

Python arsenal for RE

[v. 0x01]



Dmitriy "D1g1" Evdokimov

DSecRG

Email: d.evdokimov@dsecrg.com

Twitter: [@evdokimovds](https://twitter.com/evdokimovds)

Content

Introduction.....	4
Considered projects	6
bochs-python-instrumentation	7
Buggery.....	8
Ctypes.....	9
dislib	10
diStorm.....	11
IDAPython	12
ImmLIB.....	14
libdisassemble	15
lldb.....	16
Miasm.....	17
OllyPython.....	18
pefile.....	19
PIDA	20
ProcessTap.....	21
pyasm.....	22
PyBox.....	23
PyCodin.....	24
pydasm.....	25
Pydb	26
PyDBG.....	27
PyDbgEng.....	28
pydbgr	29
pydot.....	30
PyEA.....	31
PyEMU.....	32
pyew.....	33
pygdb.....	34
pyHIEW.....	35

pykd.....	36
Pylibemu	37
pyMem	38
pyREtic.....	39
PySTP.....	40
PythonGdb.....	41
python-haystack	42
python-ptrace	43
pytracer	44
radapy	45
uhooker.....	46
Vivisect.....	47
vtrace.....	48
WinAppDbg.....	49
Z3-python	50
Note.....	51
About Author.....	52
About ERPScan.....	53
About DSecRG — Research center of ERPScan.....	54
Our Contacts.....	55

Introduction

PRAEMONITUS PRAEMUNITUS

This whitepaper is a collection of various python engines, extensions, libraries, shells, that aids in the job code for understanding, analyzing and sometimes breaking.

Quite ordinary, but the Python programming language has become a language of hackers. And it is not surprising, because it has all the necessary qualities:

- Free
- Developer productivity
- Portable
- Powerful
 - Automatic memory management
 - Built-in object types
 - Built-in tools
 - Dynamic typing
 - Library utilities
 - Programming-in-the-large support
 - Third-party utilities
- OOP
- Mixable
- Easy to learn

A great role in this were played by such projects as IDA Pro, WinDBG, OllyDebug, gdb, which, being a de-facto standard among disassemblers and debuggers, eventually began to support the scripting engines in Python. Of course, they had maintained their own API for plug-in developing, and it was not a small number of them, but exactly with the appearance of the Python support they received a strong push in the development: increased the number of plug-in, increased community, and of course their flexibility also increased, which allowed them to interact both with each other and with other applications, using the best aspects of each other. But in the beginning of the path there was naturally only hacker spirit and idea.

But everything step by step went to this: with the increasing of technologies' complexity the software complexity is growing too, and specialists in information security need to keep pace with this development (and sometimes even be ahead). It is almost impossible to qualitatively examine the application for an adequate time by hand with a disassembler or a debugger. And automation can help in this situation (XXI century after all).

We live in a very rapidly developing world, in which it is very difficult to keep track of everything happening therefore it is very difficult to be always aware of all. Sometimes even in a specific area (in our case, in the field of reverse engineering) for an experienced specialist, not to mention the beginners, who make their first steps. So here I tried to collect and review the most interesting and useful Python projects for reverse engineering.

In my opinion today there is very few structured knowledge about hacking, reversing engineering, software exploitation techniques. If many of the older sciences are very well structured and well oriented in, in our field it is very difficult to make the first steps. By means of this whitepaper I will try to make a small step in the direction of awareness and systematization.

I hope that you will learn something new or remember the forgotten and possibly breathe new life into one of these projects, because some of them are unfortunately do not develop for quite a long time.

Here 43 python projects will be considered. And python tools for disassembling, debugging, visualization will be reviewed, without which today it is quite difficult and so on. Unfortunately, not all of the above projects are actively developed in the case of certain circumstances, and they were presented here, to show the original idea and bring them to the attention.

For description of each of the projects 11 characteristics were allocated:

Project	name of the engine, expansion, library, shell and etc
Author	author(s) of the project (many thanks to these guys)
Site project	site of the project, from which you can download it
Tags	a list of tags, which on my mind characterize the project more common
License	the type of license under which this project is spread
Python versions	a set of python versions with which this project compatible (may work and on other versions — if you know, please let me know)
Platforms	the list of platforms supported by the project
Processors (Architecture)	the list of processor architecture supported by the project
Base project	this is the name of the program for which it is intended (depends)
Description	short description of the project
Tools	here are the most famous and interesting tools which use this project
Useful links	references to the manuals, documentation or simply interesting blog entries concerning this project

If there is the "???" sign in the line, then this information is not known to me and I would be glad to get it.

This article is by no means exhaustive. If there is anything that I may have missed or have misstated, please email me at d.evdokimov@dsecrg.com and I will edit this post accordingly. I hope for your help in its correction, updating and improvement.

Considered projects

The list of considered projects:

1. bochs-python-instrumentation
2. Buggery
3. Ctypes
4. Dislib
5. diStorm
6. IDAPython
7. ImmLIB
8. libdisassemble
9. lldb
10. Miasm
11. OllyPython
12. Pefile
13. PIDA
14. ProcessTap
15. Pyasm
16. PyBox
17. PyCodin
18. Pydasm
19. Pydb
20. PyDBG
21. PyDbgEng
22. Pydbgr
23. Pydot
24. PyEA
25. PyEMU
26. Pyew
27. Pygdb
28. pyHIEW
29. Pykd
30. Pylibemu
31. pyMem
32. pyREtic
33. PySTP
34. PythonGdb
35. python-haystack
36. python-ptrace
37. pytracer
38. radapy
39. Uhooker
40. vivisect
41. vtrace
42. WinAppDbg
43. Z3-python

Let's start consideration of projects.

bochs-python-instrumentation

Project:	bochs-python-instrumentation
Author:	Ero Carrera (@erocarrera)
Site project:	https://github.com/zynamics/bochs-python-instrumentation
Tags:	debugger, emulator
License:	???
Python versions:	2.5
Platforms:	win/lin
Processors:	x86/x64
Base project:	Bochs (2.4.5 and 2.4.6)
Description:	This patch for Bochs provides a Python interpreter instead of Bochs' own debugger, yet still providing the debugger functionality. It also allows to interact with the instrumentation interface on-demand, by dynamically associating Python methods to handle instrumentation events.
Tools:	???
Useful links:	https://github.com/zynamics/bochs-python-instrumentation/wiki — wiki http://blog.zynamics.com/2010/07/16/recon-slides-packer-genetics-the-selfish-code-bochspython/ — presentation

Buggery

Project:	Buggery
Author:	Grugq (@thegrugq)
Site project:	https://github.com/grugq/Buggery
Tags:	scripting engine, debugger
License:	???
Python versions:	2.7
Platforms:	win
Processors:	x86/x64
Base project:	WinDbg
Description:	Python wrapper for DbgEng.
Tools:	SWFRETools (https://github.com/sporst/SWFREtools)
Useful links:	http://pastebin.com/HB4H2gPu — example

Ctypes

Project:	Ctypes
Author:	Thomas Heller
Site project:	http://sourceforge.net/projects/ctypes/ (In Python 2.5 it is already included)
Tags:	wrapper
License:	MIT License
Python versions:	more than 2.3
Platforms:	win/lin/mac
Processors:	x86/x64
Base project:	—
Description:	ctypes is a Python module allowing to create and manipulate C data types in Python. These can then be passed to C-functions loaded from dynamic link libraries.
Tools:	PyMem, WinAppDBG
Useful links:	http://docs.python.org/library/ctypes.html — official documentation http://www.mso.anu.edu.au/~tiago/talks_papers/Cython.pdf — presentation “Using Cython to optimize Python and interface with C” http://www.rohitab.com/discuss/topic/37018-api-hooking-in-python/ — “API Hooking in Python”

dislib

Project:	dislib
Author:	distorm, Gil Dabah (arkon@ragestorm.net)
Site project:	http://code.google.com/p/distorm/
Tags:	PE+ reader
License:	GNU GPL v3
Python versions:	2.5
Platforms:	win
Processors:	x86/x64
Base project:	—
Description:	A Fast Python Library for Reading PE+ Files.
Tools:	???
Useful links:	???

diStorm

Project:	diStorm
Author:	distorm, Gil Dabah (arkon@ragestorm.net)
Site project:	http://code.google.com/p/distorm/
Tags:	disassembler
License:	GNU GPL v3 and commercial license
Python versions:	2.x, 3.x
Platforms:	win/lin/mac
Processors:	x86/x64/PowerPC
Base project:	—
Description:	diStorm3 binary stream disassembler library project.
Tools:	???
Useful links:	???

IDAPython

Project:	IDAPython
Author:	Gergely Erdelyi (http://gergelyerdelyi.com/) Elias Bachaalany (@0xeb)
Site project:	http://code.google.com/p/idapython/
Tags:	scripting engine, disassemble, debugger
License:	New BSD License
Python versions:	2.4-2.7
Platforms:	win/mac
Processors:	x86
Base project:	IDA Pro (from 5.1)
Description:	IDAPython is an IDA Pro plugin that integrates the Python programming language, allowing scripts to run in IDA Pro. These programs have access to IDA Plugin API, IDC and all modules available for Python. The power of IDA Pro and Python provides a platform for easy prototyping of reverse engineering and other research tools.
Tools:	mynav — http://code.google.com/p/mynav/ Dr. Gadget — http://www.openrce.org/blog/view/1570/Dr. Gadget IDAPython plugin rtti-helper-scripts — https://github.com/zynamics/rtti-helper-scripts msdn-plugin-ida — https://github.com/zynamics/msdn-plugin-ida ida2sql-plugin-ida — https://github.com/zynamics/ida2sql-plugin-ida IDA file Patcher — http://code.google.com/p/reverse-engineering-scripts/ Idagrapher — https://code.google.com/p/idagrapher/

IDAPython

Useful links:

http://www.hex-rays.com/idapro/idapython_docs/ — official documentation

<http://gergelyerdelyi.com/publication/IDAPython.pdf> — IDAPython: User Scripting for a Complex Application

<http://defcon.org/images/defcon-18/dc-18-presentations/Pridgen-Wollenweber/DEFCON-18-Pridgen-Wollenweber-IDA-Bridge.pdf> — TOOLSMITHING AN IDA BRIDGE: A TOOL BUILDING CASE STUDY

<http://magiclantern.wikia.com/wiki/IDAPython> — blog entries about IDAPython

http://dvlabs.tippingpoint.com/pub/chotchkies/SeattleToorcon2008_RECookbook.pdf — “Reverse Engineer's Cookbook” presentation

http://www.openrce.org/articles/full_view/11 — “Introduction to IDAPython” from OpenRCE

ImmLIB

Project:	ImmLIB
Author:	Immunity, Inc.
Site project:	http://www.immunityinc.com/products-immdbg.shtml
Tags:	scripting engine, disassemble, debugger
License:	Immunity Debugger License
Python versions:	2.5 and 2.7.1
Platforms:	win
Processors:	x86
Base project:	ImmunityDebugger
Description:	Immunity Debugger's Python API includes many useful utilities and functions. Your scripts can be as integrated into the debugger as the native code. This means your code can create custom tables, graphs, and interfaces of all sorts that remain within the Immunity Debugger user experience.
Tools:	pvefindaddr — http://redmine.corelan.be:8800/projects/pvefindaddr mona — http://redmine.corelan.be/projects/mona
Useful links:	http://debugger.immunityinc.com/Documentation/ — official documentation http://beist.org/research/public/immunity1/imm_present_jff.pdf — presentation http://www.corelan.be/index.php/2010/01/26/starting-to-write-immunity-debugger-pycommands-my-cheatsheet/ — cheatsheet by Corelan https://forum.immunityinc.com/board/ — forum

libdisassemble

Project:	libdisassemble
Author:	Immunity Inc. , atlas (atlas@r4780y.com) Matthew Carpenter (mcarpenter@intelguardians.com)
Site project:	http://www.immunitysec.com/resources-freesoftware.shtml
Tags:	disassembler
License:	GNU GPL v2
Python versions:	2.5
Platforms:	win/lin
Processors:	x86
Base project:	—
Description:	Libdisassembly is simply a python library for disassembling x86 opcodes. It has been made for Immunity's PDB Project (a vulnerability development focused debugger), and is partially based on mammon libdisasm opcode list. There is still a lot of work to do with the Metadata, but the library tries to return as much information it can get off of an opcode.
Tools:	???
Useful links:	???

lldb

Project:	lldb
Author:	University of Illinois/NCSA
Site project:	http://lldb.llvm.org/
Tags:	scripting engine, debugger, disassembler
License:	University of Illinois/NCSA Open Source License
Python versions:	2.6
Platforms:	mac/lin
Processors:	x86/x64/ARM
Base project:	LLDB
Description:	lldb also has a built-in Python interpreter, which is accessible by the "script" command. All the functionality of the debugger is available as classes in the Python interpreter, so the more complex commands that in gdb you would introduce with the "define" command can be done by writing Python functions using the lldb-Python library, then loading the scripts into your running session and accessing them with the "script" command.
Tools:	Example — http://llvm.org/svn/llvm-project/lldb/trunk/examples/python/disasm.py
Useful links:	http://llvm.org/svn/llvm-project/lldb/trunk/test/python_api/ — API, http://llvm.org/devmtg/2010-11/Clayton-LLDB.pdf — “LLDB Modular Debugging Infrastructure” presentation

Miasm

Project:	Miasm
Author:	Serpilliere (serpilliere@droids-corp.org)
Site project:	http://code.google.com/p/miasm/ http://code.google.com/p/smiasm/
Tags:	framework, disassembler, emulator, intermediate language
License:	GNU GPL v2
Python versions:	2.5
Platforms:	win/lin
Processors:	x86/PowerPC/ARM
Base project:	—
Description:	Miasm is a free and open source reverse engineering framework. Miasm aims at analyzing/modifying/generating binary programs. Miasm embed its own disassembler, intermediate language and instruction semantic. To emulate code, it uses libtcc to jit C code generate from intermediate representation. It can emulate shellcodes, parts of binaries. Python callback can be executed to emulate library functions.
Tools:	???
Useful links:	http://miasm.googlecode.com/hg/doc/slides.pdf - Miasm (incomprehensible documentation)

OllyPython

Project:	OllyPython
Author:	Scott Knight (knightsc@gmail.com)
Site project:	http://code.google.com/p/ollypython/
Tags:	scripting engine, debugger
License:	New BSD License
Python versions:	2.4
Platforms:	win
Processors:	x86
Base project:	OllyDbg
Description:	OllyPython is an OllyDbg plugin that integrates the Python programming language, allowing scripts to run in OllyDbg.
Tools:	???
Useful links:	http://www.team509.com/modules.php?name=News&file=article&sid=48 — sample of use in entry blog

pefile

Project:	pefile
Author:	Ero Carrera (@erocarrera)
Site project:	http://code.google.com/p/pefile/
Tags:	PE+ reader
License:	MIT License
Python versions:	2.x
Platforms:	win/lin/mac
Processors:	x86/x64
Base project:	—
Description:	pefile is a multi-platform Python module to read and work with Portable Executable (aka PE) files. Most of the information in the PE Header is accessible, as well as all the sections, section's information and data. pefile requires some basic understanding of the layout of a PE file. Armed with it it's possible to explore nearly every single feature of the file.
Tools:	IDA PEiD — http://code.google.com/p/reverse-engineering-scripts/
Useful links:	http://code.google.com/p/pefile/wiki/UsageExamples — usage examples http://www.gerryeisenhaur.com/2011/01/04/using-python-and-pefile-to-extract-embedded-code/ — usage examples http://www.recon.cx/en/f/lightning-ecarrera-win32-static-analysis-in-python.pdf — “Win32 Static Analysis in Python” presentation https://www.blackhat.com/presentations/bh-usa-07/Carrera/Presentation/bh-usa-07-carrera.pdf — “4 x 5: Reverse Engineering Automation with Python” presentation

PIDA

Project:	PIDA
Author:	Pedram Amini (@pedramamini)
Site project:	http://code.google.com/p/paimei/ (part of PaiMei)
Tags:	visualization
License:	GNU GPL v2 or later
Python versions:	2.?
Platforms:	win/mac
Processors:	x86
Base project:	IDAPython, pGRAPH
Description:	Built on top of pGRAPH, PIDA aims to provide an abstract and persistent interface over binaries (DLLs and EXEs) with separate classes for representing functions, basic blocks and instructions. The end result is the creation of a portable file that when loaded allows you to arbitrarily navigate throughout the entire original binary.
Tools:	PaiMei (http://code.google.com/p/paimei/)
Useful links:	http://pedram.redhive.com/PyDbg/docs/ — official overview

ProcessTap

Project:	ProcessTap
Author:	Roberto Paleari (@rpaleari) Lorenzo Martignoni (@martignlo) Lorenzo Cavallaro (http://www.few.vu.nl/~sullivan/)
Site project:	http://code.google.com/p/processtap/
Tags:	scripting engine, DBI
License:	GNU GPL v3
Python versions:	2.5, 2.6
Platforms:	lin
Processors:	x86/x64
Base project:	PinTool, (Valgrind, QEMU, DinamoRIO)
Description:	ProcessTap is a dynamic tracing framework for analyzing closed source-applications. ProcessTap is inspired by DTrace and SystemTap, but it is specific for analyzing closed-source user-space applications. ProcessTap leverages dynamic binary instrumentation to intercept the events of interest (e.g., function calls, system call, memory accesses, and conditional control transfers). Although the current implementation relies on PinTool, alternative back-ends for instrumentation (e.g., Valgrind, Qemu, or DynamoRIO) can be used. The language used in ProcessTap for writing scripts to instrument applications is Python.
Tools:	???
Useful links:	http://code.google.com/p/processtap/source/browse/#svn%2Ftrunk%2Fexamples — examples

pyasm

Project:	pyasm
Author:	Grant Olson (kgo@grant-olson.net)
Site project:	http://www.grant-olson.net/python/pyasm
Tags:	dynamic assembler
License:	GNU AGPL v3
Python versions:	2.4 and 2.6
Platforms:	win/lin
Processors:	x86
Base project:	—
Description:	<p>Pyasm is a full-featured dynamic assembler written entirely in Python. By dynamic, I mean that it can be used to generate and execute machine code in python at runtime without requiring the generation of object files and linkage. It essentially allow 'inline' assembly in python modules on x86 platforms. Pyasm can also generate object files (for windows) like a traditional standalone assembler, although you're probably better off using one of the many freely available assemblers if this is you primary goal.</p>
Tools:	???
Useful links:	<p>http://codeflow.org/entries/2009/jul/31/pyasm-python-x86-assembler/ — example</p> <p>http://www.docstoc.com/docs/29701848/PyASM-Users-Guide-V-03 — PyASM User's Guide</p>

PyBox

Project:	PyBox
Author:	Felix Leder (felix.leder@googlemail.com) Daniel Plohmann (daniel.plohmann@googlemail.com)
Site project:	http://code.google.com/p/pyboxed/
Tags:	monitoring of processes, sandbox
License:	GNU GPL v3
Python versions:	2.6 or above
Platforms:	win
Processors:	x86
Base project:	—
Description:	PyBox (short for "Python Sandbox") is a flexible and light-weight process and system analysis framework. A user-level framework for rootkit-like monitoring of processes.
Tools:	???
Useful links:	https://eldorado.tu-dortmund.de/bitstream/2003/27336/1/BookOfAbstracts_Spring5_2010.pdf — "PyBox — A Python approach to sandboxing", http://code.google.com/p/pyboxed/wiki/WikiStart — wiki, http://www.troopers.de/wp-content/uploads/2011/04/TR11_Leder_What_is_happening_in_your.pdf — "Do you know what's happening in your <put app title here>?" presentation

PyCodin

Project:	PyCodin
Author:	Adrián Manrique (@n0km, adrian@coresecurity.com), Andrés López Luksenberg (aluksenberg@coresecurity.com)
Site project:	http://corelabs.coresecurity.com/index.php?module=Wiki&action=view&type=tool&name=PyCodin
Tags:	DBI
License:	GNU GPL v2
Python versions:	2.5
Platforms:	win
Processors:	x86/x64
Base project:	QEMU
Description:	PyCodin is an open source Python library that allows instrumentation of low-level code for different architectures. It came out from the necessity of developing a testing environment for low-level code that exploits vulnerabilities (a.k.a. shellcode). The library provides a virtual CPU front-end, allowing the manipulation of a virtualized memory space and creating different scenarios, giving the developer new tools to control the execution. PyCodin also allows runtime inspection and modification of the execution context of the instrumented program. The first version of the tool uses Qemu as the virtualization back-end.
Tools:	???
Useful links:	http://corelabs.coresecurity.com/index.php?module=Wiki&action=attachment&type=researcher&page=Adrian_Manrique&file=publication%2FPyCodin_-_Instrumentando_codigo_sin_dolor%2Fpycodin-ManriqueLuksenberg-PyconArgentina2010.pdf — “Pycodin: Instrumentando código sin dolor” presentation (spanish)

pydasm

Project:	pydasm
Author:	Ero Carrera (@erocarrera)
Site project:	http://dkbza.org/pydasm.html
Tags:	disassembler
License:	???
Python versions:	2.?
Platforms:	win/lin
Processors:	x86
Base project:	libdasm
Description:	pydasm is a python wrapper for libdasm. It attempts to capture all the functionality of libdasm and bring its versatility to Python.
Tools:	PaiMei (http://code.google.com/p/paimei/)
Useful links:	http://www.recon.cx/en/f/lightning-ecarrera-win32-static-analysis-in-python.pdf — “Win32 Static Analysis in Python” presentation https://www.blackhat.com/presentations/bh-usa-07/Carrera/Presentation/bh-usa-07-carrera.pdf — “4 x 5: Reverse Engineering Automation with Python” presentation

Pydb

Project:	Pydb
Author:	Rocky Bernstein
Site project:	http://bashdb.sourceforge.net/pydb/
Tags:	scripting engine , debugger
License:	GNU GPL
Python versions:	less than 2.5
Platforms:	lin
Processors:	x86
Base project:	gdb
Description:	pydb is an expanded version of the Python debugger loosely based on the gdb command set and the stock Python debugger. It also has all of the features found in an earlier version of pydb.py that was distributed with the debugger GUI ddd.
Tools:	???
Useful links:	http://bashdb.sourceforge.net/pydb/pydb/lib/index.html — official documentation

PyDBG

Project:	PyDBG
Author:	Pedram Amini (@pedramamini)
Site project:	http://code.google.com/p/paimei/ (part of PaiMei)
Tags:	debugger
License:	GNU GPL v2
Python versions:	2.4-2.5
Platforms:	win/mac
Processors:	x86
Base project:	—
Description:	PyDbg exposes most of the expected debugger functionality and then some. Hardware / software / memory breakpoints, process / module / thread enumeration and instrumentation, system DLL tracking, memory reading/writing and intelligent dereferencing, stack and SEH unwinding, exception and event handling, endian manipulation routines, memory snapshot and restore functionality, disassembly (libdasm) engine. The abstracted interface allows for painless development of custom debugger scripts.
Tools:	PaiMei — http://code.google.com/p/paimei/ In Memory Fuzzing — http://www.corelan.be/index.php/2010/10/20/in-memory-fuzzing/ , Blocks — http://nsense.dk/tools/
Useful links:	http://pedram.redhive.com/PaiMei/docs/PyDbg/ — official documentation https://www.blackhat.com/presentations/bh-usa-07/Miller/Whitepaper/bh-usa-07-miller-WP.pdf — Hacking Leopard: Tools and Techniques for Attacking http://www.piemontewireless.net/Install_PaiMei_on_Snow_Leopard — Install PaiMei on Snow Leopard http://www.securitytube.net/video/1630 — PaiMei on python25 (video) http://www.securitytube.net/video/1638 — Paimei From Svn, Idapython 0.8.0/Ida4.9Free, And Python 2.7.1 (video)

PyDbgEng

Project:	PyDbgEng
Author:	Botten, Michael Eddington (http://phed.org/) Peter Silberman (@petersilberman)
Site project:	http://sourceforge.net/projects/pydbgeng/
Tags:	scripting engine, debugger
License:	GNU GPL
Python versions:	2.5
Platforms:	win
Processors:	x86/x64
Base project:	WinDdg
Description:	PyDbgEng is a Python Wrapper For Microsoft Debug Engine. Its features include: user mode debugging, kernel mode debugging, soft and hw breakpoints, symbol server and etc.
Tools:	PyDbgExt — http://sourceforge.net/projects/pydbgext/ KStalker — http://pydbgeng.sourceforge.net/kstalker.htm
Useful links:	http://pydbgeng.sourceforge.net/examples.htm — usage examples http://flierlu.blogspot.com/search?q=PyDbgEng — series of records in blog

pydbgr

Project:	pydbgr
Author:	Rocky Bernstein
Site project:	http://code.google.com/p/pydbgr/
Tags:	debugger
License:	GNU GPL v3
Python versions:	2.6-2.7
Platforms:	lin
Processors:	x86
Base project:	gdb
Description:	A rewrite of pydb from the ground up.
Tools:	???
Useful links:	http://code.google.com/p/pydbgr/wiki/Tutorial — Installing and Using pydbgr

pydot

Project:	pydot
Author:	Ero Carrera (@erocarrera)
Site project:	http://code.google.com/p/pydot/
Tags:	visualization
License:	MIT License
Python versions:	2.5
Platforms:	win/lin/mac
Processors:	—
Base project:	up to Graphviz 2.26.3
Description:	Python interface to Graphviz's Dot language. pydot allows to easily create both directed and non directed graphs from Python.
Tools:	???
Useful links:	http://pythonhaven.wordpress.com/2009/12/09/generating_graphs_with_pydot/ — Generating Graph Visualizations with pydot and Graphviz (blog post) http://www.graphviz.org/Documentation.php — graphviz documentation https://www.ohloh.net/p/pydot — homepage of pydot

PyEA

Project:	PyEA
Author:	Roberto Paleari (@rpaleari) Lorenzo Martignoni (@martignlo)
Site project:	http://roberto.greyhats.it/projects.html
Tags:	static/dynamic code analyser
License:	GNU GPL v2 or later
Python versions:	2.5, 2.6
Platforms:	win
Processors:	x86/x64
Base project:	—
Description:	PyEA (Python Executable Analyser) is a hybrid static/dynamic code analyser written in Python. The analyser was originally developed to statically analyse IA-32 malicious programs, but has soon evolved into a generic analyser for compiled programs. PyEA currently supports PE and ELF executables, disassembles executables using a recursive disassembler, and translates each machine instruction into an intermediate form, that makes side effects explicit.
Tools:	???
Useful links:	???

PyEMU

Project:	PyEMU
Author:	Cody Pierce (@codypierce)
Site project:	http://code.google.com/p/pyemu/
Tags:	emulator
License:	New BSD License
Python versions:	2.5
Platforms:	win
Processors:	x86
Base project:	—
Description:	PyEmu tries to provide a fully scriptable IA-32 emulator in python. The aim is for security researchers and malware analysis. By having a flexible community driven emulator in a high level language one can roll their own purpose driven scripts to solve common problems.
Tools:	???
Useful links:	https://www.blackhat.com/presentations/bh-usa-07/Pierce/Whitepaper/bh-usa-07-pierce-WP.pdf — whitepaper from BH USA 07 http://www.youtube.com/watch?v=nkTb6m96cio — video from BH USA 07 http://www.inreverse.net/?p=223 — entry in blog about usage PyEMU

pyew

Project:	pyew
Author:	Joxean Piti
Site project:	http://code.google.com/p/pyew/
Tags:	scripting engine , analyze malware
License:	GNU GPL v2
Python versions:	???
Platforms:	win/lin
Processors:	x86/x64
Base project:	-
Description:	Pyew is a (command line) python tool like radare and *iew oriented, mainly, to analyze malware. It does have support for hexadecimal viewing, disassembly (Intel 16, 32 and 64 bits), PE and ELF file formats (it does code analysis the right way), following direct call/jmp instructions, OLE2 format, PDF format (limited) and more. It also supports plugins to add more features to the tool.
Tools:	???
Useful links:	http://joxeankoret.com/blog/?s=pyew — entries in blog about usage pyew

pygdb

Project:	pygdb
Author:	Michael Eddington (mike@phed.org) Frank Laub (frank.laub@gmail.com)
Site project:	http://code.google.com/p/pygdb/
Tags:	scripting engine , debugger
License:	MIT License
Python versions:	2.5
Platforms:	lin/mac
Processors:	x86
Base project:	gdb
Description:	This is a simple python wrapper around GDB. pygdb is a pygtk interface to gdb. It offers two terminal windows, one for gdb, one for the process to be debugged. On the top it has standard buttons like run, continue, step in, step over, step out and quit. On a second window you can add watches and breakpoints. Furthermore, you can inspect the backtrace and launch gvim on the current executed line by pressing a button. pygdb stays synchronized with gvim (by using gvim --servername calls).
Tools:	???
Useful links:	???

pyHIEW

Project:	pyHIEW
Author:	Elias Bachaalany (@0xeb)
Site project:	http://code.google.com/p/pyhiew/
Tags:	scripting engine, disassembler
License:	Artistic License/GPL
Python versions:	2.5 and 2.7
Platforms:	win
Processors:	x86/x64
Base project:	HIEW
Description:	PyHiew is a Hiew External Module that allows users to write Python scripts that interface with Hiew.
Tools:	???
Useful links:	https://0xeb.wordpress.com/?s=pyHiew — entries in blog about usage pyHIEW

pykd

Project:	pykd
Author:	Team (http://pykd.codeplex.com/team/view)
Site project:	http://pykd.codeplex.com/
Tags:	scripting engine, debugger
License:	Microsoft Public License
Python versions:	2.6.5
Platforms:	win
Processors:	x86/x64
Base project:	WinDdg
Description:	Python extension for WinDbg. pykd not repeat functional from Debug Engine, and implements the API, convenient for daily work in WinDbg.
Tools:	???
Useful links:	http://pykd.codeplex.com/documentation — official documentation , http://pykd.blogspot.com/ — blog about pykd (RU)

Pylibemu

Project:	Pylibemu
Author:	Angelo Dell'Aera (buffer@antifork.org , @angelodellaera)
Site project:	https://github.com/buffer/pylibemu
Tags:	emulator
License:	GNU Lesser General Public License, version 3 or later
Python versions:	2.5 or later
Platforms:	win/lin
Processors:	x86
Base project:	Libemu
Description:	Pylibemu is a wrapper for the Libemu library.
Tools:	???
Useful links:	???

pyMem

Project:	pyMem
Author:	Fabien Reboia (srounet@gmail.com)
Site project:	https://github.com/srounet/Pymem
Tags:	wrapper
License:	THE POSTCARD LICENSE
Python versions:	more than 2.5
Platforms:	win
Processors:	x86/x64
Base project:	—
Description:	Pymem is a memory wrapper built on top of python ctypes and windll imports to facilitate process memory access in Read or Write. It has functionalities such as Opening a process in debug mode, hijacking threads, listing process modules and much more.
Tools:	???
Useful links:	http://www.mmowned.com/forums/world-of-warcraft/bots-programs/memory-editing/285120-pymem-python-process-memory-editing.html — code example

pyREtic

Project:	pyREtic
Author:	Rich Smith (mynameismeerkat@gmail.com)
Site project:	http://code.google.com/p/pyretic/
Tags:	debugger
License:	GNU GPL v3
Python versions:	???
Platforms:	win/lin/mac
Processors:	x86/x64
Base project:	—
Description:	pyREtic and the REpdb debugger allow easier access to obtaining source from closed source Python applications. In a nutshell it allows you to take a object in memory back to source code, without needing access to the bytecode directly on disk. This can be useful if the applications pyc's on disk are obfuscated in one of many ways.
Tools:	???
Useful links:	http://pyretic.googlecode.com/files/pyREtic%20%20In%20memory%20reverse%20engineering%20for%20obfuscated%20Python%20bytecode.pdf — whitepaper http://prezi.com/kmyvgiobs1d/pyretic-rich-smith-blackhatdefcon-2010/ — slides from BlackHat/Defcon 2010

PySTP

Project:	PySTP
Author:	Roberto Paleari (@rpaleari) Lorenzo Martignoni (@martignlo)
Site project:	http://security.dico.unimi.it/~roberto/pystp/
Tags:	STP, solver
License:	GNU GPL v2
Python versions:	2.5
Platforms:	win/lin
Processors:	—
Base project:	STP
Description:	PySTP is a Python extension module that interfaces with STP. STP is a decision procedure for the theory of fixed-width bitvectors and arrays, and PySTP enables Python scripts to use STP.
Tools:	???
Useful links:	???

PythonGdb

Project:	PythonGdb
Author:	???
Site project:	http://sourceware.org/gdb/wiki/PythonGdb (In gdb 7 it is already included)
Tags:	scripting engine, debugger
License:	GNU GPL
Python versions:	2.x
Platforms:	lin
Processors:	x86/x64
Base project:	Gdb
Description:	Integrate Python scripting into Gdb.
Tools:	<p>gdbx — http://www.cinsk.org/wiki/En: Debugging with GDB: gdbx.py</p> <p>gdb-heap — https://fedorahosted.org/gdb-heap/</p> <p>runFuzzer — http://www.groundworkstech.com/projects/dynamips-gdb-mod</p> <p>tmalloc_gdb — http://localhostr.com/download/wBNwUx1/tcmalloc_gdb.tar</p>
Useful links:	<p>http://sourceware.org/gdb/wiki/PythonGdbTutorial — official tutorial</p> <p>http://sourceware.org/gdb/onlinedocs/gdb/Python-API.html — API</p> <p>https://www.wzdftpd.net/blog/index.php?post/2010/12/20/Python-scripts-in-GDB — entry in blog</p> <p>http://dmalcolm.fedorapeople.org/presentations/PyCon-US-2011/GdbPythonPresentation/GdbPython.html#1 — presentation from PyCON US 2011</p>

python-haystack

Project:	python-haystack
Author:	Loïc Jaquemet (loic.jaquemet@gmail.com)
Site project:	http://pypi.python.org/pypi/haystack https://github.com/trolldbois/python-haystack/
Tags:	search in memory
License:	GNU GPL
Python versions:	2.6-2.7
Platforms:	lin/win
Processors:	x86
Base project:	ctypes, python-pttrace, WinAppDbg
Description:	The basic functionality is to search in a process' memory maps for a specific C Structures.
Tools:	sslsnoop – https://github.com/trolldbois/sslsnoop ctypes-kernel – https://github.com/trolldbois/ctypes-kernel
Useful links:	???

python-pttrace

Project:	python-pttrace
Author:	Victor Stinner (@victor_stinner)
Site project:	http://pypi.python.org/pypi/python-pttrace
Tags:	debugger , wrapper
License:	GNU GPL v2
Python versions:	2.5,3.0
Platforms:	lin/bsd/darwin
Processors:	x86/x64
Base project:	—
Description:	python-pttrace is a debugger using ptrace (Linux, BSD and Darwin system call to trace processes) written in Python.
Tools:	Fuzil — https://bitbucket.org/haypo/fusil/wiki/Home
Useful links:	https://bitbucket.org/haypo/python-pttrace/wiki/Home — wiki

pytracer

Project:	pytracer
Author:	Rocky Bernstein (rocky@gnu.org)
Site project:	http://code.google.com/p/pytracer/
Tags:	debugger
License:	GNU GPL v3
Python versions:	2.5-2.7
Platforms:	win/lin
Processors:	x86
Base project:	—
Description:	A more flexible interface to <code>sys.settrace</code> allowing, for example, chained trace hooks. We allow several trace hooks to get registered and unregistered and allow tracing to be turned on and off temporarily without losing the trace hooks. You can also indicate filters on events for which trace hooks should fire and mark methods that should automatically be ignored.
Tools:	???
Useful links:	???

radapy

Project:	radapy
Author:	pancake (http://nopcode.org) nibble.ds earada (@earada)
Site project:	http://radare.org/doc/html/Section10.6.html#python
Tags:	scripting
License:	GNU GPL v3
Python versions:	2.5 and 2.6
Platforms:	win/lin
Processors:	x86/x64
Base project:	radare2
Description:	The second scripting language implemented in radare was 'python'. The python interface for C is not as nice as the LUA one, and it is obviously not as optimal as LUA, but it gives a very handy syntax and provides a full-featured list of libraries and modules to extend your script.
Tools:	???
Useful links:	http://radare.nopcode.org/y/ — radare official site

uhooker

Project:	uhooker
Author:	Core Security Technologies
Site project:	http://oss.coresecurity.com/projects/uhooker.htm
Tags:	hooker
License:	Core Security Technologies (for non-commercial use)
Python versions:	more than 2.3
Platforms:	win
Processors:	x86
Base project:	OllyDBG 1.10
Description:	The Universal Hooker is a tool to intercept execution of programs. It enables the user to intercept calls to API calls inside DLLs, and also arbitrary addresses within the executable file in memory.
Tools:	???
Useful links:	http://oss.coresecurity.com/uhooker/doc/index.html — official documentation http://www.irmpc.com/downloads/whitepapers/High-Level Reverse Engineering.pdf — usage

Vivisect

Project:	Vivisect
Author:	invisigoth kenshoto (@invisig0th)
Site project:	https://www.kenshoto.com/wiki/index.php/Main_Page
Tags:	static analysis, emulator
License:	???
Python versions:	???
Platforms:	win/lin/mac
Processors:	x86/x64
Base project:	—
Description:	Python based static analysis and emulation framework.
Tools:	???
Useful links:	http://visi.kenshoto.com/wiki/index.php/VivisectExamples — example

vtrace

Project:	vtrace
Author:	invisigoth kenshoto (@invisig0th)
Site project:	http://code.google.com/p/vtrace-mirror/
Tags:	debugger
License:	MIT License
Python versions:	???
Platforms:	win/lin/darwin/freebsd/solaris
Processors:	x86/x64
Base project:	—
Description:	vtrace is a cross-platform debugging api written in python. Each supported platform has it's own support module.
Tools:	vdebug — http://code.google.com/p/vdebug/
Useful links:	http://www.moreops.com/blog/2011/02/24/fuzzing-engine-with-vtrace/ — entry in blog https://github.com/pdasilva/vtrace_scripts – vtrace script examples

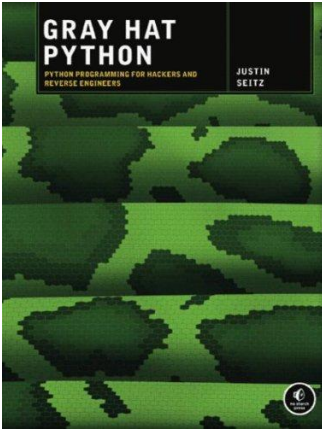
WinAppDbg

Project:	WinAppDbg
Author:	Mario Vilas (@Mario_Vilas)
Site project:	http://winappdbg.sourceforge.net/
Tags:	debugger
License:	BSD license
Python versions:	2.4-2.7 , 3.x (experimental)
Platforms:	win
Processors:	x86/x64(partial)
Base project:	—
Description:	The WinAppDbg python module allows developers to quickly code instrumentation scripts in Python under a Windows environment.
Tools:	http://winappdbg.sourceforge.net/Tools.html
Useful links:	http://winappdbg.sourceforge.net/ProgrammingGuide.html — programming guide

Z3-python

Project:	Z3-python
Author:	Sascha Böhme
Site project:	http://www4.in.tum.de/~boehmes/z3-python.html
Tags:	solver, SMT, binding, interface
License:	???
Python versions:	2.5.1 and 2.5.2
Platforms:	win
Processors:	x86/x64
Base project:	Z3
Description:	<p>This is a Python binding to the SMT solver Z3. Since it is based on Python's dynamic foreign function interface ctypes, no compilation is required. Z3 is a high-performance theorem prover being developed at Microsoft Research. Z3 supports linear real and integer arithmetic, fixed-size bit-vectors, extensional arrays, uninterpreted functions, and quantifiers. Z3 is integrated with a number of program analysis, testing, and verification tools from Microsoft Research. These include: Spec#/Boogie, Pex, Yogi, Vigilante, SLAM, F7, SAGE, VS3, FORMULA, and HAVOC. It can read problems in SMT-LIB and Simplify formats.</p>
Tools:	???
Useful links:	http://research.microsoft.com/en-us/um/redmond/projects/z3/ — site Z3

Note



In addition, I would like to note the outstanding book “GRAY HAT PYTHON” (<http://nostarch.com/ghpython.htm>) by Justin Seits, which I recommend everyone to read.

A lot of useful tips for using IDAPython and automation RE can be found at the laboratory TippingPoint (<http://dvlabs.tippingpoint.com>) in the section MindshaRE.

I would like to note, that it will be quite wrong to think that python is popular only for the purposes of RE because there is a large number of fuzzers (Peach, Sulley, PI) and web-utilities (http://www.gdssecurity.com/l/constricting_the_web_final.pdf), tools for penetration testers (<http://dirk-loss.de/python-tools.htm>) on python, designed to help security researchers.

Unfortunately I still did not manage to use all of this, but if the need arises, then I will know what can help me for sure.

Good luck with your research!

P.S. Later I will try to arrange it as a website and promptly update.

About Author

Dmitriy Evdokimov — Security Researcher.

Research areas: SAP (ABAP) security, reverse engineering, and source code analysis.

The student of St. Petersburg State Polytechnic University, computer science department, he focuses on SAP security, particularly on Kernel, BASIS and ABAP security. He has official acknowledgements from SAP and Oracle for the vulnerabilities found. His interests cover reverse engineering, software verification/program analysis (SMT, DBI, IL), vulnerability research and development of exploits, software for static and dynamic code analysis written in Python. He is a contributor to the OWASP-EAS project. "Security soft" section editor in Russian hacker magazine "XAKEP". One of the Defcon Russia (DCG #7812) and ZeroNights conferences organizers.

Contacts

Email: d.evdokimov@dsecrg.com

Twitter: [@evdokimovds](https://twitter.com/evdokimovds)

About ERPScan



ERPScan is an innovative company engaged in the research of ERP security particularly in SAP and develops products for SAP system security. Apart from this the company renders consulting services for secure configuration, development and implementation of SAP systems, and conducts comprehensive assessments and penetration testing of custom solutions.

Our flagship product "ERPScan Security Scanner for SAP" is innovative product for automatic assessment of SAP platform security and standard compliance.

About DSecRG — Research center of ERPScan



[DSecRG — Leading SAP AG partner in discovering and solving security vulnerabilities.](#) ERPScan expertise is based on research conducted by the DSecRG research center - a subdivision of ERPScan company. It deals with vulnerability research and analysis in business critical applications particularly in SAP and publishes whitepapers about it. SAP AG gives acknowledgements for security researchers from DSecRG almost every month on their site. Now DSecRG experts are on the first place in [SAP public acknowledgements](#) chart.

DSecRG experts are frequent speakers in prime International conferences held in USA, EUROPE, CEMEA and ASIA such as BlackHat, HITB, SourceBarcelona, DeepSEC, Confidence, Troopers, T2, InfoSecurity. DSecRG researchers gain multiple acknowledgements from biggest software vendors like SAP, Oracle, IBM, VMware, Adobe, HP, Kasperskiy, Apache, Alcatel and others for finding vulnerabilities in their solutions.

DSecRG has high-qualified experts in staff who have experience in different fields of security, from Web applications and reverse engineering to SCADA systems, accumulating their experience to conduct research in SAP system security.

Our Contacts

E-mail: info@dsecrg.com

Web: www.dsecrg.com