Windows Fenster-Kommunikation

Ein Vortrag von Volker Birk, dingens@bumens.org
Chaos Computer Club ERFA Kreis Ulm
http://www.ulm.ccc.de, http://www.ccc.de

Um was geht es?

- „Personal Firewalls“ und Virensucher sollen die Rechner „sicher“ machen.
- Sandboxing von Prozessen ohne VM?
- Windows ist *extrem* kommunikativ.
- IPC mal ganz anders…

Kommunikation: ein Überblick

- IPC im Push-Verfahren ohne Sicherheitssystem
- Messages, Hooks, Callbacks
- Windows 2.0: DDE
- Windows 3.x: OLE
- Visual Basic 1.0: VBX
- COM/DCOM/COM+, ActiveX: COM

Kommunikation: ein Überblick

- IPC im Push-Verfahren ohne Sicherheitssystem
- Messages, Hooks, Callbacks
- Windows 2.0: DDE
- Windows 3.x: OLE
- Visual Basic 1.0: VBX
- COM/DCOM/COM+, ActiveX: COM

Wie funktioniert denn die Windoze-GUI?

- Windoze ist ein Timesharing-System
  - Hardwaretreiber im Kernel, meist interrupt-gesteuert
  - Prozesse und Threads im Userland
- Windoze ist eine nachrichtenbasierende GUI
  - System Message Queue -> System Dispatcher
  - -> Thread Message Queue -> Thread Dispatcher
  - -> WindowProc für je eine Fensterklasse

Wie funktioniert denn die Windoze-GUI?

"Click!"

CPU

Driver

IRQ12

System-Message-Q

System Dispatcher

Thread-Message-Q

Translate

WM_NCHITTEST

Thread Dispatche

WindowProc
(Message Handler)

CCC ERFA Kreis Ulm, Volker Birk dingens@bumens.org

hello, world

```
int WinMain(HINSTANCE hInstance,
            HINSTANCE hPrevInstance,
            LPSTR     lpCmdLine,
            int       nCmdShow) {
    MSG msg;

    if (!hPrevInstance) InitApp(hInstance);
    InitInstance(hInstance, nCmdShow);

    while (GetMessage(&msg, NULL, 0, 0)) {
        TranslateMessage(&msg);
        DispatchMessage(&msg);
    }

    return msg.wParam;
}
```

Thread Dispatcher

hello, world

```
ATOM InitApp(HINSTANCE hInstance) {
    WNDCLASSEX wcex;
    memset(&wcex, 0, sizeof(WNDCLASSEX));

    wcex.cbSize = sizeof(WNDCLASSEX);

    wcex.style = CS_HREDRAW | CS_VREDRAW;
    wcex.lpfnWndProc = (WNDPROC) WndProc;
    wcex.hInstance = hInstance;
    wcex.hIcon = LoadIcon(NULL, IDI_APPLICATION);
    wcex.hCursor = LoadCursor(NULL, IDC_ARROW);
    wcex.hbrBackground = (HBRUSH)(COLOR_WINDOW+1);
    wcex.lpszClassName = "HelloWorldClass";

    return RegisterClassEx(&wcex);
}
```

Message Handler

hello, world

```
LRESULT CALLBACK WndProc(HWND hWnd, UINT message,
    WPARAM wParam, LPARAM lParam) {
    PAINTSTRUCT ps;
    HDC hdc;

    switch (message) {
    case WM_PAINT:
        hdc = BeginPaint(hWnd, &ps);
        RECT rt;
        GetClientRect(hWnd, &rt);
        DrawText(hdc, "hello, world", 12, &rt,
            DT_CENTER);
        EndPaint(hWnd, &ps);
        break;
    case WM_CLICK:
            ...
}
```

Ein Angriffspunkt: Hooks.

- Message Hooks lassen sich von beliebigen Applikationen aus vor beliebige Dispatcher installieren.
- Nachrichten gehen so gefiltert oder geändert zu den Message Handlern.
- Sicherheitssystem? Fehlanzeige.

Ein Angriffspunkt: Messages.

- Jedes Fenster verarbeitet Nachrichten.
- Kein Sicherheitssystem.
- Default-Implementierungen sind extrem gesprächig, z.B.:
  - WM_GETTEXT, WM_SETTEXT
  - WM_LBUTTONDOWN, WM_LBUTTONUP
  - WM_CLICK, WM_DCLICK
  - WM_COMMAND
  - Shatter Attacks, z.B. WM_TIMER

Shatter Attack: Wünsch Dir was!

- The **WM_TIMER** message is posted to the installing thread's message queue when a timer expires.

  Parameters
  wParam
  [in] Specifies the timer identifier.
  lParam
  [in] Pointer to an application-defined callback function that was passed to the SetTimer function when the timer was installed.

- WM_SETTEXT und Dein Code steht im Prozess.
- DefWindowProc ist so freundlich und wertet WM_TIMER aus.

Kommunikation: ein Überblick

- IPC im Push-Verfahren ohne Sicherheitssystem
- Messages, Hooks, Callbacks
- Windows 2.0: DDE
- Windows 3.x: OLE
- Visual Basic 1.0: VBX
- COM/DCOM/COM+, ActiveX: COM

DDE: IPC mit Windows Messages

- Oft vergessen, aber noch da: DDE.
- Cold & Hot Links
- Zwischenablageprotokoll zum Initialisieren
- DDEPoke/DDESend
- DDEExecute führt Befehle aus und startet Makros
- Office, Corel etc. unterstützen die volle Bandbreite
- Immer noch verwendet: Open, New, Print verbs und Hinzufügen von Symbolen in PROGMAN.EXE (heute: EXPLORER.EXE)
- Mit NetDDE über RPC im Netz.

Kommunikation: ein Überblick

- IPC im Push-Verfahren ohne Sicherheitssystem
- Messages, Hooks, Callbacks
- Windows 2.0: DDE
- Windows 3.x: OLE
- Visual Basic 1.0: VBX
- COM/DCOM/COM+, ActiveX: COM

Object Linking and Embedding

- Linked Objects.
- Embedded Objects.
- OLE2: inplace editing.
- Klassiker: Einfügen Objekt, CMD.EXE.
- OLE Automation.

Kommunikation: ein Überblick

- IPC im Push-Verfahren ohne Sicherheitssystem
- Messages, Hooks, Callbacks
- Windows 2.0: DDE
- Windows 3.x: OLE
- Visual Basic 1.0: VBX
- COM/DCOM/COM+, ActiveX: COM

Visual Basic

- RAD über Component Software
- Applikationen visuell zusammenklicken
- Architektur? Flickenteppich.
- VBX: Visual Basic Klassen in DLLs
- Nachfolger: OCX, ActiveX

Kommunikation: ein Überblick

- IPC im Push-Verfahren ohne nennenswertes Sicherheitssystem
- Messages, Hooks, Callbacks
- Windows 2.0: DDE
- Windows 3.x: OLE
- Visual Basic 1.0: VBX
- COM/DCOM/COM+, ActiveX: COM

Component Object Model

- COM Komponenten in Prozessen.
- COM Komponenten als OCX.
- OLE Automation
- Unterschiede zwischen OLE Document Server und COM Komponente verschwimmen, die meisten sind beides
- ActiveX: wir wollen kein Java, haben aber nichts passendes, also OCX.

Und darauf aufbauend: Middleware.

- Manche Middleware hält sich nicht dran:
- z.B.: Ich will mit ISDN ins Internet
  - CAPIPort: ISDN / CAPI / TAPI / NDIS
  - NDISWAN: ISDN / CAPI / WAN-Adapter / NDIS

Und darauf aufbauend: Middleware (z.B. ADO).

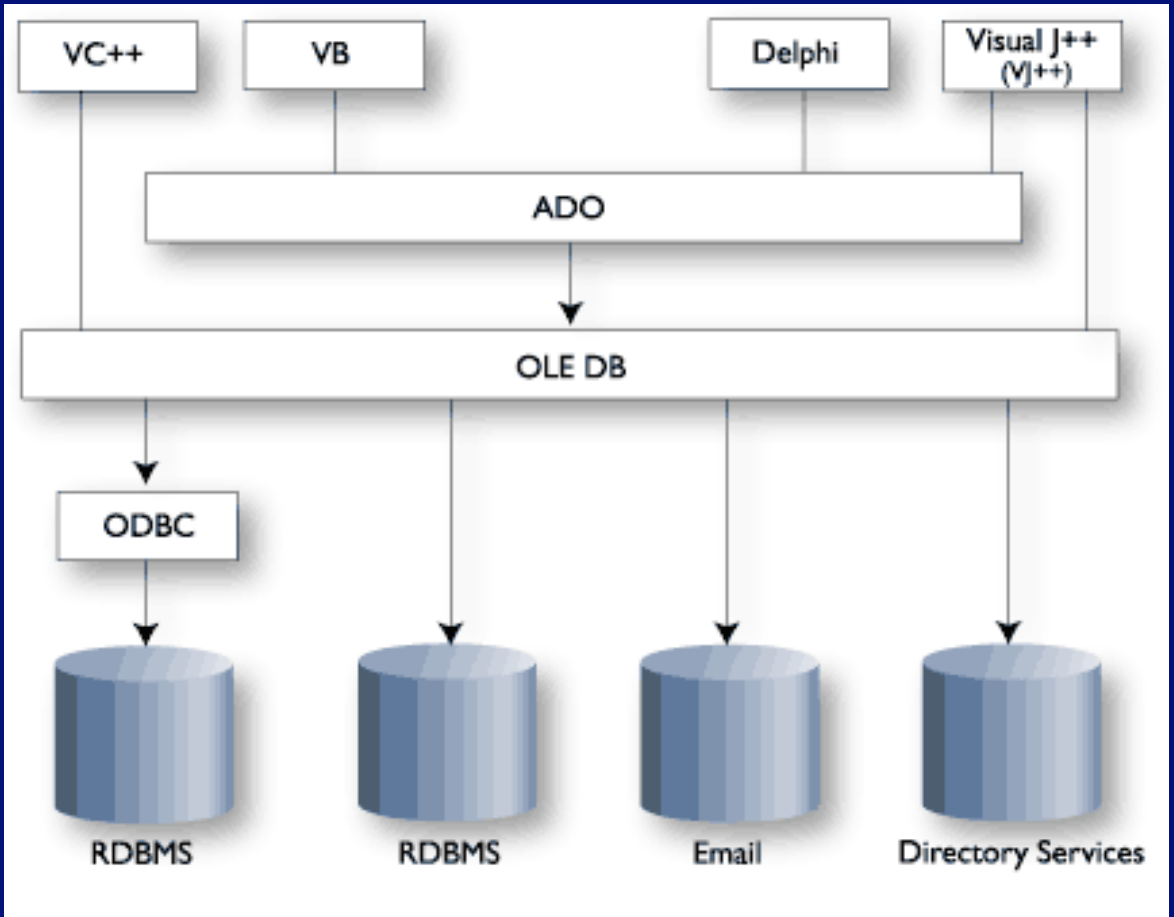

FIGURE 6: Various routes an application can take in ADO

ADO: simpel und effektiv

```
Set Cnxn = CreateObject("ADODB.Connection")
Cnxn.Open strCnxn
Set rs = CreateObject("ADODB.Recordset")
rs.ActiveConnection = Cnxn
rs.Source = strSQL : rs.Open

Do While Not rs.EOF
        tmp = trim(rs(0) & "")
        for i=1  to rs.Fields.Count - 1
                tmp = tmp & strSeparator & trim(rs(i) & "")
        next
        WScript.StdOut.WriteLine tmp
        rs.MoveNext
Loop
```

Windows Explorer: Dein Freund und Helfer.

- Explorer Extensions rechnen im Prozess Explorer.EXE
- Aktivierung durch Benennung eines Verzeichnisses.
- Aktivierung durch DESKTOP.INI
- 1001 Möglichkeit in der Registry
- Ausführen von Code und keiner merkt‘s.
- Gängige Viren nutzen keine 5% der Möglichkeiten.

Undundund…

- Windows platzt vor Middleware.
- Die Schwierigkeit ist eher rauszufinden, wo es was fertig gibt.
- Das System ist unüberschaubar komplex.
- .NET kommt da jetzt DRAUF.

Resümee

- Windows wird nicht sicher zu kriegen sein.
- User Centric Design resultierte in unglaublicher Komplexität.
- KISS.
- Microsoft, quo vadis?

Chaos Computer Club.

Kabelsalat ist gesund.

Vielen Dank für Eure Aufmerksamkeit!

Volker Birk, CCC ERFA Kreis Ulm

mailto:dingens@bumens.org

http://www.ulm.ccc.de

http://www.ccc.de