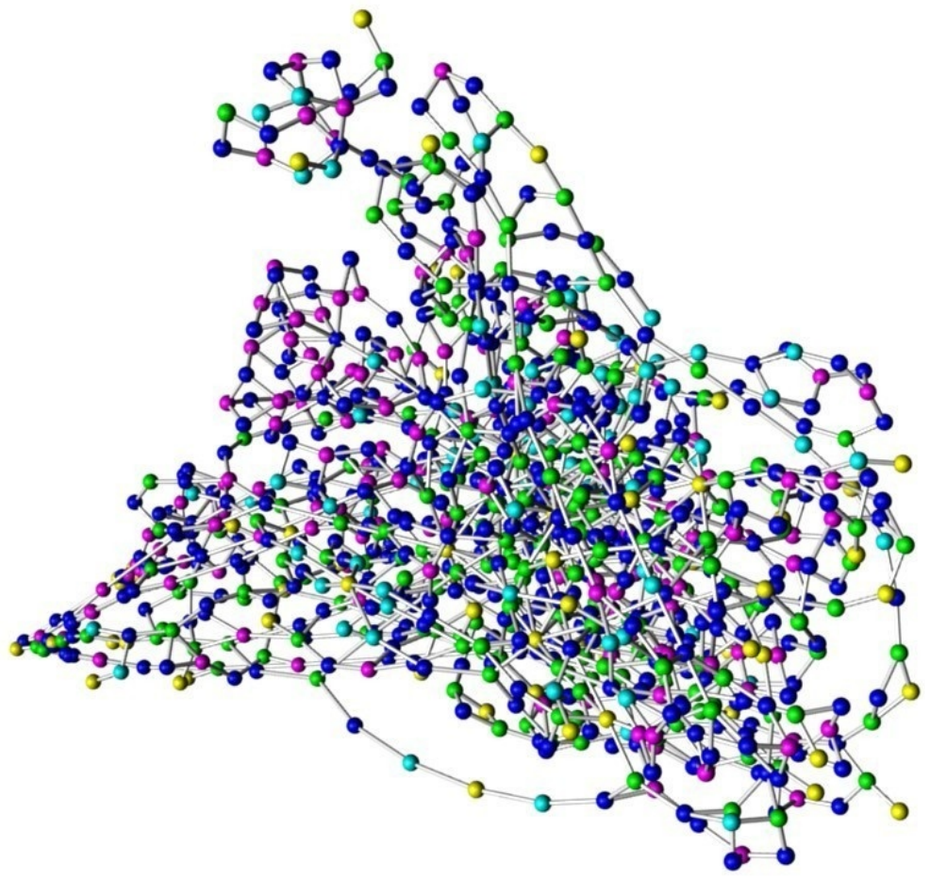


Reconstruction automatique de signature

```
mov [rbx], eax
push qword [rbp+0x10]
push rax
call 0xddf
mov [rbx+0x4], edi
push qword [rbp+0x18]
pop qword [rbx+0xc]

pop rdi
pop rsi
pop rbx
leave
ret 0x14
mov [rdx], ebx
jmp 0x9
inc dword [rip+0x40812a]
cmp dword [rbx], 0x0
jnz 0x1d
inc dword [rbx]
push qword [rip+0x408146]
call 0x9f5
jmp qword near [rip+0x404070]
pop rbx
leave
ret 0xc
lea eax, [rbp-0x4]
push rax
push 0x0
push rbx
push dword 0x402778
push 0x0
push 0x0
call 0x984
push rax
call 0x95a
mov dword [rip+0x40812a], 0x0
push rsi
call 0xde4
push qword [rbp+0x8]
push dword 0x40814e
call 0x740
push qword [rbp+0x8]
push qword [rbp+0x8]
call 0xfffffffffffff8fb
jmp 0xf
leave
ret 0x4
push dword 0x4057e5
push qword [rbp+0x8]
call 0x66a
push qword [rbp+0xc]
push qword [rbp+0x8]
call 0xfffffffffffff70
jmp 0xa
push qword [rbp-0x4]
call 0x5b3
leave
ret 0x4
push dword 0xafc8
call 0xffffffffffffe106
add eax, 0x1388
invalid
```

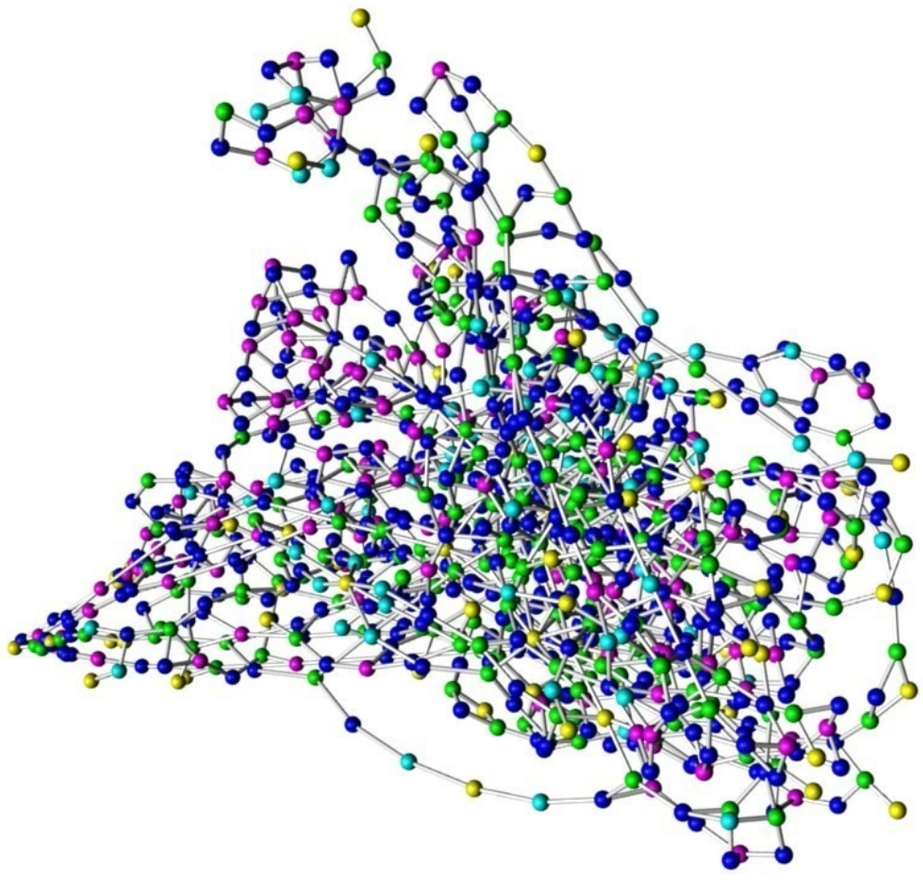
```
Sample name: Email-Worm.Win32.Bagle.a
Number of nodes: 1022
```



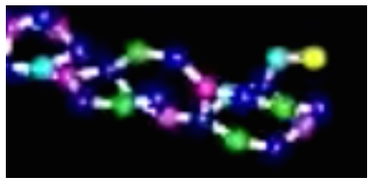
Reconstruction automatique de signature

```
mov [rbx], eax
push qword [rbp+0x10]
push rax
call 0xddf
mov [rbx+0x4], edi
push qword [rbp+0x18]
pop qword [rbx+0xc]

pop rdi
pop rsi
pop rbx
leave
ret 0x14
mov [rdx], ebx
jmp 0x9
inc dword [rip+0x40812a]
cmp dword [rbx], 0x0
jnz 0x1d
inc dword [rbx]
push qword [rip+0x408146]
call 0x9f5
jmp qword near [rip+0x404070]
pop rbx
leave
ret 0xc
lea eax, [rbp-0x4]
push rax
push 0x0
push rbx
push dword 0x402778
push 0x0
push 0x0
call 0x984
push rax
call 0x95a
mov dword [rip+0x40812a], 0x0
push rsi
call 0xde4
push qword [rbp+0x8]
push dword 0x40814e
call 0x740
push qword [rbp+0x8]
push qword [rbp+0x8]
call 0xfffffffffffff8fb
jmp 0xf
leave
ret 0x4
push dword 0x4057e5
push qword [rbp+0x8]
call 0x66a
push qword [rbp+0xc]
push qword [rbp+0x8]
call 0xffffffffffffff70
jmp 0xa
push qword [rbp-0x4]
call 0x5b3
leave
ret 0x4
push dword 0xafc8
call 0xfffffffffffe106
add eax, 0x1388
invalid
```



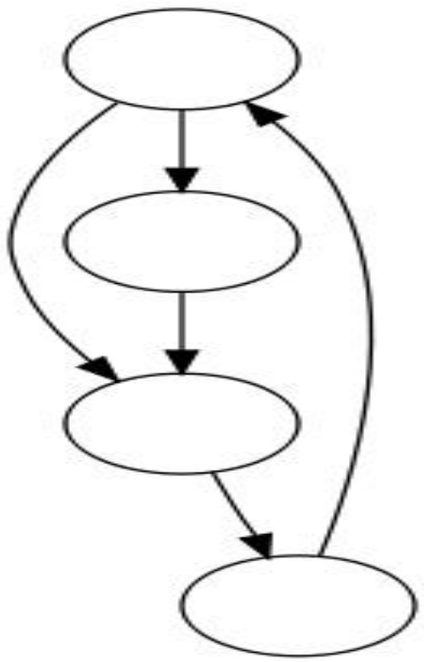
```
Sample name: Email-Worm.Win32.Bagle.a
Number of nodes: 1022
```



L’analyse morphologique en un clin d’oeil

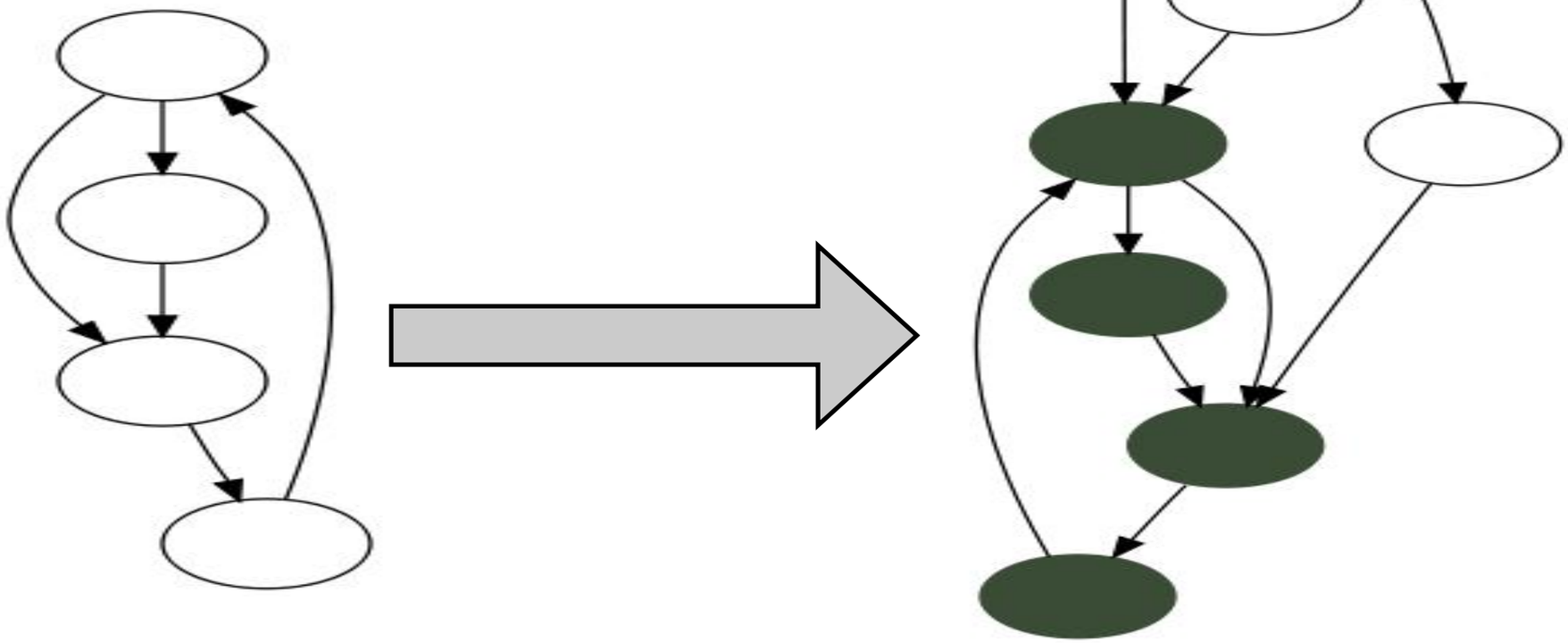
L’analyse morphologique en un clin d’oeil

Les signatures sont des graphes



L’analyse morphologique en un clin d’oeil

Les signatures sont des graphes

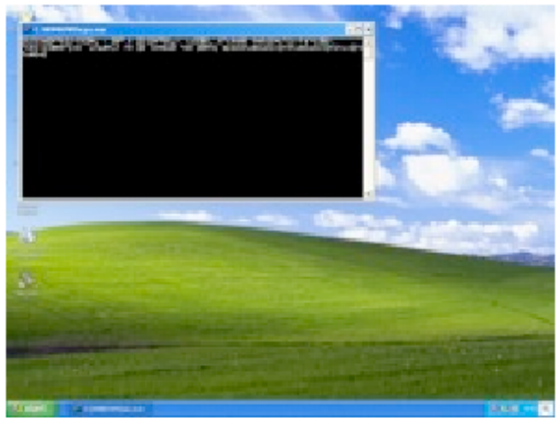


La détection d’un graphe dans un programme témoigne de l’infection

Stuxnet, Duqu, Flame, Waledac et Regin

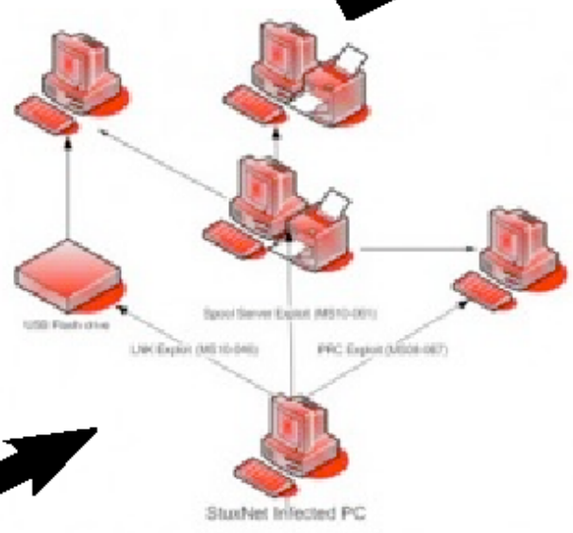
Ingénierie Sociale

Infection

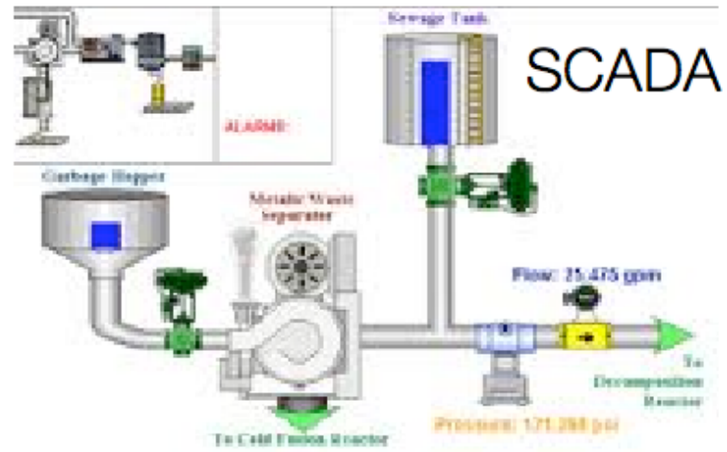


0-day exploit
Faille de sécurité MS10-046

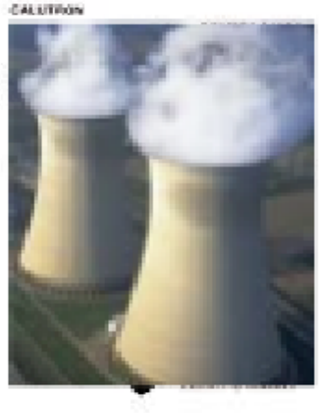
Attaque ciblée



Propagation



Destruction



Historique
Début en Juin 2009
Activité Mars et Avril 2010
Connaissance en Juillet

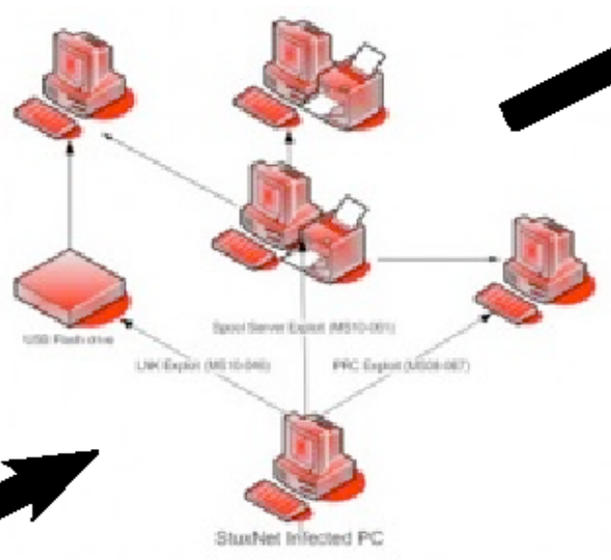
Duqu

Ingénierie Sociale

Infection

0-day exploit
Faille de sécurité MS11-087

Propagation



Vol d'informations

dsc00001.jpg

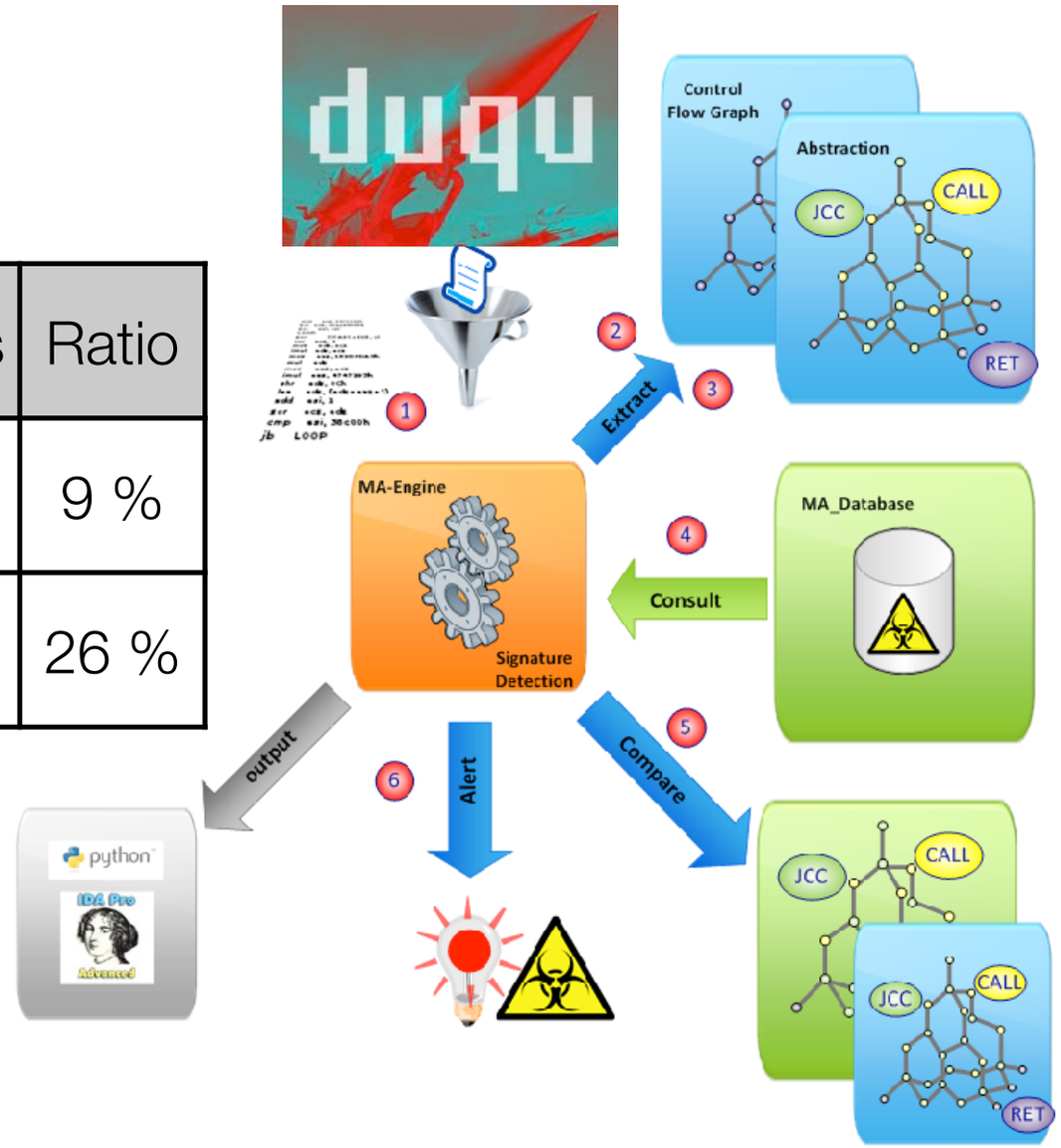
Historique
Début Juin 2010 ?
Découverte en Sept. 2011 par Crysys (Budapest)

En route,

```
aurelien:~/R$ ./sigtool --learn-sub --reduction ola duqu.db netp191_Decrypted.int
LEARN_OK: "netp191_Decrypted.int", graph is 5863 nodes
aurelien:~/R$ ./sigtool --dist --reduction ola duqu.db maindll.decrypted.unpacked.dll_
DIST: "maindll.decrypted.unpacked.dll_":
        100.0% (826/826), 26.46% (826/3121): "netp191_Decrypted.int"
```

La détection de Duqu dans Stuxnet

	taille	sites	communs	Ratio
Stuxnet	1.10^6	9032	826	9 %
Duqu	3.10^5	3121	826	26 %



La correspondance, en détail

maindll.decrypted.	subroutine	netp191_Decrypt	subroutine
10042DB0	sub_10042CD2	100136DB	sub_100135FD
10042DC0	sub_10042CD2	100136EB	sub_100135FD
10042DC5	sub_10042CD2	100136F0	sub_100135FD
10042DD3	sub_10042CD2	100136FE	sub_100135FD
10042DE0	sub_10042CD2	1001370B	sub_100135FD
10043116	msvcr80$__beginthreadex	1001353F	msvcr80$__beginthreadex
1004311F	msvcr80$__beginthreadex	10013548	msvcr80$__beginthreadex
10043138	msvcr80$__beginthreadex	10013561	msvcr80$__beginthreadex
1004314D	msvcr80$__beginthreadex	10013576	msvcr80$__beginthreadex
10043155	msvcr80$__beginthreadex	1001357E	msvcr80$__beginthreadex
10043157	msvcr80$__beginthreadex	10013580	msvcr80$__beginthreadex
10043161	msvcr80$__beginthreadex	1001358A	msvcr80$__beginthreadex
1004317B	msvcr80$__beginthreadex	100135A4	msvcr80$__beginthreadex
10043183	msvcr80$_free	100144E3	msvcr80$_free
1004318D	msvcr80$_free	100144ED	msvcr80$_free

Table 3

DLL Exports

Export #	Function
1	Infect connected removable drives, starts RPC server
2	Hooks APIs for Step 7 project file infections
4	Calls the removal routine (export 18)
5	Verifies if the threat is installed correctly
6	Verifies version information
7	Calls Export 6
9	Updates itself from infected Step 7 projects
10	Updates itself from infected Step 7 projects
14	Step 7 project file infection routine
15	Initial entry point
16	Main installation
17	Replaces Step 7 DLL
18	Uninstalls Stuxnet
19	Infects removable drives
22	Network propagation routines
24	Check Internet connection
27	RPC Server
28	Command and control routine
29	Command and control routine
31	Updates itself from infected Step 7 projects
32	Same as 1

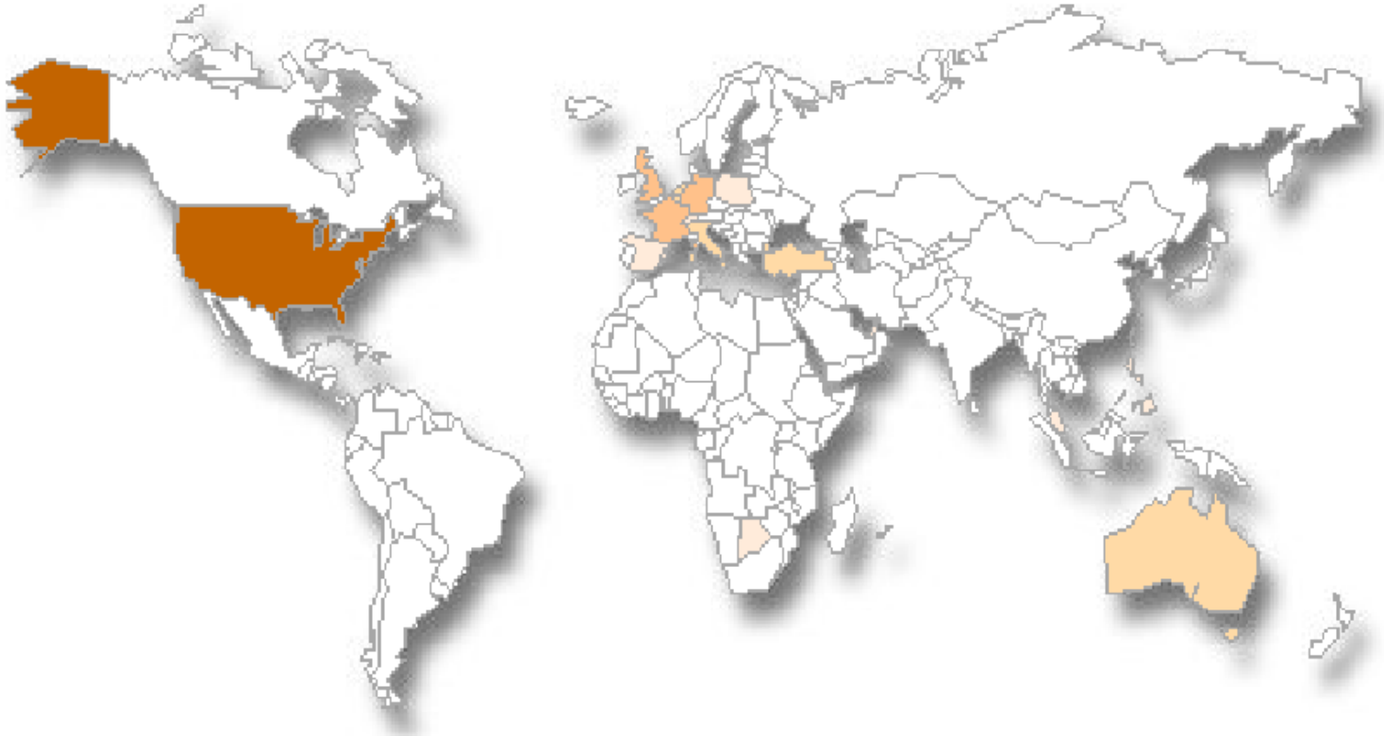
DLL Exports

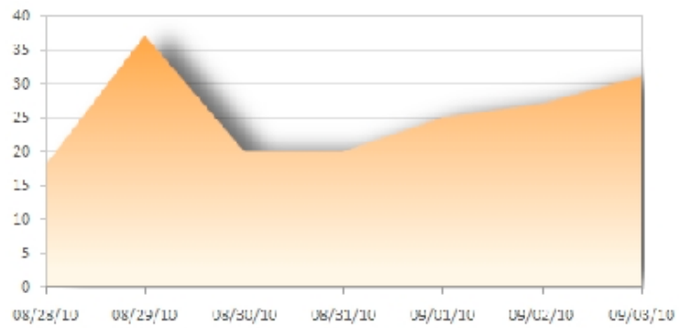
Export #	Function
1	Initialize the data
2	Run export number 6
3	Get the version information from the configuration data
4	Inject itself into a suitable process and run export 5 (only if on a 32bit platform)
5	System setup • Pre-install: Drop the provided load-point driver and create service • Post-install: Load the resource 302 DLL (resource 302 is a loader for the main payload)
6	Cleanup routine
7	Start the RPC component
8	he same as export 1, but with a delay timer

Waledac, un robot à pourriel

Wikipédia

- Waledac est présenté comme l'un des dix plus importants botnets aux État-Unis, mais constitué de centaines de milliers d'ordinateurs infectés à travers le monde. Des ordinateurs zombies principalement utilisés à leur insu pour inonder la planète de spam, à raison de plus de **1,5 milliard** de messages par jour ! (soit **1%** du volume total de spam sur Internet)



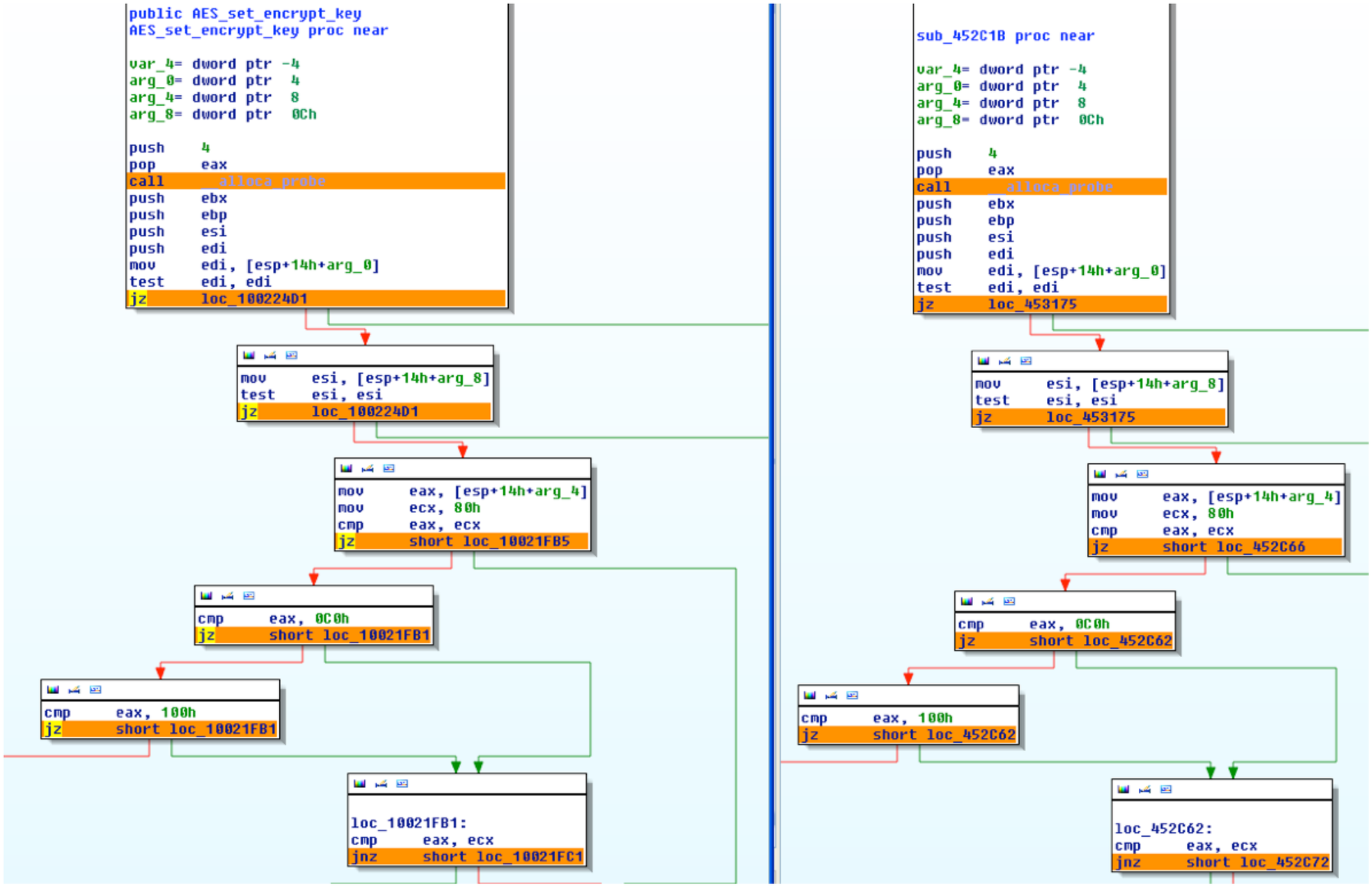


Waledac, une étude du code

Aucune identification des fonctions usuelles de la Libc (outil IDA de Hex-rays)

```
.text:00452C1B
.text:00452C1B sub_452C1B      proc near               ; CODE XREF: sub
.text:00452C1B                                         ; sub_45317E+16↓
.text:00452C1B
.text:00452C1B var_4           = dword ptr -4
.text:00452C1B arg_0           = dword ptr  4
.text:00452C1B arg_4           = dword ptr  8
.text:00452C1B arg_8           = dword ptr  0Ch
.text:00452C1B
.text:00452C1B                 push    4
.text:00452C1D                 pop     eax
.text:00452C1E                 call    __alloca_probe
.text:00452C23                 push    ebx
.text:00452C24                 push    ebp
.text:00452C25                 push    esi
.text:00452C26                 push    edi
.text:00452C27                 mov     edi, [esp+14h+arg_0]
.text:00452C2B                 test    edi, edi
.text:00452C2D                 jz      loc_453175
.text:00452C33                 mov     esi, [esp+14h+arg_8]
.text:00452C37                 test    esi, esi
.text:00452C39                 jz      loc_453175
.text:00452C3F                 mov     eax, [esp+14h+arg_4]
.text:00452C43                 mov     ecx, 80h
.text:00452C48                 cmp     eax, ecx
.text:00452C4A                 jz      short loc_452C66
.text:00452C4C                 cmp     eax, 0C0h
.text:00452C51                 jz      short loc_452C62
.text:00452C53                 cmp     eax, 100h
.text:00452C58                 jz      short loc_452C62
.text:00452C5A                 push    0FFFFFFFEh
.text:00452C5C                 pop     eax
.text:00452C5D                 jmp     loc 453178
```

Waledac utilise une librairie de cryptographie



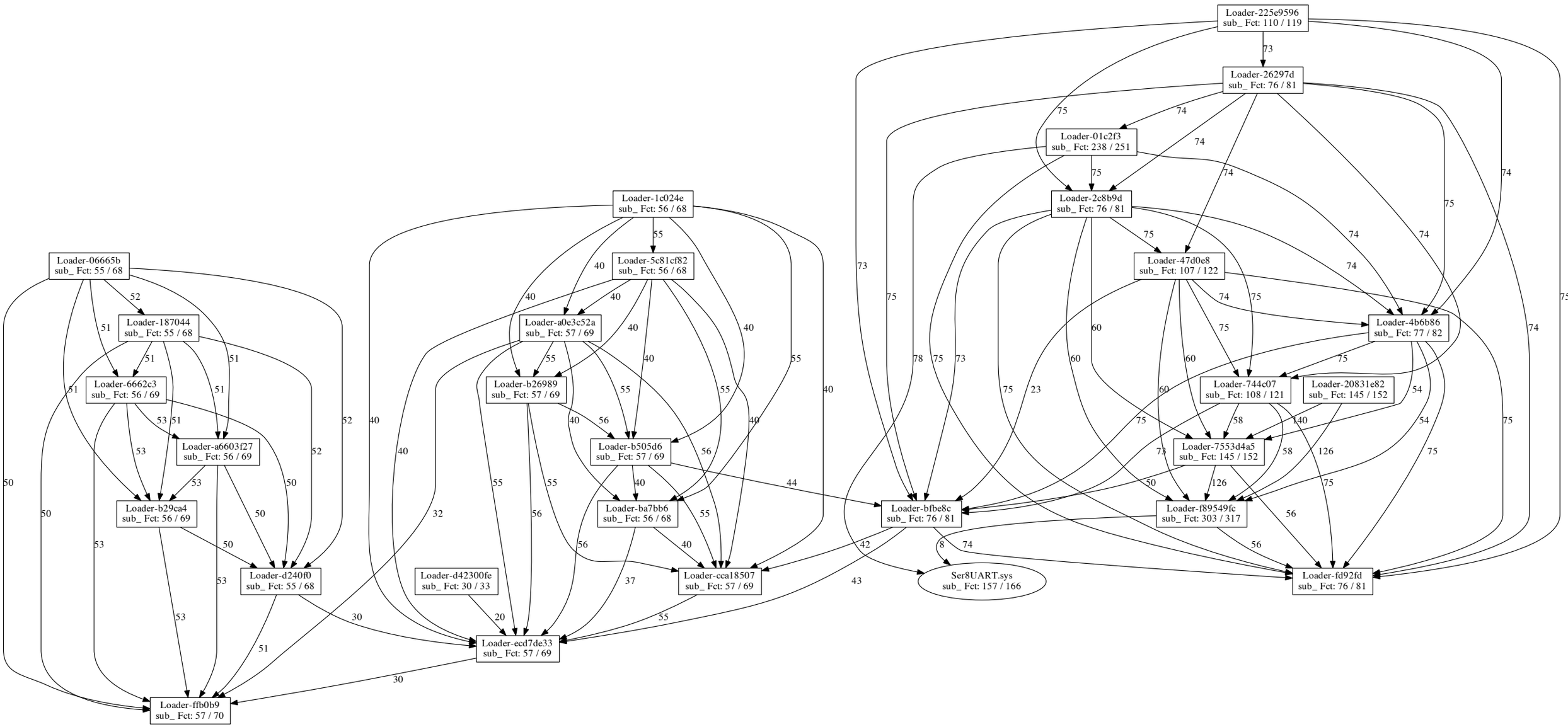
Waledac et ses fonctions cryptographiques

OpenSSL (libeay32-098e.dll)		Waledac (Waledac48.int)	
Adresse	Fonction	Adresse	Fonction
10002CDE	CRYPTO_new_ex_data	00455B5C	sub_455B55
10002CD0	CRYPTO_new_ex_data	00455B5E	sub_455B55
10002D0A	CRYPTO_new_ex_data	00455B72	sub_455B55
10002D0C	CRYPTO_new_ex_data	00455B74	sub_455B55
10021F6D	AES_set_encrypt_key	00452C1E	sub_452C1B
10021F7C	AES_set_encrypt_key	00452C2D	sub_452C1B
10021F88	AES_set_encrypt_key	00452C39	sub_452C1B
10021F99	AES_set_encrypt_key	00452C4A	sub_452C1B
10021FA0	AES_set_encrypt_key	00452C51	sub_452C1B
100224E0	AES_set_decrypt_key	00453184	sub_45317E
100224F0	AES_set_decrypt_key	00453194	sub_45317E
100224FA	AES_set_decrypt_key	0045319E	sub_45317E

Des détails qui comptent

- Waledac utilise une librairie OpenSSL de 1997 (0.9.8e)
- La librairie a été optimisée pour gain d’espace,
- Fonctions de cryptage
  - AES_set_encrypt_key, AES_set_decrypt_key
  - X509_PUBKEY_set, X509_PUBKEY_get
  - RSA / DSA : RSA_free, DSA_size, DSA_new_method

Regin en famille



Quelques conclusions sur l'analyse morphologique

- De bons tests
- Points positifs
  - Une grande robustesse aux mutations
  - Tient compte de la sémantique des programmes
  - Méthode quasi-automatique
  - Utilisable à différents niveaux d'analyse
- Points négatifs
  - Demande une couverture importante du code
  - Vitesse de calcul encore lente