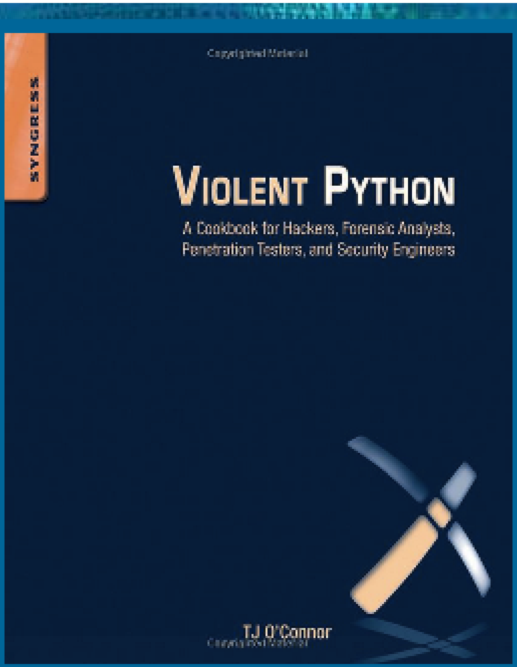
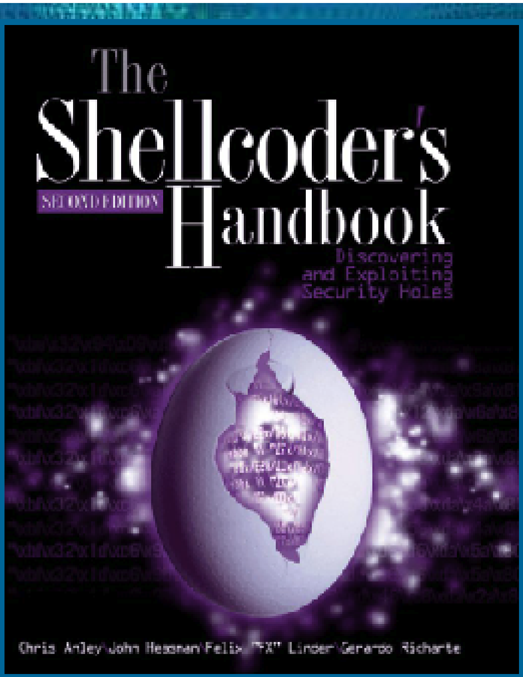

SYNGRESS
VIOLENT PYTHON
A Cookbook for Hackers, Forensic Analysts, Penetration Testers, and Security Engineers
TJ O'Connor

Violent Python and Exploit Development

Winter Working Connections -- Frisco, Texas

Dec 15-17, 2014 Sam Bowne

Schedule · Lecture Notes · Projects · Links · Home Page


The Shellcoder's Handbook
SECOND EDITION
Discovering and Exploiting Security Holes
Chris Anley John Heasman Felix "FX" Linder Gerardo Richarte

Class Description

Even if you have never programmed before, you can quickly and easily learn how to make custom hacking tools in Python. In hands-on projects, participants will create tools and hack into test systems, including:

- **Port scanning**
- **Login brute-forcing**
- **Port knocking**
- **Cracking password hashes**
- **Sneaking malware past antivirus engines**

With just a few lines of Python, it's easy to create a keylogger that defeats every commercial antivirus product, from Kaspersky to FireEye.

In the exploit development section, students will take over vulnerable systems with simple Python scripts. Hands-on projects will include:

- Linux buffer overflow
- Buffer overflow on Windows 7
- Exploiting Windows Server 2012
- Fuzzing a vulnerable server
- Structured Exception Handler exploitation on Windows
- Defeating Data Execution Protection with Return-Oriented Programming

Technical Requirements

Participants need a computer (Windows, Mac, or Linux) with VMware Player or VMware Fusion. USB thumbdrives will be available with Kali Linux and Windows Server 2008 virtual machines to use.

All the class materials are freely available on my Web page (samsclass.info) for anyone to use.

Prerequisite Knowledge

Participants should be familiar with networking and security concepts at the Network+ and Security+ level. Previous programming experience is helpful but not necessary.

Learning Outcomes

Upon successful completion of this course, the student will be able to:

A. Read and write simple Python scripts.
B. Perform network attacks, including port scanning, port knocking, and brute-forcing logins.
C. Compile Python scripts to Windows executables.
D. Bypass antivirus products with Python.
E. Find buffer overflow vulnerabilities with fuzzing.
F. Create remote code execution exploits for Linux and Windows targets.
G. Understand and defeat Windows defenses, including ASLR and DEP.

Textbooks

Violent Python: A Cookbook for Hackers, Forensic Analysts, Penetration Testers and Security Engineers by TJ O'Connor -- ISBN-10: 1597499579 (2012) Buy from Amazon

***The Shellcoder's Handbook: Discovering and Exploiting Security Holes*, by Chris Anley, John Heasman, Felix Lindner, Gerardo Richarte; ASIN: B004P5O38Q** Buy from Amazon

Schedule

Mon 12-15	Violent Python
Tue 12-16	Exploit Development
Wed 12-17	Special Topics TBA

8:30 am	morning class starts
10:30 am - 10:45 am	break
12:00 pm	morning class ends for lunch
1:00 pm	afternoon class starts
3:00 pm - 3:15 pm	break
5:00 pm	afternoon class ends

Lectures

Violent Python

Violent Python: Introduction and Motivation (pptx)

Demo: Banner-grabbing -- students do Projects 2 & 3
Demo: HTTP requests -- students do Projects 4 & 5 & 2x

When Vulnerability Disclosure Gets Ugly

Data Breaches and Password Hashes (pptx)
Links for demonstrations

Demo: Password hashes -- students do Projects 6 & 7
Security Problems at Colleges (pptx)

Demo: Antivirus evasion -- students do Projects 8 - 10

Exploit Development

Ch 1: Before you Begin (pptx)
Ch 2: Stack overflows on Linux (pptx)

Exploiting Windows: Introduction
Ch 6: The Wild World of Windows (pptx)
Buffer Overflow Defenses

The lectures are in Word and PowerPoint formats.
If you do not have Word or PowerPoint you can use Open Office.

Back to Top

Projects

Violent Python

Project 2: CodeCademy I (15 pts.)
Project 3: Basic Port Scanning with Python (15 pts. + 15 extra credit)

Project 4: CodeCademy II (20 pts.)
Project 5: HTTP Scanning with Python (15 pts. + 35 extra credit)
Project 2x: Port Scanning with IPv6 and Python (10-45 pts. extra credit)

Project 6: CodeCademy III (20 pts.)
Project 7: Password Hashes with Python (15 pts. + 40 extra credit)

Project 8: Antivirus Evasion with Python (20 pts.)
Project 9: Keylogger with Python (15 pts. + 25 pts. extra credit)
Project 10: Defeating Norton Antivirus with Python (20 pts. + 30 extra)

Project 13: XOR Encryption in Python (10 pts. + 40 extra credit)

Project 12: Automating Keypresses in Windows (10 Points + 15 pts. extra)
Project 4x: Automating Keypresses in Mac OS X (25 pts. extra)
Project 11: Cookie Cadger (15 pts.) (new 10-8-14)
Attack server configuration

Exploit Development

Proj 5: Using Jasmin to run x86 Assembly Code (15 pts.)
Proj 5x: Assembly Code Challenges (30 points)

Linux Buffer Overflow Without Shellcode
Linux Buffer Overflow
Exploiting "Vulnerable Server" for Windows 7
Windows Server 2012 Buffer Overflow Defenses and EMET
Exploiting Easy RM to MP3 Converter on Windows 7
Fuzzing "Vulnerable Server"
Developing a SEH-Based Stack Overflow Exploit for "Vulnerable Server"
Defeating DEP with ROP

Other Exploitation Projects

Project 4: Social Engineering Toolkit Java Exploit (15-25 pts.) (rev. 1-18-14)

SQL Injection Projects

Project 19: SQL Injection with SQLol (20 pts) (rev. 7-27-13)
Project 20: Exploiting SQLi with Havij and Input Filtering (20 pts) (rev. 7-27-13)
Proj SQL-X3: Exploiting a SQL Injection with sqlmap (10 pts) N
Proj SQL-X4: Fixing a SQL Injection Vulnerability with Parameterized Queries (15 pts.) N

Password Hash Projects

Project 12: Cracking Linux Password Hashes with Hashcat (15 pts.) (updated 1-31-14)
Project X16: Cracking Windows Password Hashes with Hashcat (15 pts.) (new 6-16-13)

Web Projects

Project 11: Cookie Cadger (15 pts.) (new 10-8-14)
Project 21: Hijacking HTTPS Sessions with SSLstrip (15 pts.) (revised 11-6-14) sslstrip-0.4.tar.gz
Project X6: Reverse-Engineering an Authentication Cookie (15 pts. extra credit)
Project X8: Password Guessing Games (up to 30 pts.) (URL fixed 4-22-13)
Project X9: Password Brute Force Challenges (up to 30 pts.)

Cultural Enrichment

How to view someones IP address and connection speed with TRACER T! - YouTube

I Pwned Your Server - YouTube

Back to Top

Links

Links for Chapter Lectures

Ch 1a: Anatomy of a Program in Memory - Excellent explanation from 2009
Ch 1b: assembly - difference between 'or eax,eax' and 'test eax,eax'

Ch 2a: Smashing the Stack for Fun and Profit by Aleph One
Ch 2b: Assembly Programming Tutorial
Ch 2c: GDB Command Reference - set disassembly-flavor command
Ch 2d: GDB Tutorial

Ch 3b: What's the difference of the Userland vs the Kernel?
Ch 3c: Protection ring - Wikipedia
Ch 3d: The GNU C Library: glibc
Ch 3e: linux - What is the difference between exit() and exit_group()
Ch 3f: Two excellent syscall examples with explanations
Ch 3g: c - Linux system call table or cheetsheet in assembly language - Stack Overflow
Ch 3h: NASM Tutorial
Ch 3i: Shellcode in C - What does this mean? - Stack Overflow
Ch 3k: C code to test shellcode, simpler than that in the textbook
Ch 3l: execve(2): execute program - Linux man page
Ch 3m: Linux Syscall Reference
Ch 3n: Ways to do syscall: INT 0x80 and call *%gs:0x10 explained

Ch 4a: Format String Exploitation-Tutorial By Saif El-Sherel **(updated 1-25-18, ty B Meixell)**
Ch 4b: Exploiting Format String Vulnerabilities (from 2001)
Ch 4c: Advanced Format String Attacks (Paul Haas, Slides from DEF CON 18)
Ch 4d: Advanced Format String Attacks with demo videos
Ch 4e: Defcon 18 - Advanced format string attacks Paul Haas - YouTube
Ch 4f: Introduction to format string exploits -- with helpful gdb tips
Ch 4g: Ace Stream Media Format String Vulnerability (from 2014)
Ch 4h: Cisco Email Security Appliance Format String Vulnerability (9-9-2015)
Ch 4i: Graphviz Remote Format String Vulnerability, affects Ubuntu (from 2014)
Ch 4j: Polycom - H.323 Format String Vulnerability (from 2013)
Ch 4k: Python RRDtool Module Function Format String Vulnerability (from 2013)
Ch 4l: Broadcom UPnP Stack Format String Vulnerability (from 2013)
Ch 4m: pidgin-otr log_message_cb() Function Format String Vulnerability (from 2012)
Ch 4n: atexit(3) - Linux man page
Ch 4o: GOT and PLT for pwning
Ch 4p: PLT and GOT - the key to code sharing and dynamic libraries

Ch 5a: A Memory Allocator by Doug Lea
Ch 5b: Understanding the Heap & Exploiting Heap Overflows
Ch 5c: Several Interesting Heap Overflow Example Programs
Ch 5d: Four Excellent Heap Overflow Exercises **(updated 1-25-18, ty B Meixell)**
Ch 5e: Wonderful Exploit Exercises including VMs and heap overflows
Ch 5f: Dangling Pointer paper from Black Hat 2007
Ch 5g: Working example of a "Dangling Pointers" exploit?
Ch 5h: Dangling Pointers: Vulnerability and Exploitation Basics
Ch 5i: Much ado about NULL: Exploiting a kernel NULL dereference
Ch 5j: HEAP BASED EXPLOITATION. Scott Hand CSG 2/22/12 - PDF (link working 2-28-18)

Ch 6a: theForger's Win32 API Tutorial
Ch 6b: Process Explorer
Ch 6c: Portable Executable - Wikipedia
Ch 6d: PEview (PECOFF file viewer)
Ch 6e: Rebasing Win32 DLLs
Ch 6f: Why is 0x00400000 the default base address for an executable?
Ch 6g: VA (Virtual Adress) & RVA (Relative Virtual Address) - Stack Overflow
Ch 6h: Exploiting the LNK Vulnerability with Metasploit
Ch 6i: Is there any difference between a GUID and a UUID? - Stack Overflow

Ch 6j: Service Control Manager - Wikipedia
Ch 6k: Microsoft RPC Remote Procedure Call and End Point Mapper with Network Traces
Ch 6l: Setting Up Kernel-Mode Debugging over a Network Cable Manually (Windows Debuggers)
Ch 6k: IMMUNITY Debugger
Ch 6l: Operating Systems Development - Portable Executable (PE)
Ch 6m: RPC Endpoint Mapper in a network trace

L7a: AMD64 Architecture Processor (pdf, downloads immediately) **(updated 1-25-18, ty B Meixell)**
L7b: x64 Architecture - Windows 10 hardware dev
L7c: Introduction to x64 Assembly | Intel Developer Zone
L7d: Behind Windows x64's 44-bit Virtual Memory Addressing Limit
L7e: Windows 8.1 removes the 44-bit limitation (2015)
L7f: X86-64 (AMD64) Tutorial
L7g: AMD CPUID Specification
L7h: Searchable Linux Syscall Table for x86 and x86_64
L7i: Intel® 64 and IA-32 Architectures Software Developer’s Manual
L7j: 64-bit Linux stack smashing tutorial: Part 1
L7k: x86-64 - Wikipedia--AMD 64-bit processors only use 48-bit address space
L7l: Linux/x86_64 execve"/bin/sh"; shellcode 30 bytes
L7m: Writing shellcode for Linux and *BSD - Spawning a shell
L7n: Execve Shellcode 64 bit
L7o: Writing 64-Bit Shellcode - Part 1
L7p: 64 Bits Linux Stack Based Buffer Overflow **(updated 1-25-18, ty B Meixell)**
L7q: memory management - What and where are the stack and heap? - Stack Overflow

Ch 8a: Win32 Thread Information Block - Wikipedia
Ch 8b: TEB structure (Windows)
Ch 8c: Process Environment Block - Wikipedia
Ch 8d: PEB structure (Windows)
Ch 8e: assembly - What is the "FS" "GS" register intended for? - Stack Overflow
Ch 8f: Preventing the Exploitation of Structured Exception Handler (SEH) Overwrites with SEHOP (2009)
Ch 8h: SEH Based Overflow Exploit Tutorial - InfoSec Resources
Ch 8i: Windows ISV Software Security Defenses (2010)
Ch 8j: Software Defense: mitigating stack corruption vulnerabilities (2014)
Ch 8k: SEHOP per-process opt-in support in Windows 7
Ch 8l: Vista SP1 and Server 2008: Controlling SEHOP security protection
Ch 8m: Reducing the Effective Entropy of GS Cookies (2007)

Ch 8n: HeapCreate function (Windows)
Ch 8o: Heap Overflow: Vulnerability and Heap Internals Explained - InfoSec Resources
Ch 8p: Intercepting Calls to COM Interfaces - CodeProject
Ch 8q: Exploiting Lingering Vulnerabilities in Default COM Objects (pdf, 2011)
Ch 8r: OLE/COM Object Viewer Download
Ch 8s: Active X Exploitation - InfoSec Resources
Ch 8t: HEAP SPRAYING – ACTIVEX CONTROLS UNDER ATTACK -- STEP-BY-STEP INSTRUCTIONS (2013) **(updated 1-25-18, ty B Meixell)**
Ch 8u: Dranzer | Vulnerability Analysis | The CERT Division
Ch 8v: dranzer download | SourceForge.net
Ch 8w: dzzie/COMRaider on GitHub
Ch 8x: ActiveX vulnerabilities exploitation (from 2010)
Ch 8y: Win32 Thread Information Block - Wikipedia
Ch 8qq: No Loitering: Exploiting Lingering Vulnerabilities in Default COM Objects (paper) | Internet Society
Ch 14a: What is linux-gate.so.1?
Ch 14b: Heap overflow using Malloc Maleficarum
Ch 14c: Windows 8 Heap Internals
Ch 14d: RdRand - Wikipedia
Ch 14e: "We cannot trust" Intel and Via's chip-based crypto, FreeBSD developers say
Ch 14f: Windows 10 security overview
Ch 14g: Windows heap cookie is only 8 bits long

Ch 17a: Awesome-Fuzzing: A curated list of fuzzing resources (Books, courses - free and paid, videos, tools, tutorials and vulnerable applications to practice on)

Ch 18a: Cscope Home Page
Ch 18b: Using Cscope on large projects (example: the Linux kernel)
Ch 18c: Exuberant Ctags
Ch 18d: Splint Home Page
Ch 18e: Splint the static C code checker
Ch 18f: OPEN SOURCE STATIC CODE ANALYSIS SECURITY TOOLS
CH 18g: More Tricks For Defeating SSL In Practice
Ch 18h: CVE-2003-0161 -- Sendmail prescan() function vulnerability
Ch 18i: Port 25 (SMTP) - Remote Sendmail Header Processing Vulnerability
Ch 18j: PHP Hash Comparison Weakness A Threat To Websites, Researcher Says
Ch 18k: Dangling pointer - Wikipedia
Ch 18l: How to Create a Secure Login Script in PHP and MySQL - wikiHow

Fuzz 1: Failure Observation Engine (FOE) tutorial - YouTube
Fuzz 2: Fuzz Testing for Dummies (2011)
Fuzz 3: Analyze Crashes to Find Security Vulnerabilities in Your Apps
Fuzz 4: VBinDiff - Visual Binary Diff
Fuzz 5: vbindiff(1) - Linux man page
Fuzz 6: An Introduction to Fuzzing: Using fuzzers (SPIKE) to find vulnerabilities - InfoSec Resources
Fuzz 7: Fuzzing with Peach Part 1
Fuzz 8: GlobalSCAPE CuteZIP Stack Buffer Overflow | Rapid7
Fuzz 9: Android Intent Fuzzer
Fuzz 10: Basic Fuzzing Framework (BFF) | Vulnerability Analysis | The CERT Division
Fuzz 11: HOWTO : CERT Basic Fuzzing Framework (BFF) on Ubuntu Desktop 12.04 LTS
Fuzz 12: Fuzzer Automation with SPIKE - InfoSec Resources
Fuzz 13: Fuzzing with Spike to Find Overflows
Fuzz 14: [Python] IRC Fuzzer - IRCdFuzz.py
Fuzz 15: american fuzzy lop
Fuzz 16: Bug Hunting Using Fuzzing and Static Analysis
Fuzz 17: Fuzzing Tools in Kali Linux

Ch 16a: Socket.NoDelay Property
Ch 16b: Flawfinder Home Page

Hopper 1: Use The Debugger with Hopper Disassembler/Decompiler - YouTube
Hopper 2: Tutorial
Hopper 3: Hopper Download
Hopper 4: Linux Installation
Hopper 5: Intro to Hopper - YouTube
Hopper 6: Crackmes | Reverse Engineering Mac OS X
Hopper 7: Linux x86 Program Start Up -- EXCELLENT EXPLANATION

Miscellaneous Links

SmashTheStack Wargaming Network
Great exploit tutorials from 2012 in the WayBack Machine
Exploit Exercises
farlight.org -- useful exploits and shells
Bypassing AV Scanners -- OLLYDBG PROJECT IN HERE
Valgrind Tutorial

Bypassing EMET's EAF with custom shellcode using kernel pointer (from 2011)
Disarming Enhanced Mitigation Experience Toolkit (EMET) v 5.0
Rootkits by Csaba Barta (from 2009) **(updated 1-25-18, ty B Meixell)**
PSA: don't run 'strings' on untrusted files -- WORTH EXPLOITING
From 0-day to exploit -- Buffer overflow in Belkin N750 (CVE-2014-1635)
Disarming and Bypassing EMET 5.1
BinScope Binary Analyzer -- vulnerability detector
Popular security suites open to attack -- DEP and ASLR Not Enabled
GDB: Debugging stripped binaries
USBPcap -- USE FOR PROJECTS
PBKDF2 - Wikipedia
Installing VMware Tools on Kali Linux
Kali Linux Downloads
IMMUNITY : Download
How to setup Dark Comet RAT (with download and pictures) : hacking
Cython: C-Extensions for Python -- MAKES SMALL EXEs
HT Editor -- powerful binary ELF editor
ntpdc local buffer overflow - Exploit Development example, interesting GDB commands
Seven Resume Strategies for the Long-Term Unemployed
KdExploitMe - Hackable Windows Kernel Driver -- USE FOR PROJECTS
64-bit Linux Return-Oriented Programming
Exploit Exercises -- GOOD FOR PROJECTS
WIRESHARK 1.12.4 and below Access Violation and Memory Corruption PoC
Fuzzing with AFL-Fuzz, a Practical Example (AFL vs binutils) -- USEFUL FOR PROJECT
Radare portable reversing framework
Hopper: The OS X and Linux Disassembler -- GOOD FOR PROJECTS
Gdbinit: user-friendly gdb configuration file -- GOOD FOR PROJECTS
Format String Bug Exploration -USEFUL FOR PROJECT
90s-style security flaw puts "millions" of routers at risk -- LOOKS GOOD FOR A PROJECT
Exploit Development Class for Win 7 64-bit -- USEFUL FOR PROJECTS
EDB (Evan's Debugger) -- Like OllyDbg on Linux ty @offsectraining
Sophos AV Bypass - YouTube
New buffer overflow protection in gcc 4.9 -fstack-protector-strong
Old Versions of Kali Linux
Animated Metasploit Linux Payload in gdb - YouTube
Stack Smashing On A Modern Linux System
Buffer Overflow Vulnerability Lab

VMware Tools installation fails when Easy Install is in progress -- GOOD SOLUTION

Installing VMware Tools in an Ubuntu virtual machine

How to turn OFF (or at least override) syntax highlighting in nano via ~/.nanorc?

Exploit writing tutorial part 11 : Heap Spraying Demystified | Corelan Team

MemGC and Control Flow Guard (May, 2015)

How exploit writers find bugs in Java Machine? - Reverse Engineering Stack Exchange

Mac OS Xploitation (2009)

Modern Binary Exploitation class from RPI

A binary analysis, count me if you can -- VERY USEFUL

picoCTF 2014 Baleful - Solving with Pin -- INTERESTING TECHNIQUE

How to detect a NX stack and other protections against buffer overflows -- VERY USEFUL

ROP for Linux ELF files: finding JMP ESP

Performing a ret2libc Attack **(updated 1-25-18, ty B Meixell)**

How to disable ASLR in linux permanently.

Python multiprocessing.Pool: -- EXCELLENT EXAMPLE

Rooting Freshly -- GOOD EXAMPLE OF PENETRATING A LINUX WEB SERVER

Exploiting memory corruption bugs in PHP Part 3: Popping Remote Shells

Execute Bash Commands Without Spaces with Brace Expansion

x64dbg: An open-source x64/x32 debugger for windows -- ALTERNATIVE TO IDA PRO

gdb bug on 64-bit ubuntu with fix: No module name libstdcxx - Stack Overflow

gdb - debugging with pipe using mkfifio

Fuzzing on MacOS X -- MANY USEFUL TIPS

Carnegie Mellon - Tools - VulWiki

The Ultimate Disassembly Framework -- Capstone

binjitsu/binjitsu: CTF framework and exploit development library

How To Install VMware Workstation 11 On Ubuntu 14.10

Exploitation of mem-corruptions vulns in remote C/C++ programs without source or binary

Artistic Rendering of Exploit Development Process

Blind Return Oriented Programming (BROP)

Linux Assembly Tutorial - Step-by-Step Guide

A fundamental introduction to x86 assembly programming

RIP ROP: Intel's cunning plot to kill stack-hopping exploits at CPU level with "shadow stack" (June, 2016)

Introductory Intel x86: Architecture, Assembly, Applications - YouTube

Assembly Primer for Hackers (Part 1) System Organization Tutorial.mp4 - YouTube

ARM Exploitation: Return Oriented Programming on ARM (on Linux)

How to read arbitrary RAM with format string vulnerability

The best resources for learning exploit development -- MANY GOOD PROJECT IDEAS

Use The Debugger with Hopper Disassembler/Decompiler - YouTube
Over the Wire Narnia Level 2 -) 3 -- GOOD EXTRA CREDIT PROJECT
Demystifying the Execve Shellcode (Stack Method)
Program exiting after executing int 0x80 instruction when running shellcode
Debugging - Modifying Code At Runtime
How to specify base addresses for sections with gcc -- ESSENTIAL FOR KALI 2017 PROJECTS
Windows Kernel Exploitation Tutorial
[Kernel Exploitation] 2: Payloads
Infosec_Reference/Exploit Development
Requests: HTTP for Humans -- Requests 2.18.4 documentation
PEDA - Python Exploit Development Assistance for GDB
Getting cozy with exploit development
Bypassing NX/DEP -- PoC || GTFO
Simple ASLR/NX bypass on a Linux 32 bit binary
Binary Analysis Tool -- INTERESTING FOR PROJECTS
Linux Kernel Debugging with VMWare Player Free
Force GCC to push arguments on the stack before calling function (using PUSH instruction)
Analyzing Metasploit linux/x86/exec payload
EXPLOITATION PROJECT: HeapSpray, SEH, EggHunter
Vulnserver -- GMON command SEH based overflow exploit
OakSim: ARM Assembly Simulator
ARM Assembly and Exploitation -- USEFUL FOR PROJECTS
VM of Ubuntu with ARM in QEMU
x64dbg -- Recommended by @malwareunicorn

New Unsorted Links

Radare2 Projects: "Practical case : Buffer Overflow 0x01 : https://t.co/rMSdRZFzfv 2)Methods and macros: the call stack : https://t.co/oDNYb0sAsr 3) Practical case: Patch Me 0x01 : https://t.co/Ta2cgWQm4E 4)Conditions and loops : https://t.co/hcZg1yNx3Z cc @LibraAnalysis"
L7r: x86-64 - Wikipedia
Immunity error: pycommands: error importing module -- caused by using 64-bit Python
The Cost of Buffer Security Checks in Visual C
Ch 14h: GS (Buffer Security Check) -- Official Microsoft Documentation
Enable or disable specific mitigations used by Exploit protection | Microsoft Docs
Control Flow Guard | Microsoft Docs
vulnserver/vulnserver.c at master ï¿½ stephenbradshaw/vulnserver ï¿½ GitHub
Dangling Pointers Avoid them Strictly!

Wxploiting Format Strings in Windows
6 Best Wireshark Alternatives for Android
DLL Hijacking with Ghidra--USE FOR PROJECT
wntools --CTF framework and exploit development library
Return Oriented Programming on ARM (32-bit)--USE FOR PROJECTS
Reverse Engineering with Ghidra -- USE FOR PROJECTS
Online Courses -- Ghidra
Heap Overflow Exploitation on Windows 10 Explained
Honggfuzz finding a double-free in VLC -- USE FOR PROJECT
How to Compile 32-bit Apps on 64-bit Ubuntu?
Debug 32 bit application with gdb in 64 bit environment
Modern Windows Exploit Development.pdf
Dump TEB/PEB in immunitydbg - Reverse Engineering Stack Exchange
Ch 7r: Maximum addressable memory under the current operating systems
L7r: Maximum addressable memory under the current operating systems
Demystifying Dot NET Reverse Engineering, Part 1: Big Introduction
Demystifying dot NET reverse engineering - PART 2: Introducing Byte Patching
Demystifying dot NET reverse engineering - PART 3: Advanced Byte Patching
Bypassing SEHOP

Back to Top

Last Updated: 12-17-14 5:46 am