

Advanced Excel Hacking Workshop

Didier Stevens

<http://.DidierStevens.com/excel.zip>

No Exploits

Just Features

Unzip excel.zip to **c:\excel**

Password: Workshop

VBA (Visual Basic for Applications)

is a complete Windows programming language

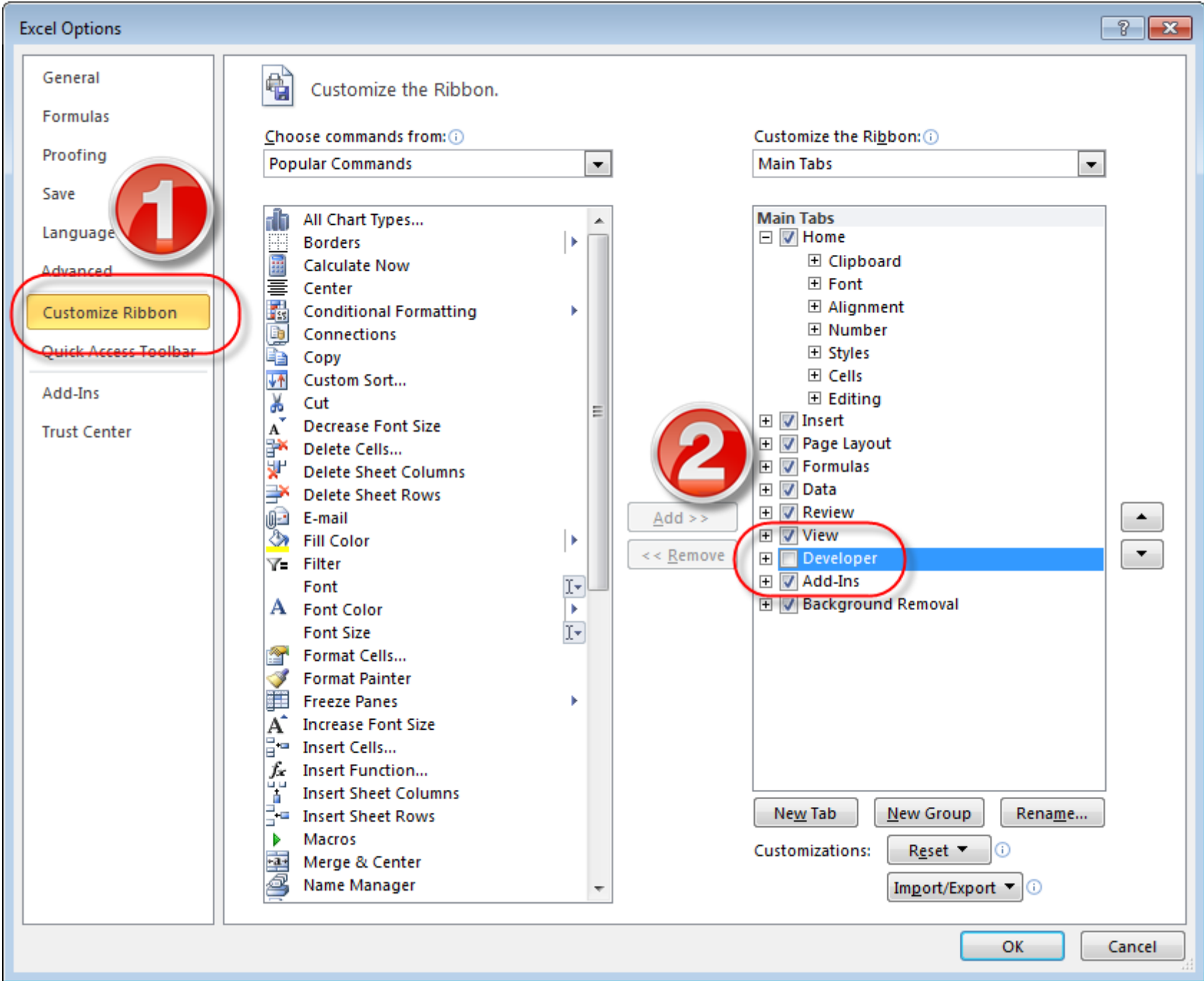
VBS (Visual Basic Script)

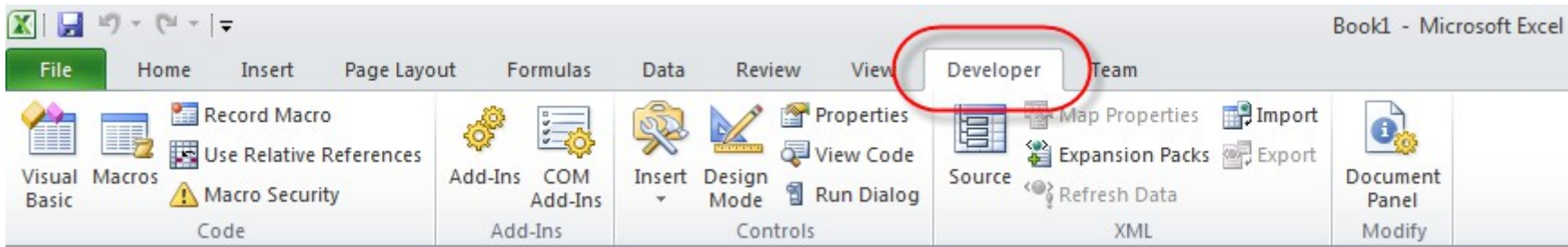
is NOT a complete Windows programming
language

VBA has access to the Windows API

VBA: MS Office (Word, Excel Powerpoint, ...),
AutoCAD, ...

Excel: what I prefer as a User Interface





Exercise 1:

“Hello World” message box with VBA

VBA7

Introduced with Office 2010

Support for 64-bit

32-bit Excel or 64-bit Excel?

Excel 2007 or earlier: 32-bit

Excel 2010 or 2013:

Check File/Help

- Save
- Save As
- Open
- Close
- Info
- Recent
- New
- Print
- Save & Send
- Help**
- Options
- Exit

Support



Microsoft Office Help
Get help using Microsoft Office.



Getting Started
See what's new and find resources to help you learn the basics quickly.



Contact Us
Let us know if you need help or how we can make Office better.

Tools for Working With Office



Options
Customize language, display, and other program settings.



Check for Updates
Get the latest updates available for Microsoft Office.



Product Activated

Microsoft Office Professional Plus 2010
This product contains Microsoft Access, Microsoft Excel, Microsoft OneNote, Microsoft Outlook, Microsoft PowerPoint, Microsoft InfoPath.

About Microsoft Excel

Version: 14.0.7106.5001 (64-bit)
[Additional Version and Copyright Information](#)
Part of Microsoft Office Professional Plus 2010
© 2010 Microsoft Corporation. All rights reserved.
[Microsoft Customer Services and Support](#)
Product ID: 02257-211-1490356-49815
[Microsoft Software License Terms](#)

3 new VBA7 keywords:

PtrSafe
LongLong
LongPtr

2 new VBA7 compilation constants

VBA7
Win64

I use Win64

If Win64 is defined, I know that I'm using VBA7 on
a 64-bit application

Thus I use the new keywords
(PtrSafe, LongLong, LongPtr)

If Win64 is not defined, I know that I am on 32-bit application.

And then I DO NOT use the new keywords.

Exercise 2:

“Hello World” message box with API

32-bit, 64-bit & both

API functions:

not only basic types as arguments,

but also structures

```
Private Declare PtrSafe Sub GetSystemTime Lib  
"kernel32.dll" (st As SYSTEMTIME)
```

Private Type SYSTEMTIME

wYear As Integer

wMonth As Integer

wDayOfWeek As Integer

wDay As Integer

wHour As Integer

wMinute As Integer

wSecond As Integer

wMilliseconds As Integer

End Type

Exercise 3:

GetSystemTime

32-bit, 64-bit & both

InstalledPrograms

NetworkMashup-32

TaskManager.xls / TaskManagerSC.xls

Problem: writing a lot of VBA code

Datapipe

Modify C source code datapipe

datapipe.exe → datapipe.dll

```
int APIENTRY WinMain(HINSTANCE hInstance, HINSE
{
    RedirectIOToConsole();
    puts("Datapipe");
    puts("-----");
    puts("https://DidierStevens.com");
    puts("");
    puts("IPv4 addresses of this machine:");
    PrintIPs();
    puts("");
    DataPipe();
    puts("Press return to end");
    getchar();
}
```

```
BOOL WINAPI DllMain(HINSTANCE hiDLL, DWORD dwRea
{
    switch (dwReason)
    {
        case DLL_PROCESS_ATTACH:
            RedirectIOToConsole();
            puts("Datapipe");
            puts("-----");
            puts("https://DidierStevens.com");
            puts("");
            puts("IPv4 addresses of this machine:");
            PrintIPs();
            puts("");
            DataPipe();
            return FALSE;
    }
}
```

DLL to shellcode

CreateMemoryModuleShellCode.py datapipe-
dll.dll datapipe-dll.dll.bin

Shellcode to VBA

shellcode2vba.py datapipe-dll.dll.bin datapipe-
dll.dll.bin.base64.vba

ReactOS cmd and regedit

```
/*
 * main function
 */
int cmd_main (int argc, const TCHAR *argv[])
{
    HANDLE hConsole;
    TCHAR startPath[MAX_PATH];
    CONSOLE_SCREEN_BUFFER_INFO Info;

    lpOriginalEnvironment = DuplicateEnvironment();

    GetCurrentDirectory(MAX_PATH, startPath);
    _tchdir(startPath);

    SetFileApisToOEM();
    InputCodePage = 0;
    OutputCodePage = 0;

    AllocConsole();

    hConsole = CreateFile(_T("CONOUT$"), GENERIC_READ|GENERIC_WRITE,
        FILE_SHARE_READ|FILE_SHARE_WRITE, NULL,
        OPEN_EXISTING, 0, NULL);
```

```
#include <precomp.h>
```

```
INT WINAPI
```

```
DllMain(
```

```
    IN PVOID hInstanceDll,
```

```
    IN ULONG dwReason,
```

```
    IN PVOID reserved)
```

```
{
```

```
    switch (dwReason)
```

```
    {
```

```
        case DLL_PROCESS_ATTACH:
```

```
            cmd_main(0, NULL);
```

```
            break;
```

```
        case DLL_THREAD_ATTACH:
```

```
            break;
```

```
        case DLL_THREAD_DETACH:
```

```
            break;
```

```
        case DLL_PROCESS_DETACH:
```

```
            break;
```

```
    }
```

```
    return TRUE;
```

```
}
```

```
<group>
```

```
<module name="cmd" type="win32dll" installbase="system32" installname="cmd.dll"
```

```
<include base="ReactOS">include/reactos/wine</include>
```

Putty

Didier Stevens Labs

FOUNDED BY DIDIER STEVENS



20% discount sale for Brucon:

PDF Analysis workshop videos on CD: €20

White Hat Shellcode workshop videos on CD: €20

x64 workshop videos on CD: €20

All videos on CD: €50

<http://DidierStevensLabs.com>