APT3 Uncovered:
The code evolution of Pirpi

Your functions are showing (and leaving a trail)

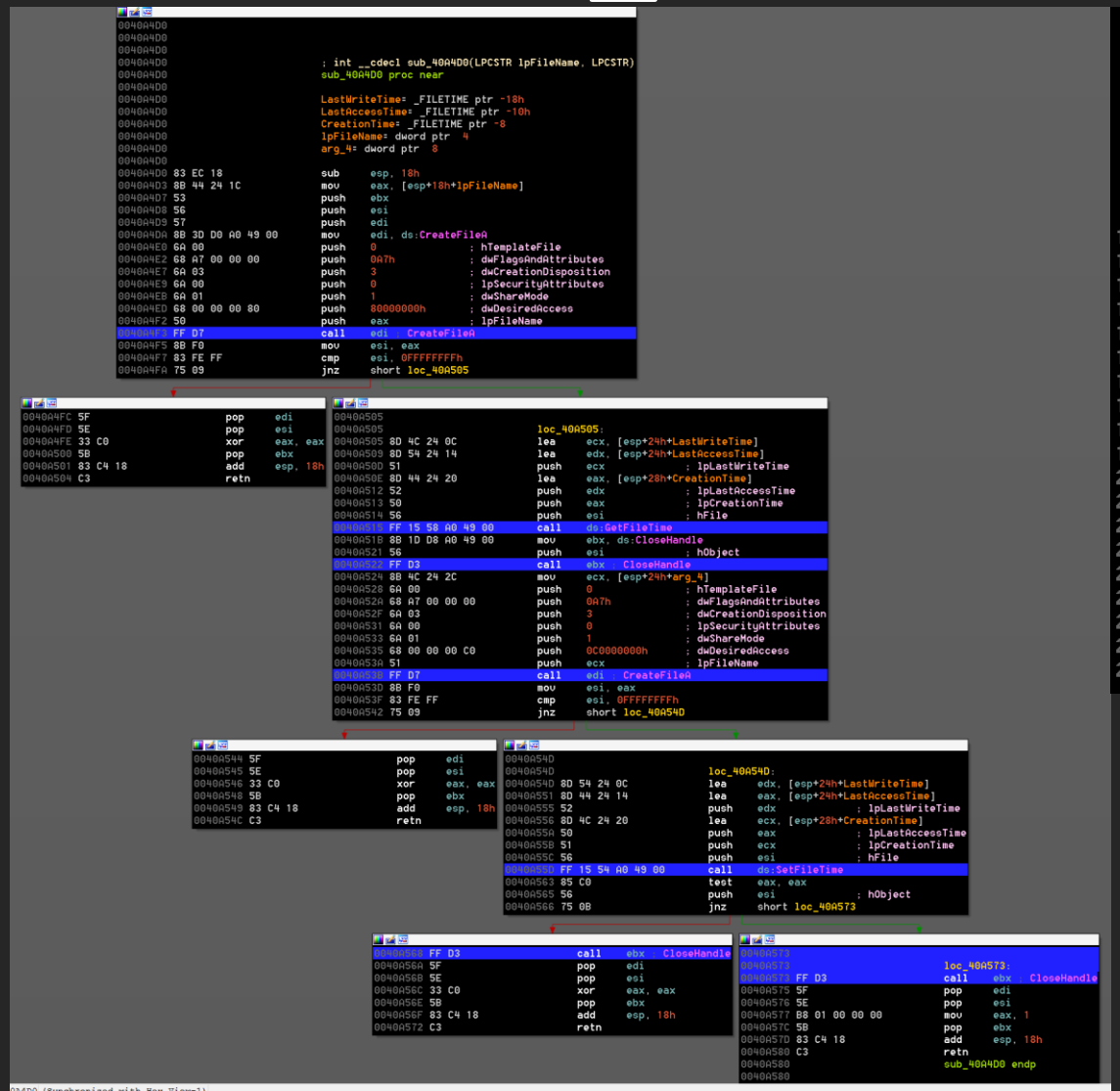
Micah Yates - @m1k4chu

Palo Alto Networks

Your functions are showing

- Persistent Code Reuse
- Finding The Trail
- Pirpi.201x vs Operation Clandestine DoubleTap
- Just give me all the Pirpi you have
- EXEProxy

Persistent Code Reuse: function_1



```
signed int __usercall sub_10006DE0@<eax>(int _EBX@<ebx>, void *a2)
{
  void *v3; // esi@1
  HANDLE v5; // esi@3
  struct _FILETIME LastWriteTime; // [esp+0h] [ebp-30h]@1
  struct _FILETIME LastAccessTime; // [esp+8h] [ebp-28h]@1
  struct _FILETIME CreationTime; // [esp+10h] [ebp-20h]@3
  LPCSTR lpFileName; // [esp+20h] [ebp-10h]@3

  _EAX = a2;
  LastAccessTime.dwLowDateTime = 0;
  LastWriteTime.dwHighDateTime = 167;
  LastWriteTime.dwLowDateTime = 3;
  BYTE1(_EBX) *= 2;
  __asm { xlat }
  v3 = _EAX;
  if ( _EAX == (void *)-1 )
    return 0;
  GetFileTime(_EAX, &CreationTime, &LastAccessTime, &LastWriteTime);
  CloseHandle(v3);
  v5 = CreateFileA(lpFileName, 0xC0000000, 1u, 0, 3u, 0xA7u, 0);
  if ( v5 == (HANDLE)-1 )
    return 0;
  if ( !SetFileTime(&LastWriteTime, (const FILETIME *)0x80000000, (const FILETIME *)1, 0) )
    return 0;
  CloseHandle(v5);
  return 1;
}
```

Persistent Code Reuse

- What is Pirpi?
  - Targeted malware delivered via 0-days
    - (CVE-2010-3962)
    - (CVE-2014-1776)
    - (CVE-2014-4113)
    - (CVE-2014-6332)
    - (CVE-2015-3113)
    - (CVE-2015-5119)
  - Appended to a valid gif
  - Obfuscated Command and Control
  - Compromised infrastructure

Finding a Trail

- Why are you doing this?
  - It’s fun
  - You don’t know what you don’t know

- Pick 2 binaries by:
  - Family
  - Compile Times
  - File Size
  - % similarity (ssdeep)

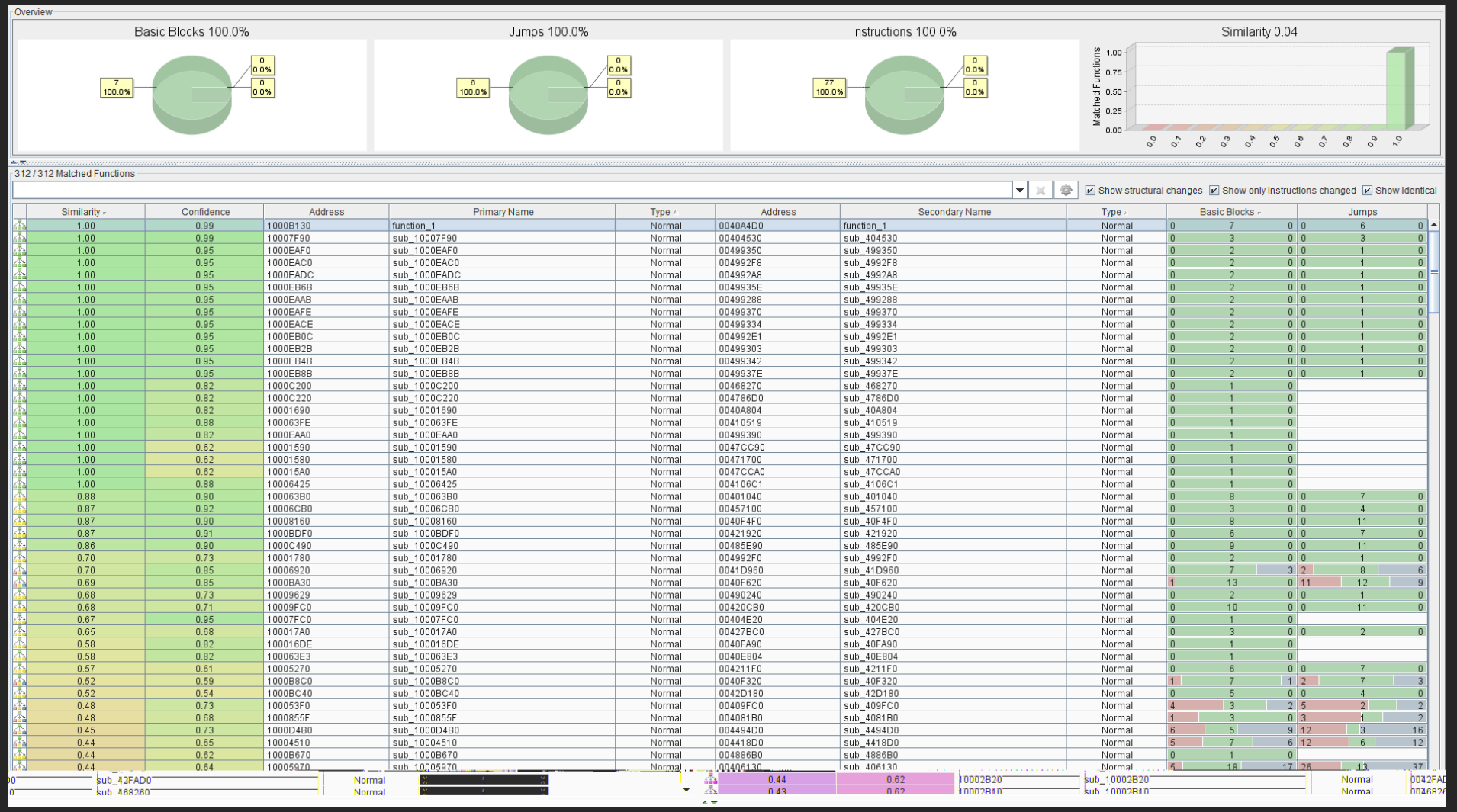
It started out with a diff, how did it end up like this?

- Inspect Binaries w/ IDA Tools
  - Bindiff – Hex Rays
  - IDAScope – Plohmann/Hanel

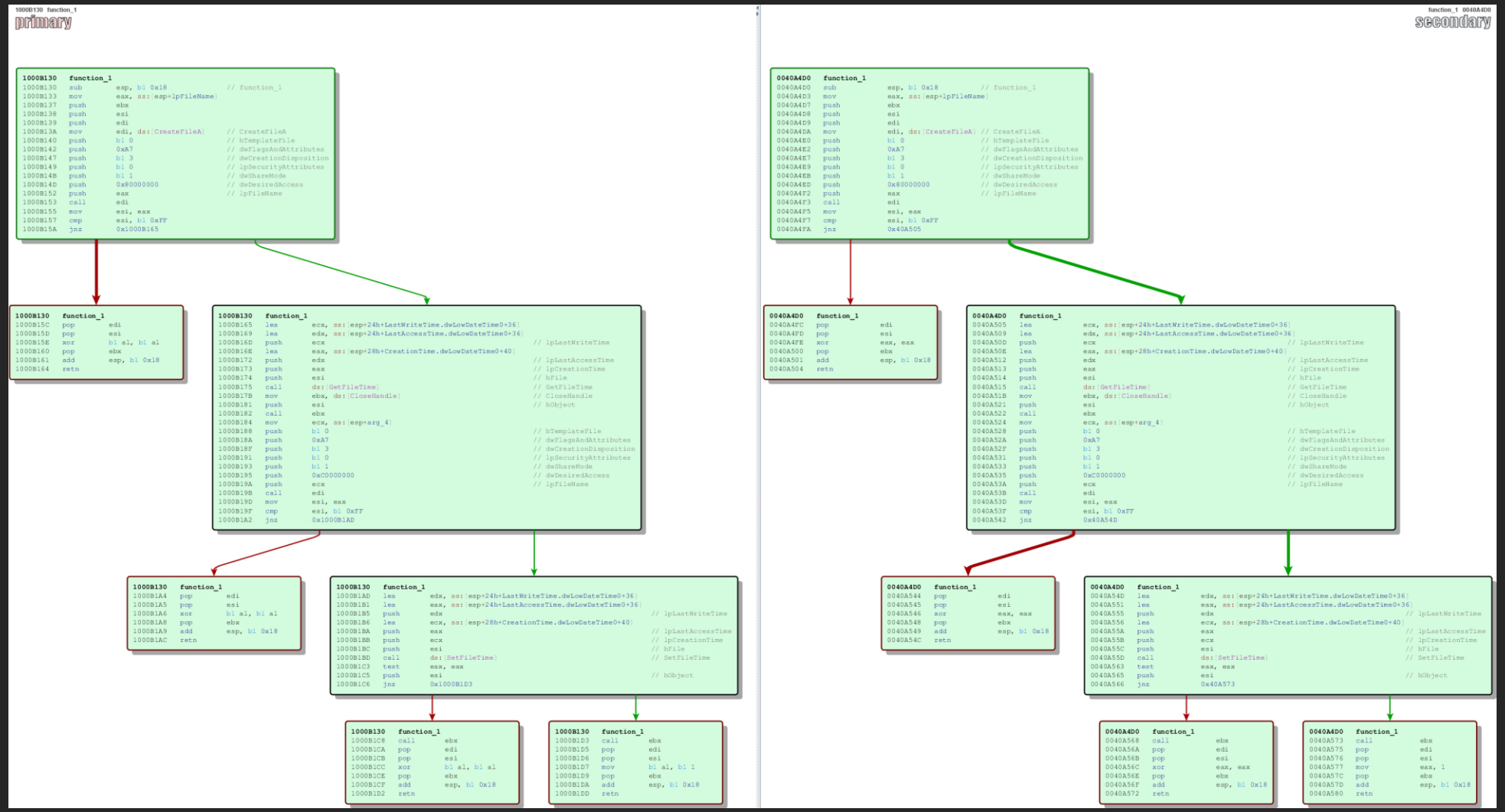
- Bindiff two known Pirpi samples
  - fb838cda6118a003b97ff3eb2edb7309
  - e33804e3e15920021c5174982dd69890

File	Ratio	First sub.	Last sub.	Times sub.	Sources	Size
89390b83250cdf898d6eb627e035bc7b1202aa6bbbc8fd394223da2d4f7317a8 fb838cda6118a003b97ff3eb2edb7309 armadillo pedll via-tor	35 / 55	2014-05-01 18:12:26	2014-07-25 17:04:41	3	3	84.0 KB
4ce90c9a49ee026618ffe9741d958a764419389492e4c231e61f0a6f4932789c e33804e3e15920021c5174982dd69890 peexe	7 / 56	2015-08-25 22:44:02	2015-09-09 14:34:55	4	2	788.0 KB

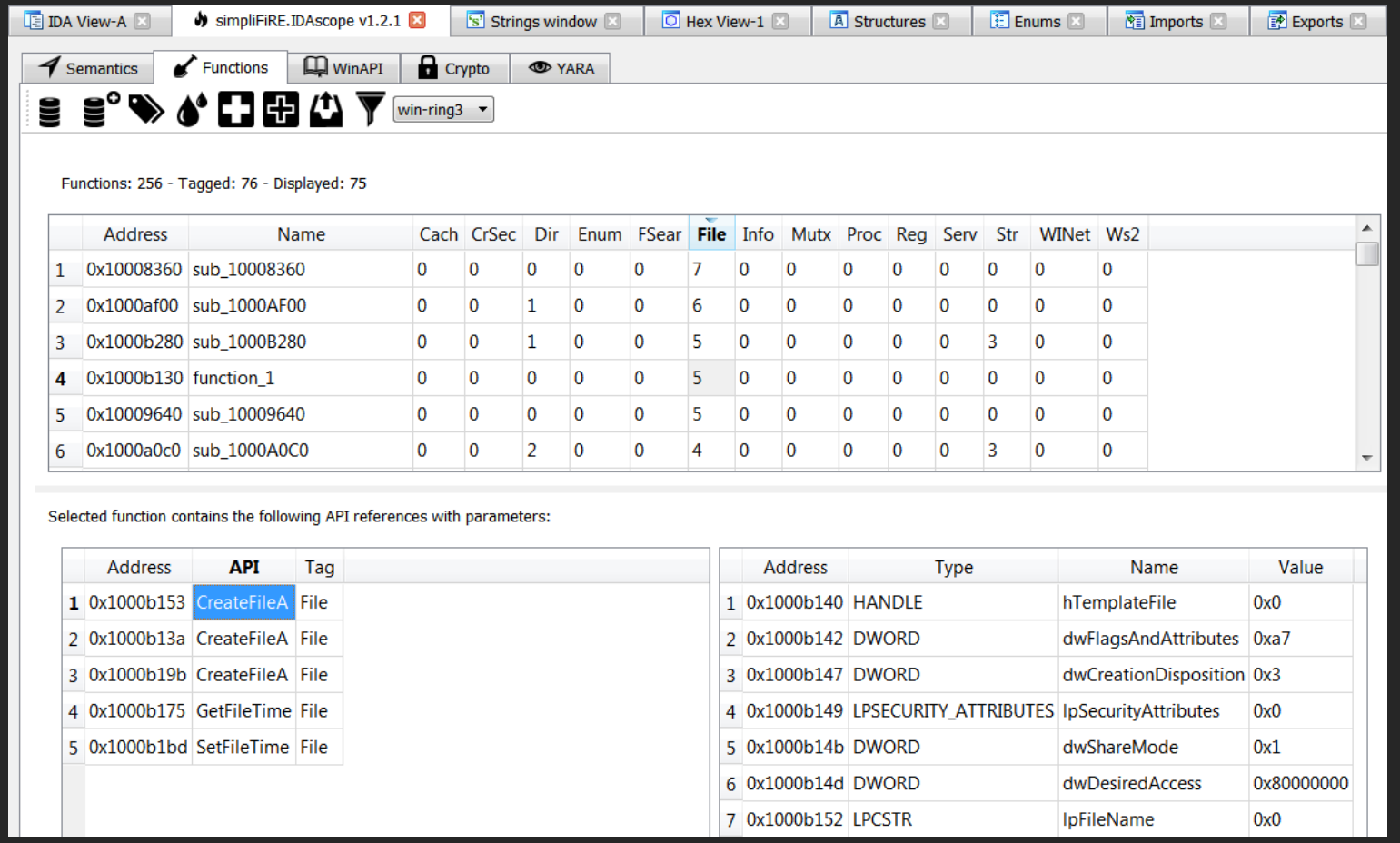
It was only a diff



It was only a diff



Aside:
Putting the FUN in Function



	Address	Name	Cach	CrSec	Dir	Enum	FSear	File	Info	Mutx	Proc	Reg	Serv	Str	WINet	Ws2
1	0x10008360	sub_10008360	0	0	0	0	0	7	0	0	0	0	0	0	0	0
2	0x1000af00	sub_1000AF00	0	0	1	0	0	6	0	0	0	0	0	0	0	0
3	0x1000b280	sub_1000B280	0	0	1	0	0	5	0	0	0	0	0	3	0	0
4	0x1000b130	function_1	0	0	0	0	0	5	0	0	0	0	0	0	0	0
5	0x10009640	sub_10009640	0	0	0	0	0	5	0	0	0	0	0	0	0	0
6	0x1000a0c0	sub_1000A0C0	0	0	2	0	0	4	0	0	0	0	0	3	0	0

	Address	API	Tag
1	0x1000b153	CreateFileA	File
2	0x1000b13a	CreateFileA	File
3	0x1000b19b	CreateFileA	File
4	0x1000b175	GetFileTime	File
5	0x1000b1bd	SetFileTime	File

	Address	Type	Name	Value
1	0x1000b140	HANDLE	hTemplateFile	0x0
2	0x1000b142	DWORD	dwFlagsAndAttributes	0xa7
3	0x1000b147	DWORD	dwCreationDisposition	0x3
4	0x1000b149	LPSECURITY_ATTRIBUTES	lpSecurityAttributes	0x0
5	0x1000b14b	DWORD	dwShareMode	0x1
6	0x1000b14d	DWORD	dwDesiredAccess	0x80000000
7	0x1000b152	LPCSTR	lpFileName	0x0

Pirpi.201x vs Operation Clandestine DoubleTap

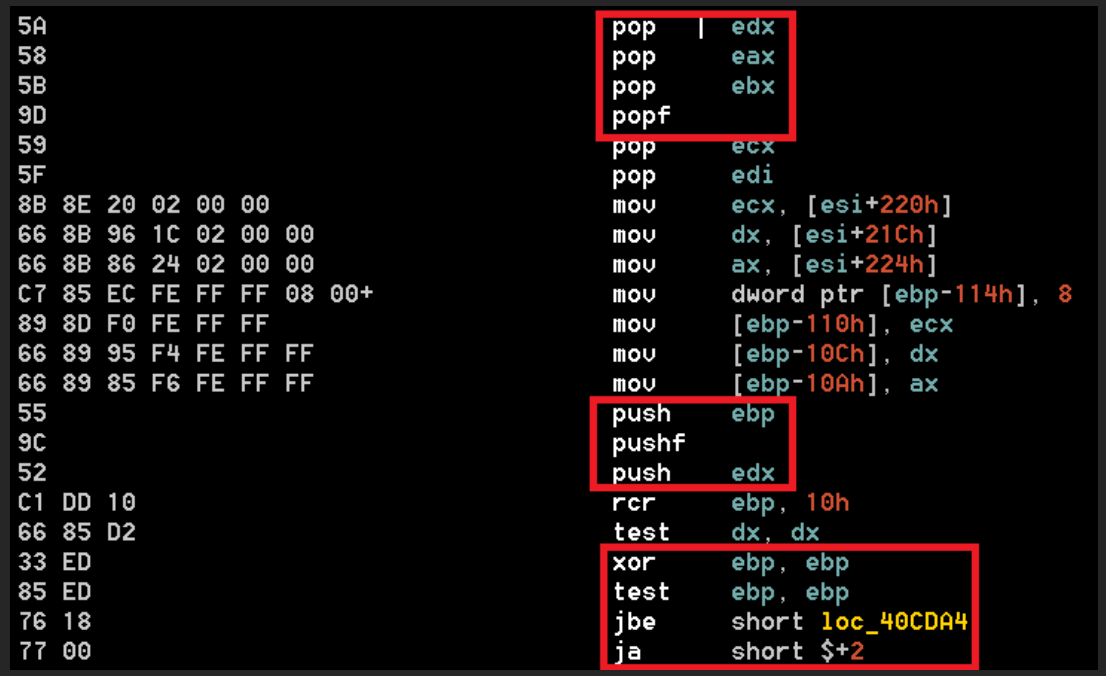
Sample MD5	Pirpi.2014 Bindiff		Pirpi.2015 Bindiff	
	Similarity	Confidence	Similarity	Confidence
FA3578C2ABE3F37DDDA76EE40C5A1608	89.5%	98.6%	29.8%	69.5%
1A4B710621EF2E69B1F7790AE9B7A288	92.7%	98.8%	29.4%	69.5%
F4884C0458176AAC848A911683D3DEF5	91.4%	98.7%	29.6%	71.6%
4CA97FF9D72B422589266AA7B532D6E6	93.7%	98.7%	30.7%	71.6%
B48E578F030A7B5BB93A3E9D6D1E2A83	100%	100%	34.3%	73.0%
1B0E6BA299A522A3B3B02015A3536F6F	34.3%	73.0%	100%	100%

Table 4. Resulting Similarity and Confidence Rates of Pirpi Samples

Name	Address	Ordinal
IePramGet	10002BC0	1
IeSet	10002B30	2
DllEntryPoint	1000E87A	[main entry]

File	Ratio	First sub.	Last sub.	Times sub.	Sources	Size
89390b83250cdf898d6eb627e035bc7b1202aa6bbbc8fd394223da2d4f7317a8fb838cda6118a003b97ff3eb2edb7309 armadillo pedll via-tor	35 / 55	2014-05-01 18:12:26	2014-07-25 17:04:41	3	3	84.0 KB
4ce90c9a49ee026618ffe9741d958a764419389492e4c231e61f0a6f4932789ce33804e3e15920021c5174982dd69890 peexe	7 / 56	2015-08-25 22:44:02	2015-09-09 14:34:55	4	2	788.0 KB

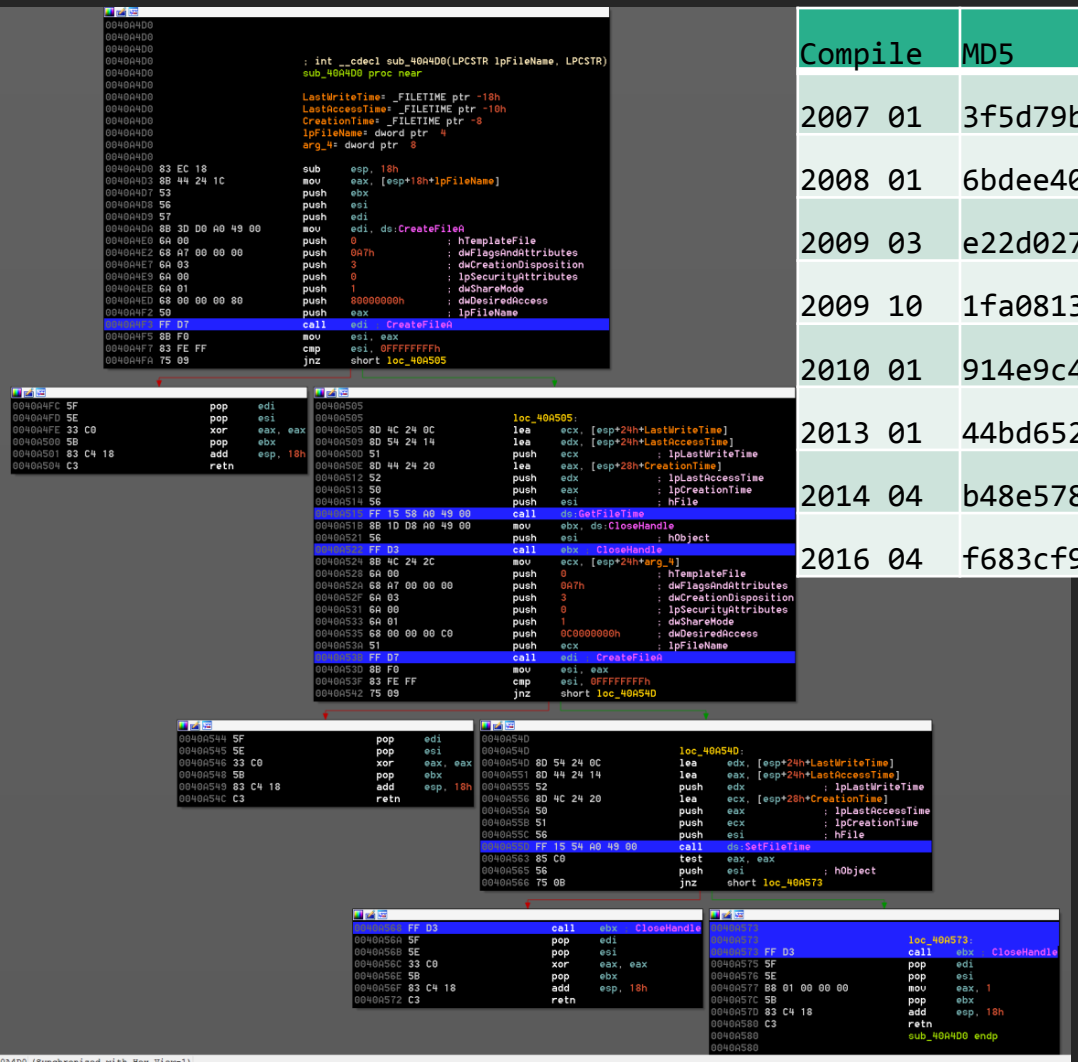
Pirpi.201x vs Operation Clandestine DoubleTap



```
5A                                      pop     edx
58                                      pop     eax
5B                                      pop     ebx
9D                                      popf
59                                      pop     ecx
5F                                      pop     edi
8B 8E 20 02 00 00                       mov     ecx, [esi+220h]
66 8B 96 1C 02 00 00                    mov     dx, [esi+21Ch]
66 8B 86 24 02 00 00                    mov     ax, [esi+224h]
C7 85 EC FE FF FF 08 00+                mov     dword ptr [ebp-114h], 8
89 8D F0 FE FF FF                       mov     [ebp-110h], ecx
66 89 95 F4 FE FF FF                    mov     [ebp-10Ch], dx
66 89 85 F6 FE FF FF                    mov     [ebp-10Ah], ax
55                                      push    ebp
9C                                      pushf
52                                      push    edx
C1 DD 10                                rcr     ebp, 10h
66 85 D2                                test    dx, dx
33 ED                                   xor     ebp, ebp
85 ED                                   test    ebp, ebp
76 18                                   jbe     short loc_40CDA4
77 00                                   ja      short $+2
```

```
rule purpi_anti_disasm
{
meta:
author = "Micah Yates"
description = "Anti-disassembly instructions used by Pirpi malware"
strings:
$re1 = { 5? 9c 5? } // push, pushf, push
$re2 = { 5? 5? 5? 9d ?? ?? } // 3x pop,  popf
$re3 = { 33 ?? 85 ?? 7? ?? 7? 00 } // xor, test, jge, jl
condition:
all of them
}
```

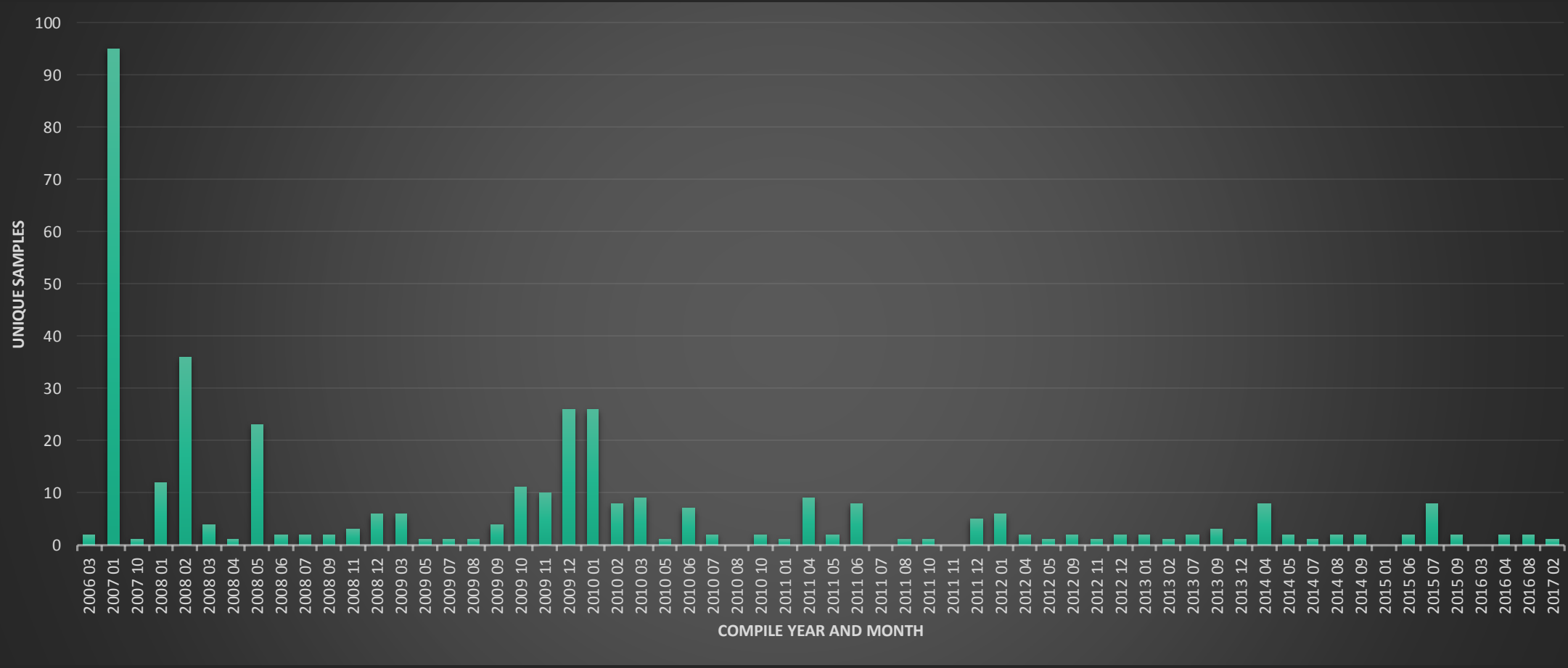
Just give me all the Pirpi you have: function_1



Compile	MD5	Beacon
2007 01	3f5d79b262472a12e3666118a7cdc2ca	www.msnmessengerupdate.com/index31382/f2ae95b93i97.bmp
2008 01	6bdee405ed857320aa8c822ee5e559f2	www.msnmessengerupdate.com/image19582/7e7e7eb7fi7f.gif
2009 03	e22d02796cfb908aaf48e2e058a0890a	www.office2008updates.com/dream/dream.php?
2009 10	1fa0813be4b9f23613204c94e74efc9d	ini.msnmessengerupdate.net/smart/smartmain.php?
2010 01	914e9c4c54fa210ad6d7ed4f47ec285f	ini.office2005updates.net/smart/smartmain.php?
2013 01	44bd652a09a991100d246d8280cac3ac	218.42.147.106/bussinesses/caetrwet/
2014 04	b48e578f030a7b5bb93a3e9d6d1e2a83	product.sorgerealty.com
2016 04	f683cf9c2a2fdc27abff4897746342c4	ste.mullanclan.com

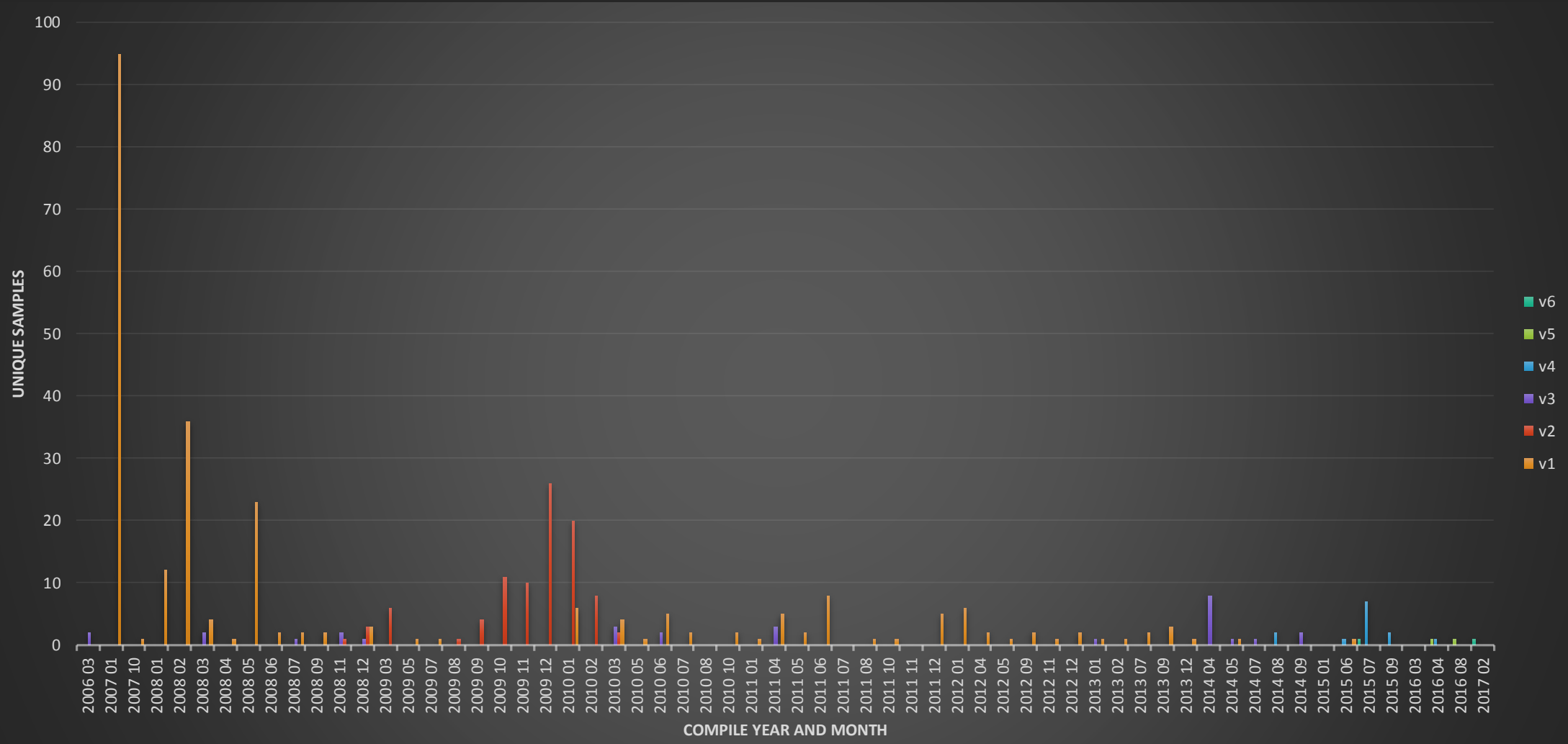
11monitor_MutexTest
MutexTest_IeInBrowser

Just give me all the Pirpi you have: function_1

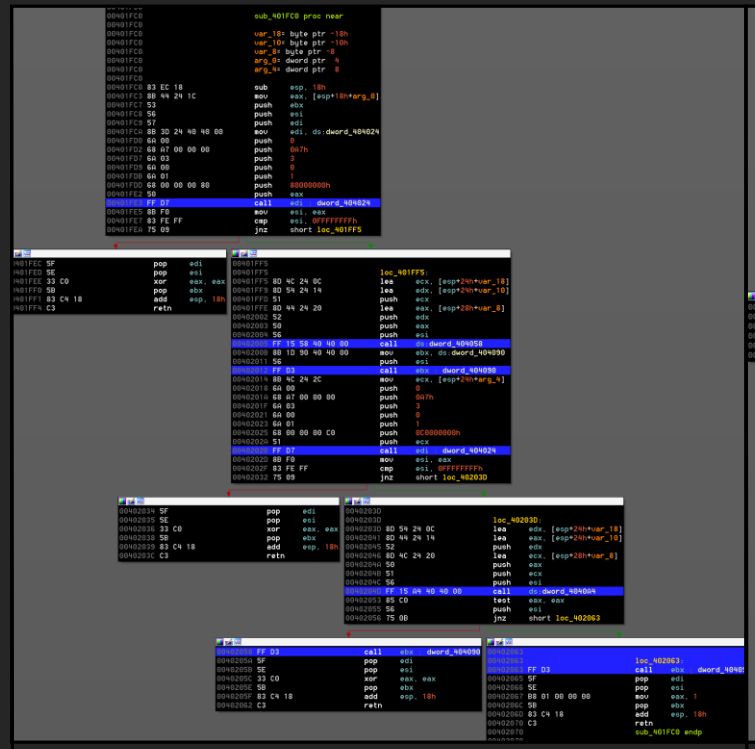



I got boxes full of ~~Pepe~~PIRPI!

function_1 - minor versions



function_1 - minor versions

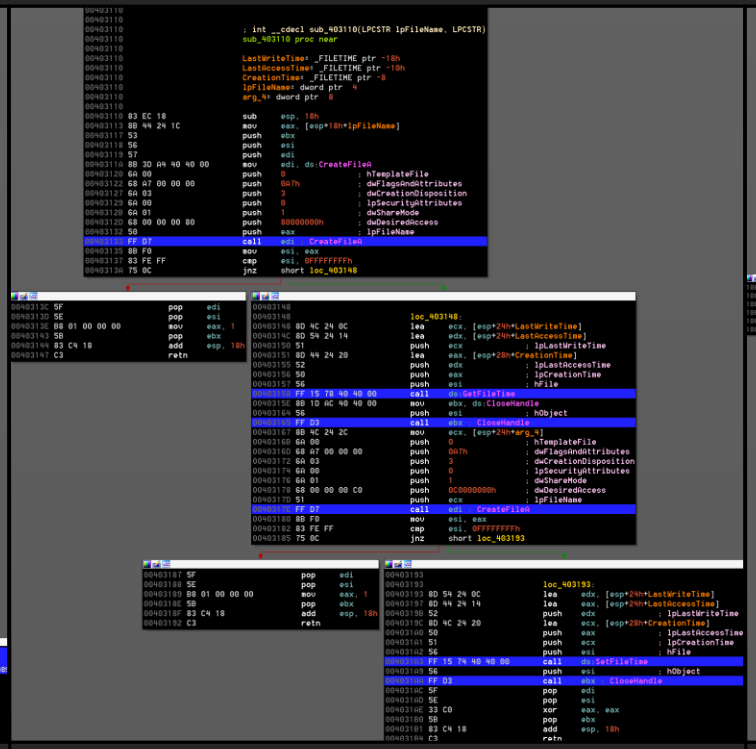


V1 - 2007

MD5:

98011f5b7b957a142f14cbda57a5ea82

@00401FC0

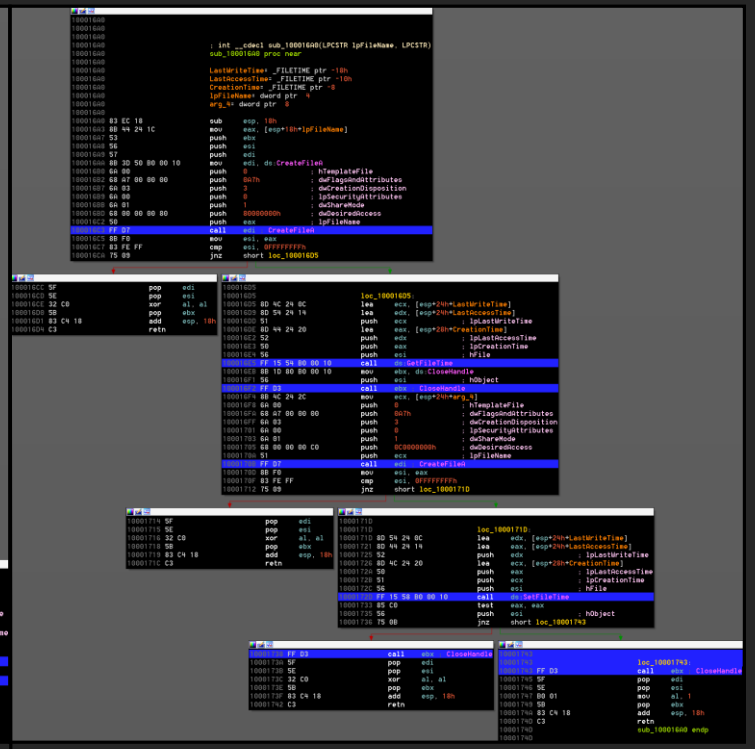


V2 - 2008

MD5:

272cb6c16e083ca143d40c63005753a2

@00403110



V3 – 2006

MD5:

acd8d34d8360129df1c8d03f253ba747

@100016A0

function_1 - minor versions

V4 - 2014

MD5:

c006faaf9ad26a0bd3bbd597947da3e1

@0040B320

V5 - 2016

MD5:

f683cf9c2a2fdc27abff4897746342c4

@00401D60

V6 - 2017

MD5:

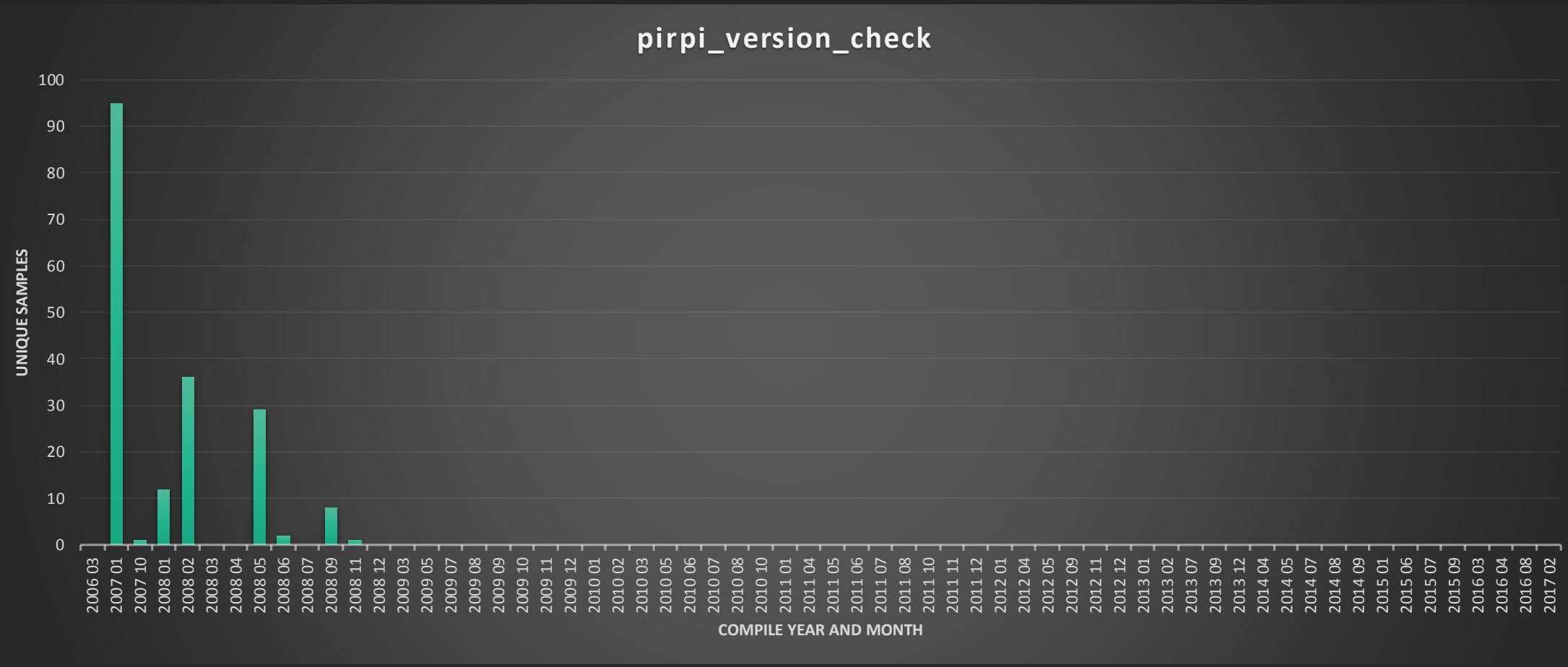
07b4d539a6333d7896493bafd2738321

@00404A20

function_2

```
1  signed int sub_402B30()
2  {
3    struct _OSVERSIONINFOA VersionInformation; // [esp+0h] [ebp-94h]@1
4
5    VersionInformation.dwOSVersionInfoSize = 148;
6    GetVersionExA(&VersionInformation);
7    if ( VersionInformation.dwPlatformId == 1 )
8    {
9      if ( VersionInformation.dwMajorVersion == 4 )
10     {
11       if ( !VersionInformation.dwMinorVersion )
12         return 95;
13       if ( VersionInformation.dwMinorVersion == 10 )
14         return 98;
15       if ( VersionInformation.dwMinorVersion == 90 )
16         return 19781;
17     }
18   }
19   else if ( VersionInformation.dwPlatformId == 2 )
20   {
21     switch ( VersionInformation.dwMajorVersion )
22     {
23       case 3u:
24         return 20052;
25       case 5u:
26         if ( !VersionInformation.dwMinorVersion )
27           return 2000;
28         if ( VersionInformation.dwMinorVersion == 1 )
29           return 22608;
30         if ( VersionInformation.dwMinorVersion == 2 )
31           return 2003;
32         break;
33       case 6u:
34         return 22121;
35     }
36   }
37   return 0;
38 }
```

function_2

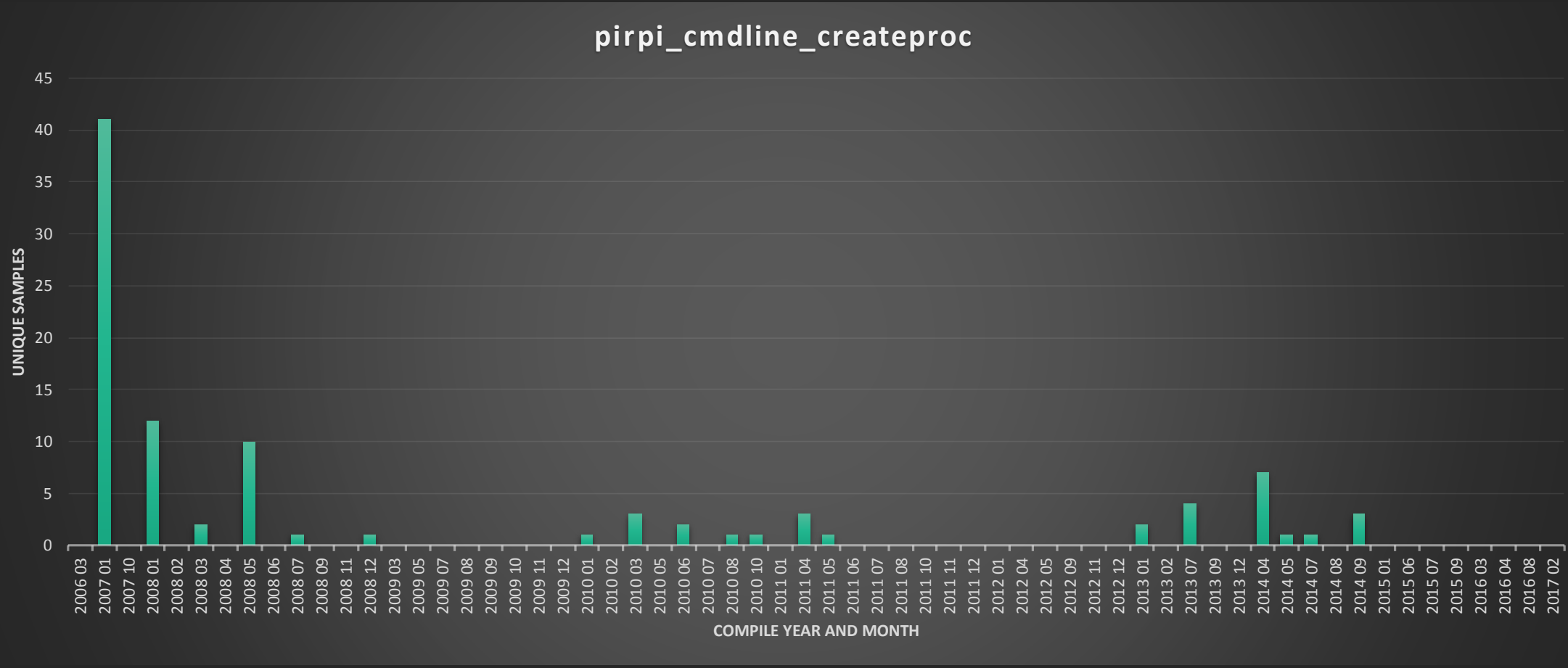

pirpi_version_check
UNIQUE SAMPLES
100
90
80
70
60
50
40
30
20
10
0
2006 03
2007 01
2007 10
2008 01
2008 02
2008 03
2008 04
2008 05
2008 06
2008 07
2008 09
2008 11
2008 12
2009 03
2009 05
2009 07
2009 08
2009 09
2009 10
2009 11
2009 12
2010 01
2010 02
2010 03
2010 05
2010 06
2010 07
2010 08
2010 10
2011 01
2011 04
2011 05
2011 06
2011 07
2011 08
2011 10
2011 11
2011 12
2012 01
2012 04
2012 05
2012 09
2012 11
2012 12
2013 01
2013 02
2013 07
2013 09
2013 12
2014 04
2014 05
2014 07
2014 08
2014 09
2015 01
2015 06
2015 07
2015 09
2016 03
2016 04
2016 08
2017 02
COMPILE YEAR AND MONTH

function_3

```
00401160
00401160
00401160
00401160                        ; int __cdecl sub_401160(LPSTR lpCommandLine, char)
00401160                        sub_401160 proc near
00401160
00401160                        ProcessInformation= _PROCESS_INFORMATION ptr -54h
00401160                        StartupInfo= _STARTUPINFOA ptr -44h
00401160                        lpCommandLine= dword ptr  4
00401160                        arg_4= byte ptr  8
00401160
00401160 83 EC 54               sub     esp, 54h
00401163 57                     push    edi
00401164 B9 11 00 00 00         mov     ecx, 11h
00401169 33 C0                  xor     eax, eax
0040116B 8D 7C 24 14            lea     edi, [esp+58h+StartupInfo]
0040116F F3 AB                  rep stosd
00401171 89 44 24 04            mov     [esp+58h+ProcessInformation.hProcess], eax
00401175 C7 44 24 14 44 00 00 00 mov    [esp+58h+StartupInfo.cb], 44h
0040117D 89 44 24 08            mov     [esp+58h+ProcessInformation.hThread], eax
00401181 5F                     pop     edi
00401182 89 44 24 08            mov     [esp+54h+ProcessInformation.dwProcessId], eax
00401186 89 44 24 0C            mov     [esp+54h+ProcessInformation.dwThreadId], eax
0040118A 8A 44 24 5C            mov     al, [esp+54h+arg_4]
0040118E 84 C0                  test    al, al
00401190 74 0F                  jz      short loc_4011A1

00401192 66 C7 44 24 40 00 00    mov     [esp+54h+StartupInfo.wShowWindow], 0
00401199 C7 44 24 3C 01 00 00 00 mov     [esp+54h+StartupInfo.dwFlags], 1

004011A1
004011A1                        loc_4011A1:
004011A1 8D 4C 24 00            lea     ecx, [esp+54h+ProcessInformation]
004011A5 8B 44 24 58            mov     eax, [esp+54h+lpCommandLine]
004011A9 8D 54 24 10            lea     edx, [esp+54h+StartupInfo]
004011AD 51                     push    ecx             ; lpProcessInformation
004011AE 52                     push    edx             ; lpStartupInfo
004011AF 6A 00                  push    0               ; lpCurrentDirectory
004011B1 6A 00                  push    0               ; lpEnvironment
004011B3 6A 00                  push    0               ; dwCreationFlags
004011B5 6A 01                  push    1               ; bInheritHandles
004011B7 6A 00                  push    0               ; lpThreadAttributes
004011B9 6A 00                  push    0               ; lpProcessAttributes
004011BB 50                     push    eax             ; lpCommandLine
004011BC 6A 00                  push    0               ; lpApplicationName
004011BE FF 15 30 20 40 00      call    ds:CreateProcessA
004011C4 8B 4C 24 00            mov     ecx, [esp+54h+ProcessInformation.hProcess]
004011C8 F7 D8                  neg     eax
004011CA 1B C0                  sbb     eax, eax
004011CC 23 C1                  and     eax, ecx
004011CE 83 C4 54               add     esp, 54h
004011D1 C3                     retn
004011D1                        sub_401160 endp
004011D1
```

```
1  HANDLE __cdecl sub_401160(LPSTR lpCommandLine, char a2)
2  {
3    BOOL v2; // eax@3
4    struct _PROCESS_INFORMATION ProcessInformation; // [esp+0h] [ebp-54h]@1
5    struct _STARTUPINFOA StartupInfo; // [esp+10h] [ebp-44h]@1
6
7    memset(&StartupInfo, 0, sizeof(StartupInfo));
8    ProcessInformation.hProcess = 0;
9    StartupInfo.cb = 68;
10   ProcessInformation.hThread = 0;
11   ProcessInformation.dwProcessId = 0;
12   ProcessInformation.dwThreadId = 0;
13   if ( a2 )
14   {
15     StartupInfo.wShowWindow = 0;
16     StartupInfo.dwFlags = 1;
17   }
18   v2 = CreateProcessA(0, lpCommandLine, 0, 0, 1, 0, 0, 0, &StartupInfo, &ProcessInformation);
19   return v2 != 0 ? ProcessInformation.hProcess : 0;
20 }
```

function_3


pirpi_cmdline_createproc
UNIQUE SAMPLES
0
5
10
15
20
25
30
35
40
45
2006 03
2007 01
2007 10
2008 01
2008 02
2008 03
2008 04
2008 05
2008 06
2008 07
2008 09
2008 11
2008 12
2009 03
2009 05
2009 07
2009 08
2009 09
2009 10
2009 11
2009 12
2010 01
2010 02
2010 03
2010 05
2010 06
2010 07
2010 08
2010 10
2011 01
2011 04
2011 05
2011 06
2011 07
2011 08
2011 10
2011 11
2011 12
2012 01
2012 04
2012 05
2012 09
2012 11
2012 12
2013 01
2013 02
2013 07
2013 09
2013 12
2014 04
2014 05
2014 07
2014 08
2014 09
2015 01
2015 06
2015 07
2015 09
2016 03
2016 04
2016 08
2017 02
COMPILE YEAR AND MONTH

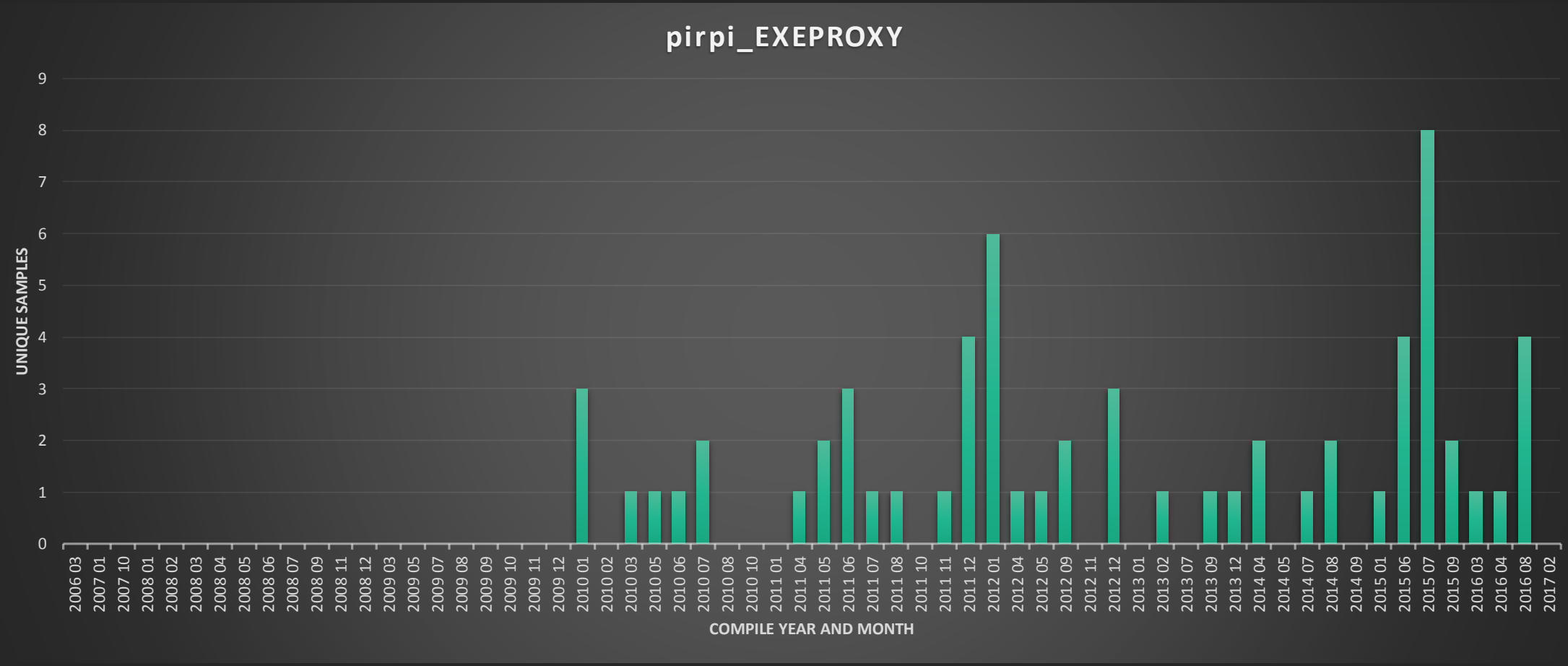
EXEProxy

```
2812816 || ETPRO TROJAN EXEPROXY CnC Beacon (INBOUND) 1 || md5,4d3874480110ba537b3839cb8b416b50
2812817 || ETPRO TROJAN EXEPROXY CnC Beacon (INBOUND) 2 || md5,4d3874480110ba537b3839cb8b416b50
2812818 || ETPRO TROJAN EXEPROXY CnC Beacon (INBOUND) 3 || md5,4d3874480110ba537b3839cb8b416b50
2812819 || ETPRO TROJAN EXEPROXY CnC Beacon (INBOUND) 4 || md5,b94bcffcacc65d05e5f508c5bd9c950c
2812820 || ETPRO TROJAN EXEPROXY CnC Beacon (INBOUND) 5 || md5,b94bcffcacc65d05e5f508c5bd9c950c
2812821 || ETPRO TROJAN EXEPROXY CnC Beacon (INBOUND) 6 || md5,b94bcffcacc65d05e5f508c5bd9c950c
```

- b94bcffcacc65d05e5f508c5bd9c950c
- Contains function_1
- Contains Anti-Disasm
- OpenSSL included
- Requires cmd line params to run

```
String

ipconfig /all

net localgroup administrators

net localgroup administrators /domain

net group "domain admins" /domain
netstat -an -p tcp
ping -a [redacted] -n 2
nbtstat -a [redacted]
netstat -an -p tcp
://[redacted]96/img/[redacted]01.exe
dir c:\
c:\aos.exe -in N111

net start aos
net stop aos
netstat -ano -p tcp
tasklist /v


://[redacted]96/img/[redacted]02.exe
dir c:\
c:\dh.exe
del c:\dh.exe
```

EXEProxy: pirpi_EXEPROXY



EXEProxy 2: Electric Boogaloo

- 4d3874480110ba537b3839cb8b416b50
- Contains function_1
- Server tool
- Requires Cmd Line Params
- No other notable anchor functions

2812549 || ETPRO TROJAN Possible EXEPROXY SSL Cert
2812788 || ETPRO TROJAN EXEPROXY DNS Lookup || md5,b94bcffcacc65d05e5f508c5bd9c950c
2812812 || ETPRO TROJAN EXEPROXY Possible HTTP CnC Beacon 1 || md5,b94bcffcacc65d05e5f508c5bd9c950c
2812813 || ETPRO TROJAN EXEPROXY Possible HTTP CnC Beacon 2 || md5,b94bcffcacc65d05e5f508c5bd9c950c
2812814 || ETPRO TROJAN EXEPROXY Possible HTTP CnC Beacon 3 || md5,b94bcffcacc65d05e5f508c5bd9c950c
2812815 || ETPRO TROJAN EXEPROXY Possible HTTP CnC Beacon 4 || md5,b94bcffcacc65d05e5f508c5bd9c950c
2812816 || ETPRO TROJAN EXEPROXY CnC Beacon (INBOUND) 1 || md5,4d3874480110ba537b3839cb8b416b50
2812817 || ETPRO TROJAN EXEPROXY CnC Beacon (INBOUND) 2 || md5,4d3874480110ba537b3839cb8b416b50
2812818 || ETPRO TROJAN EXEPROXY CnC Beacon (INBOUND) 3 || md5,4d3874480110ba537b3839cb8b416b50
2812819 || ETPRO TROJAN EXEPROXY CnC Beacon (INBOUND) 4 || md5,b94bcffcacc65d05e5f508c5bd9c950c
2812820 || ETPRO TROJAN EXEPROXY CnC Beacon (INBOUND) 5 || md5,b94bcffcacc65d05e5f508c5bd9c950c
2812821 || ETPRO TROJAN EXEPROXY CnC Beacon (INBOUND) 6 || md5,b94bcffcacc65d05e5f508c5bd9c950c
2812841 || ETPRO TROJAN EXEPROXY GET Session CnC Beacon || md5,4d3874480110ba537b3839cb8b416b50

```
">Ltime %2d:%2d:%2d Disconnect >"
"%2d:%2d:%2d >Cback:"
">LTime: %2d:%2d:%2d"
">Cback: %s:%d"
"H:%s "
"Ok. "
"K:%d"
```

EXEProxy 2: Electric Boogaloo

- a85f9b4c33061ee724e59291242b9e86
- OpenSSL Server
- Contains Public/Private Keys

```
00402EB7 2B C2                sub     eax, edx
00402EB9 8B F0                mov     esi, eax
00402EBB 68 00 0C 00 00       push    0C00h           ; Size
00402EC0 8D 84 24 88 00 00 00 lea     eax, [esp+188Ch+Dst]
00402EC7 6A 00                push    0               ; Val
00402EC9 50                   push    eax             ; Dst
00402ECA 89 74 24 28          mov     [esp+1894h+var_186C], esi
00402ECE E8 C3 2D 00 00       call    memset
00402ED3 68 00 0C 00 00       push    0C00h           ; Size
00402ED8 8D 8C 24 94 0C 00 00 lea     ecx, [esp+1898h+Buffer]
00402EDF 6A 00                push    0               ; Val
00402EE1 51                   push    ecx             ; Dst
00402EE2 E8 AF 2D 00 00       call    memset
00402EE7 57                   push    edi             ; Size
00402EE8 8D 94 24 A0 00 00 00 lea     edx, [esp+18A4h+Dst]
00402EEF 68 40 F5 45 00       push    offset aBeginCertifica ; "-----BEGIN CERTIFICATE-----\r\nMIIDvjCC"...
00402EF4 52                   push    edx             ; Dst
00402EF5 E8 96 2D 00 00       call    memcpy
00402EFA 56                   push    esi             ; Size
00402EFB 8D 84 24 AC 0C 00 00 lea     eax, [esp+18B0h+Buffer]
00402F02 68 A8 FA 45 00       push    offset aBeginRsaPrivat ; "-----BEGIN RSA PRIVATE KEY-----\r\nMIIE"...
00402F07 50                   push    eax             ; Dst
00402F08 E8 83 2D 00 00       call    memcpy
00402F0D 83 C4 30             add     esp, 30h
00402F10 85 ED                test    ebp, ebp
00402F12 0F 84 DB 01 00 00    jz      loc_4030F3
```

```
Certificate Information:
Common Name: marryks
Organization: www.bhpbilliton.com
Organization Unit: BHP BILLITON
Locality: Melbourne
State: Melbourne
Country: AU
Valid From: December 25, 2013
Valid To: December 23, 2023
Issuer: marryks, www.bhpbilliton.com
Serial Number: 0 (0x0)
```

In Summary:

- Some families never change
- Anchor Functions are Fun!
- Use public info
- Pirpi malware active since at least 2006
- Unintended findings

- Thanks:
  - Richard Wartell – Palo Alto Networks
  - Mike Scott – Palo Alto Networks

- Biblio:
  - MITRE:
    - https://attack.mitre.org/wiki/Software/S0063
  - FireEye:
    - https://www.fireeye.com/blog/threat-research/2010/11/ie-0-day-hupigon-joins-the-party.html
    - https://www.fireeye.com/blog/threat-research/2014/06/clandestine-fox-part-deux.html
    - https://www.fireeye.com/blog/threat-research/2014/11/operation_doubletap.html
    - https://www.fireeye.com/blog/threat-research/2015/06/operation-clandestine-wolf-adobe-flash-zero-day.html
  - Symantec:
    - https://www.symantec.com/connect/blogs/new-ie-zero-day-used-targeted-attacks
    - https://www.symantec.com/connect/blogs/buckeye-cyberespionage-group-shifts-gaze-us-hong-kong
    - http://www.symantec.com/content/en/us/enterprise/media/security_response/docs/Symantec-Buckeye-IOCs.txt
  - EmergingThreats:
    - https://rules.emergingthreats.net/changelogs/suricata-1.3.etpro.2015-09-10T21:29:38.txt
    - (cached by other sites, search for hash 4d3874480110ba537b3839cb8b416b50 and EXEProxy)
  - Palo Alto Networks:
    - https://researchcenter.paloaltonetworks.com/2015/07/apt-group-ups-targets-us-government-with-hacking-team-flash-exploit/
    - https://researchcenter.paloaltonetworks.com/2015/07/ups-observations-on-cve-2015-3113-prior-zero-days-and-the-pirpi-payload/