

مهندسی معکوس

برای مبتدیان

دنيس يوريچو

محتوی کارت قرعه کشی و عضویت
در باشگاه خوانندگان پندار پارس

برگردان: مهندس محسن مصطفی جوکار

انتشارات پندار پارس

سرشناسه

: یوریچو، دنیس

Yurichev, Dennis

عنوان و نام پدیدآور : مهندسی معکوس برای مبتدیان / دنیس یوریچو ؛ برگردان محسن مصطفی جوکار.

مشخصات نشر : تهران : پندار پارس ، ۱۳۹۵.

مشخصات ظاهری : ۱۲۶۶ ص.: مصور، جدول.

شابک : 978-600-8201-24-3 : ۸۵۰۰۰۰ ریال

وضعیت فهرست نویسی : فیبا

یادداشت : Reverse engineering for beginners: عنوان اصلی:

موضوع : مهندسی معکوس

موضوع : Reverse engineering

شناسه افزوده : کا : محسن مصطفی، ۱۳۶۵ - مترجم

رده بندی کنگره : ۱۶۸TA۱۳۹۵ ۹م۹/ی

رده بندی دیویی : ۶۲۰/۰۴۲

شماره کتابشناسی ملی : ۴۷۳۹۰۵

انتشارات پندار پارس



دفتر فروش: انقلاب، ابتدای کارگر جنوبی، کوی رشتچی، شماره ۱۴، واحد ۱۶ www.pendarepars.com

تلفن: ۶۶۵۷۲۳۳۵ - تلفکس: ۶۶۹۲۶۵۷۸ همراه: ۰۹۱۲۲۴۵۲۳۴۸ info@pendarepars.com

نام کتاب : مهندسی معکوس برای مبتدیان

ناشر : انتشارات پندار پارس

تالیف : دنیس یوریچو

برگردان : محسن مصطفی جوکار

چاپ نخست : آذر ماه ۹۵

شمارگان : ۵۰۰ نسخه

طرح جلد : رامین شکرالهی

چاپ، صحافی : روز

شابک : ۹۷۸-۶۰۰-۶۵۲۹-۲۴-۳

: ۸۵۰۰۰ تومان

قیمت

* هرگونه کپی برداری، تکثیر و چاپ کاغذی یا الکترونیکی از این کتاب بدون اجازه ناشر تخلف بوده و پیگرد قانونی دارد *

سخن مترجم

به جرات می توان گفت که مهندسی معکوس، مهم ترین و سخت ترین فیلد در امنیت است. به دلیل دانش بالایی که برای یادگیری این تخصص نیاز است، متأسفانه نمی توان کتابی از پایه نوشت که تمام مفاهیم را در بر گیرد؛ اما نویسنده اصلی این کتاب تلاش کرده است که از ابتدا این تکنیک را برای شما شرح دهد. نکته مهم در مورد این کتاب این است که رایگان بوده و روز به روز به ما می افزوده می شود و با کمی جست و جو در اینترنت متوجه خواهید شد که نخستین کتابی است که تمام متخصصان امنیت به شما پیشنهاد می کنند. نویسنده در این کتاب با انجام آزمایش های عملی مانند شکستن قفل دانگل یا دست کاری برنامه ها و بازی ها، این تکنیک را برای شما جالب و جذاب می سازد. هم انتیاق شما را برای یادگیری آن بیشتر کرده است. در این کتاب مثال های زیادی را خواهید دید و حتی یک وبسایت مجزا هم به تمرین های کتاب اختصاص داده شده است. گفتنی است که این کتاب با مجوز رسمی از سوی خود نویسنده ترجمه شده است.

محسن مصطفی جوکار

mohsen1365b@yahoo.com

پاییز ۹۵

فهرست

۲	پیش‌گفتار.....
۷	بخش نخست: الگوهای کد.....
۹	فصل ۱: مقدمه‌ای کوتاه برای CPU.....
۹	واژه‌نامه‌ی کوتاه.....
۱۰	۱-۱ چند واژه درباره‌ی ISAهای گوناگون.....
۱۱	فصل ۲: ماشین‌ترین تابع.....
۱۱	۲-۱ x86.....
۱۱	۲-۲ ARM.....
۱۲	۲-۳ MIPS.....
۱۳	فصل ۳: سلام دنیا.....
۱۳	۳-۱ x86.....
۱۸	۳-۲ 86-64.....
۲۰	۳-۳ GCC-کی‌بیز.....
۲۲	۳-۴ ARM.....
۲۸	اطلاعات بیشتر درباره‌ی توابع t.....
۳۱	۳-۵ MIPS.....
۳۸	۳-۶ نتیجه.....
۳۸	۳-۷ تمرین‌ها.....
۳۹	فصل ۴: آغاز و پایان تابع.....
۳۹	۴-۱ بازگشتی.....
۴۱	فصل ۵: پشته.....
۴۱	۵-۱ چرا پشته، وارونه افزایش پیدا می‌کند؟.....
۴۲	۵-۲ پشته برای چه چیزی استفاده می‌شود؟.....
۴۸	۵-۳ طرح نمونه پشته.....
۴۸	۵-۴ نويز در پشته.....
۵۱	۵-۵ تمرین‌ها.....
۵۵	فصل ۶: Printf() با چندین آرگومان.....
۵۵	۶-۱ x86.....
۶۷	۶-۲ ARM.....
۷۵	۶-۳ MIPS.....
۸۳	۶-۴ نتیجه‌گیری.....
۸۴	۶-۵ نکته تکمیلی.....
۸۷	فصل ۷: Scanf().....
۸۷	۷-۱ نمونه‌ی ساده.....
۸۸	MSVC.....
۹۸	۷-۲ متغیرهای سراسری.....

۱۰۷.....	مقداردهی متغیر سراسری
۱۰۹.....	۷-۳ چک کردن نتیجه‌ی scanf()
۱۲۲.....	۷-۴ تمرین‌ها
۱۲۳.....	فصل ۸: دسترسی به آرگومان‌های انتقال داده شده
۱۲۳.....	۸-۱ x86
۱۲۶.....	۸-۲ x64
۱۳۰.....	۸-۳ ARM
۱۳۴.....	۸-۴ MIPS
۱۳۷.....	فصل ۹: اطلاعات بیشتر درباره‌ی بازگشت نتایج
۱۳۷.....	۹-۱ تلاش برای استفاده از نتیجه‌ی یک تابع که void را برمی‌گرداند
۱۳۸.....	۹-۲ اگر از نتیجه‌ی تابع استفاده نکنیم چه می‌شود؟
۱۳۹.....	۹-۳ بازگشت یک ساختار
۱۴۱.....	فصل ۱۰: اشاره‌گرها
۱۴۱.....	۱۰-۱ نحوه‌ی مربوط به متغیرهای سراسری
۱۴۴.....	۱۰-۲ نحوه‌ی مربوط به متغیرهای محلی
۱۴۷.....	۱۰-۳ نتایج
۱۴۹.....	فصل ۱۱: عملگر JUDO
۱۵۱.....	۱۱-۱ کد مرده
۱۵۱.....	۱۱-۲ تمرین
۱۵۳.....	فصل ۱۲: پرش‌های شرطی
۱۵۳.....	۱۲-۱ نمونه‌ی ساده
۱۵۶.....	x86 + MSVC + OllyDbg
۱۵۹.....	x86 + MSVC + Hiew
۱۷۰.....	۱۲-۲ محاسبه‌ی مقدار واقعی یک عدد
۱۷۳.....	۱۲-۳ عملگر شرطی سه تایی
۱۷۷.....	۱۲-۴ گرفتن مقادیر بیشینه و کمینه
۱۸۳.....	۱۲-۵ نتیجه‌گیری
۱۸۵.....	۱۲-۶ تمرین
۱۸۷.....	فصل ۱۳: SWITCH()/CASE/DEFAULT
۱۸۷.....	۱۳-۱ شمار کمی از case‌ها
۱۹۰.....	OllyDbg
۱۹۵.....	۱۳-۱۲ ARM: Keil 6/2013 بهینه‌سازی شده (حالت Thumb)
۱۹۹.....	۱۳-۲ شمار بسیاری case
۲۱۱.....	۱۳-۳ هنگامی که چندین دستور case در یک بلوک وجود دارد
۲۱۷.....	۱۳-۴ رها کردن
۲۱۹.....	۱۳-۵ تمرین
۲۲۱.....	فصل ۱۴: حلقه‌ها
۲۲۱.....	۱۴-۱ یک نمونه‌ی ساده
۲۲۳.....	۱۴-۲ کپی عادی پلاک‌های حافظه
۲۲۷.....	۱۴-۳ نتیجه
۲۳۹.....	۱۴-۴ تمرین‌ها

۲۴۵	فصل ۱۵: پردازش رشته‌های ساده در C
۲۴۵	۱۵-۱ strlen()
۲۵۳	ARM64
۲۵۵	۱۵-۲ تمرین‌ها
۲۵۹	فصل ۱۶: جایگزین کردن دستورهای ریاضی با دستورهای دیگر
۲۵۹	۱۶-۱ ضرب
۲۶۵	۱۶-۲ تقسیم
۲۶۷	۱۶-۳ تمرین‌ها
۲۶۹	فصل ۱۷: قسمت نقطه‌ای اعشار
۲۶۹	۱۷-۱ IEEE 754
۲۶۹	۱۷-۲ x86
۲۶۹	۱۷-۳ ARM, MIPS, x86/x64 SIMD
۲۷۰	۱۷-۴ C/C++
۲۷۰	۱۷-۱ نمونه‌ی ساده
۲۸۱	۱۷-۲ رد کردن مقدار یا ممیز شناور به وسیله‌ی آرگومان
۲۸۵	۱۷-۳ نمونه‌ی هارمونی و مقایسه
۳۱۴	۱۷-۴ پیش‌بینی، تشخیص و نشانه‌گذاری معکوس لهستانی
۳۱۴	۱۷-۵ x64
۳۱۴	۱۷-۶ تمرین‌ها
۳۱۷	فصل ۱۸: آرایه‌ها
۳۱۷	۱۸-۱ نمونه‌ی ساده
۳۲۶	۱۸-۲ سرریز بافر
۳۳۳	۱۸-۳ روش‌های حفاظت از سرریز بافر
۳۳۷	۱۸-۴ یک واژه‌ی بیشتر درباره‌ی آرایه
۳۳۸	۱۸-۵ آرایه‌ای از اشاره‌گرها به رشته
۳۴۷	۱۸-۶ آرایه‌های چند بعدی
۳۵۶	۱۸-۷ بسته‌بندی یک رشته به عنوان یک آرایه‌ی دوبعدی
۳۶۱	۱۸-۸ نتیجه‌گیری
۳۶۱	۱۸-۹ تمرین‌ها
۳۷۹	فصل ۱۹: دستکاری بیت‌های ویژه
۳۷۹	۱۹-۱ بررسی بیت ویژه
۳۸۴	۱۹-۲ تنظیم و پاکسازی بیت‌های ویژه
۳۹۲	۱۹-۳ شیفت
۳۹۲	۱۹-۴ تنظیم و پاک کردن بیت‌های ویژه: مثال FPU
۳۹۹	۱۹-۵ شمارش بیت‌هایی که به یک تنظیم شده‌اند
۴۰۶	GCC 4.8.2 بهینه‌سازی نشده
۴۱۴	۱۹-۶ نتیجه
۴۱۶	۱۹-۷ تمرین‌ها
۴۲۵	فصل ۲۰: تولیدکننده‌ی هم‌جنس خطی به تولیدکننده‌ی عدد شبه تصادفی
۴۲۶	۲۰-۱ x86
۴۲۷	۲۰-۲ x64

۴۲۸	ARM ۲۰-۳ ۳۲ بیتی
۴۲۹	MIPS ۲۰-۴
۴۳۱	۲۰-۵ نسخه‌ی نخ‌کشی شده‌ی امن از مثال
۴۳۳	فصل ۲۱: ساختار
۴۳۳	۲۱-۱ MSVC: مثال SYSTEMTIME
۴۳۷	۲۱-۲ تخصیص فضا به ساختار به کمک malloc()
۴۴۰	۲۱-۳ struct tm: UNIX
۴۵۳	۲۱-۴ Field packing در ساختار
۴۵۷	OllyDbg + فیلدهای ترازشده بر روی مرز ۱ بیتی
۴۶۱	۲۱-۵ ساختارهای تودرتو
۴۶۴	۲۱-۶ فیلدهای بیت در یک ساختار
۴۶۵	Msvc
۴۶۷	MSVC + OllyDbg
۴۶۸	GCC
۴۷۳	۲۱-۷ تدبیرها
۴۷۹	فصل ۲۲: اتحاد
۴۷۹	۲۲-۱ نمونه‌ی مربوط به تولید اعداد تصادفی
۴۸۱	۲۲-۱-۱ x86
۴۸۵	۲۲-۲ ماشین حساب اسیپسون
۴۸۸	۲۲-۳ محاسبه‌ی سریع ریشه دوم
۴۸۹	فصل ۲۳: اشاره‌گر به تابع
۴۹۰	۲۳-۱ MSVC
۴۹۶	۲۳-۲ GCC
۵۰۳	فصل ۲۴: مقادیر ۶۴ بیتی در محیط ۳۲ بیتی
۵۰۳	۲۴-۱ بازگشت مقدار ۶۴ بیتی
۵۰۴	۲۴-۲ رد کردن، جمع، تفریق آرگومان‌ها
۵۰۹	۲۴-۳ ضرب، تقسیم
۵۱۳	۲۴-۴ شیفت به راست
۵۱۵	۲۴-۵ تبدیل مقدار ۳۲ بیتی به ۶۴ بیتی آن
۵۱۷	فصل ۲۵: SIMD
۵۱۸	۲۵-۱ بردارسازی
۵۱۹	Intel C++
۵۳۰	۲۵-۲ پیاده‌سازی strlen() از نوع SIMD
۵۳۵	فصل ۲۶: ۶۴ بیت
۵۳۵	۲۶-۱ x86-64
۵۴۴	۲۶-۲ ARM
۵۴۴	۲۶-۳ اعداد ممیز شناور
۵۴۵	فصل ۲۷: کار کردن با اعداد ممیز شناور با استفاده از SIMD
۵۴۵	۲۷-۱ نمونه‌ی ساده
۵۵۰	۲۷-۲ انتقال عدد با ممیز شناور به وسیله‌ی آرگومان‌ها
۵۵۲	۲۷-۳ نمونه‌ی مربوط به مقایسه

۵۵۴	۲۷-۴ ماشین حساب اپسیلون x64 و SIMD
۵۵۵	۲۷-۵ بررسی دوباره‌ی نمونه‌ی مربوط به تولیدکننده‌ی اعداد شبه تصادفی
۵۵۶	۲۷-۶ خلاصه
۵۵۷	فصل ۲۸: جزئیات مختص ARM
۵۵۷	۲۸-۱ نشانه‌ی شماره (#) پیش از عدد
۵۵۷	۲۸-۲ آدرس دهی حالت
۵۵۸	۲۸-۳ بارگذاری یک ثابت در یک ثابت
۵۶۰	۲۸-۴ Relocs در ARM64
۵۶۳	فصل ۲۹: جزئیات مخصوص MIPS
۵۶۳	۲۹-۱ بارگذاری ثابت‌ها در ثابت
۵۶۳	۲۹-۲ مطالعه‌ی بیشتر درباره‌ی MIPS
۵۶۵	بخش ۲: اصل مهم
۵۶۷	فصل ۳۰: نمایش عدد علامت‌دار
۵۶۹	فصل ۳۱: ENDIANNESS
۵۶۹	۳۱-۱ big-endian
۵۶۹	۳۱-۲ Little-endian
۵۶۹	۳۱-۳ مثال
۵۷۰	۳۱-۴ bi-endian
۵۷۰	۳۱-۵ تبدیل داده
۵۷۱	فصل ۳۲: حافظه
۵۷۳	فصل ۳۳: CPU
۵۷۳	۳۳-۱ پیش‌بینی شاخه
۵۷۳	۳۳-۲ وابستگی داده
۵۷۵	فصل ۳۴: توابع هش
۵۷۵	۳۴-۱ تابع یک‌طرفه چگونه کار می‌کند؟
۵۷۷	بخش ۳: نمونه‌های پیشرفته‌تر
۵۷۹	فصل ۳۵: تبدیل دما
۵۷۹	۳۵-۱ مقادیر از نوع عدد صحیح
۵۸۲	۳۵-۲ مقادیر اعشاری
۵۸۵	فصل ۳۶: اعداد فیبوناچی
۵۸۵	۳۶-۱ مثال ۱
۵۸۸	۳۶-۲ مثال ۲
۵۹۱	۳۶-۳ خلاصه
۵۹۳	فصل ۳۷: نمونه‌ی محاسبه CRC32
۵۹۹	فصل ۳۸: نمونه‌ی محاسبه‌ی آدرس شبکه
۶۰۱	۳۸-۱ calc_network_address()
۶۰۲	۳۸-۲ form_IP()
۶۰۴	۳۸-۳ print_as_IP()
۶۰۵	۳۸-۴ set_bit() یا form_netmask()
۶۰۶	۳۸-۵ خلاصه
۶۰۷	فصل ۳۹: حلقه‌ها؛ تکرار بسیار

۶۰۷	 ۳۹-۱ سه تکرار کننده
۶۰۸	 ۳۹-۲ دو تکرار کننده
۶۱۰	 Intel C++ 2011 ۳۹-۳
۶۱۳	 DUFF'S ۴۰: دستگاه
۶۱۷	 فصل ۴۱: تقسیم بر ۹
۶۱۷	 x86 ۴۱-۱
۶۱۸	 ARM ۴۱-۲
۶۲۰	 MIPS ۴۱-۳
۶۲۱	 ۴۱-۴ چگونه کار می کند؟
۶۲۳	 ۴۱-۵ گ. فن مقسوم علیه
۶۲۴	 ۴۱-۶ تمرین ۱
۶۲۷	 فصل ۴۲: تبدیل رشته به عدد (atoi())
۶۲۷	 ۴۲-۱ نمونه ساده
۶۲۸	 x64 MSVC 2013 ۴۲-۱-۱ بهینه سازی شده
۶۳۱	 ۴۲-۲ یاد نویی به فته
۶۳۵	 ۴۲-۳ تمرین
۶۳۷	 فصل ۴۳: توابع درون برنامه ای
۶۳۸	 ۴۳-۱ رشته ها و توابع حافظه
۶۴۹	 فصل ۴۴: محدود کردن C99
۶۵۳	 فصل ۴۵: تابع بدون شاخه ABS()
۶۵۳	 ۴۵-۱ GCC 4.9.1 x64 بهینه سازی شده
۶۵۴	 ۴۵-۲ GCC 4.9 ARM64 بهینه سازی شده
۶۵۵	 فصل ۴۶: توابع VARIADIC
۶۵۵	 ۴۶-۱ محاسبه ی میانگین حسابی
۶۵۹	 ۴۶-۲ تابع vprintf()
۶۶۱	 فصل ۴۷: اصلاح رشته
۶۶۲	 ۴۷-۱ x64: MSVC 2013 بهینه سازی شده
۶۶۴	 ۴۷-۲ x64: GCC 4.9.1 بهینه سازی نشده
۶۶۶	 ۴۷-۳ x64: GCC 4.9.1 بهینه سازی شده
۶۶۷	 ۴۷-۴ ARM64: GCC (Linaro) 4.9 بهینه سازی نشده
۶۶۹	 ۴۷-۵ ARM64: GCC (Linaro) 4.9 بهینه سازی شده
۶۷۰	 ۴۷-۶ Keil 6/2013 ARM بهینه سازی شده (حالت ARM)
۶۷۰	 ۴۷-۷ ARM: Keil 6/2013 بهینه سازی شده (حالت Thumb)
۶۷۱	 ۴۷-۸ MIPS
۶۷۳	 فصل ۴۸: تابع TOUPPER()
۶۷۳	 x64 ۴۸-۱
۶۷۵	 ARM ۴۸-۲
۶۷۶	 ۴۸-۳ خلاصه
۶۷۷	 فصل ۴۹: کد به اشتباه DISASSEMBLE شده
۶۷۷	 ۴۹-۱ disassemble کردن، از یک آغاز اشتباه (x86)
۶۷۸	 ۴۹-۲ چگونه اختلالات، تصادفی disassemble شده به نظر می رسند؟

۶۸۳	فصل ۵۰: ایجاد ابهام
۶۸۳	۵۰-۱ رشته‌های متنی
۶۸۴	۵۰-۲ کد اجرایی
۶۸۶	۵۰-۳ ماشین مجازی / شبه کد
۶۸۶	۵۰-۴ چیزهای دیگر برای یادآوری
۶۸۶	۵۰-۵ تمرین‌ها
۶۸۷	فصل ۵۱: C++
۶۸۷	۵۱-۱ کلاس‌ها
۶۸۸	MSVC—x86
۶۹۱	MSVC—x86-64
۷۰۹	ostream ۵۱-۲
۷۱۰	* ۵۱-۳ ارجاع
۷۱۱	STL ۵۱-۴
۷۵۷	فصل ۵۲: شخص‌های من
۷۶۱	فصل ۵۳: ویندوز ۱۶ بیتی
۷۶۱	۵۳-۱ مثال ۱
۷۶۲	۵۳-۲ مثال ۲
۷۶۳	۵۳-۳ مثال ۳
۷۶۴	۵۳-۴ مثال ۴
۷۶۷	۵۳-۵ مثال ۵
۷۷۲	۵۳-۶ مثال ۶
۷۷۷	بخش ۴: JAVA
۷۷۹	فصل ۵۴: جاوا
۷۷۹	۵۴-۱ مقدمه
۷۸۰	۵۴-۲ برگرداندن مقدار
۷۸۵	۵۴-۳ توابع محاسبه‌ی ساده
۷۸۸	۵۴-۴ مدل حافظه JVM
۷۸۹	۵۴-۵ فراخوانی تابع ساده
۷۹۱	۵۴-۶ beep() فراخوانی
۷۹۲	۵۴-۷ تولیدکننده‌ی اعداد شبه تصادفی خطی و همجنس
۷۹۳	۵۴-۸ پرش‌های شرطی
۷۹۶	۵۴-۹ رد کردن آرگومان‌ها
۷۹۷	۵۴-۱۰ Bitfields
۷۹۹	۵۴-۱۱ حلقه‌ها
۸۰۲	۵۴-۱۲ switch()
۸۰۳	۵۴-۱۳ آرایه‌ها
۸۱۴	۵۴-۱۴ رشته‌ها
۸۱۷	۵۴-۱۵ استثنا
۸۲۲	۵۴-۱۶ کلاس‌ها
۸۲۴	۵۴-۱۷ وصله‌ی ساده
۸۳۰	۵۴-۱۸ خلاصه

۸۳۱	بخش ۵: پیدا کردن چیزهای مهم/جالب در کد
۸۳۳	فصل ۵۵: شناسایی فایل‌های اجرایی
۸۳۳	۵۵-۱ Microsoft Visual C++
۸۳۴	۵۵-۱-۱ نام تزئینی
۸۳۴	۵۵-۲-۲ Cygwin
۸۳۴	۵۵-۲-۳ MinGW
۸۳۴	۵۵-۳ Intel FORTRAN
۸۳۴	۵۵-۴ Watcom, OpenWatcom
۸۳۴	۵۵-۴-۱ نام تزئینی
۸۳۴	۵۵-۵ Borland
۸۳۵	۵۵-۵-۱ Delphi
۸۳۷	۵۵-۶ دیگر DLL‌های شناخته شده
۸۳۹	فصل ۵۶: ارتباط با جهان سبرون (WIN32)
۸۳۹	۵۶-۱ جوابی که اغلب در Windows API استفاده می‌شوند
۸۴۰	۵۶-۲ Linger: متوقف کردن تمام توابع در مازول خاص
۸۴۳	فصل ۵۷: رشته‌ها
۸۴۳	۵۷-۱ رشته‌های متنی
۸۴۳	۵۷-۱-۱ C++
۸۴۴	۵۷-۱-۲ Borland Delphi
۸۴۴	۵۷-۱-۳ Unicode
۸۴۴	UTF-8
۸۴۵	UTF-16LE
۸۴۷	۵۷-۱-۴ Base64
۸۴۸	۵۷-۲ پیام‌های خطا/اشکال‌یابی
۸۴۸	۵۷-۳ رشته‌های ورودی مشکوک
۸۴۹	فصل ۵۸: فراخوانی ASSERT()
۸۵۱	فصل ۵۹: ثابت‌ها
۸۵۲	۵۹-۱ اعداد ورودی
۸۵۲	۵۹-۱-۱ DHCP
۸۵۳	۵۹-۲ جست‌وجو برای ثابت‌ها
۸۵۵	فصل ۶۰: پیدا کردن دستورات درست
۸۵۷	فصل ۶۱: الگوهای مشکوک کد
۸۵۷	۶۱-۱ دستورات XOR
۸۵۷	۶۱-۲ کد اسمبلی با دست نوشته شده
۸۵۹	فصل ۶۲: استفاده از اعداد ورودی در هنگام ردیابی
۸۶۱	فصل ۶۳: چیزهای دیگر
۸۶۱	۶۳-۱ ایده کلی
۸۶۱	۶۳-۲ C++
۸۶۱	۶۳-۳ برخی از الگوهای فایل پایری
۸۶۲	۶۳-۴ مقایسه "اسنپ‌شات‌های" حافظه
۸۶۳	۶۳-۴-۱ رجیستری ویندوز

۸۶۳	Blink-comparator ۶۳-۴-۲
۸۶۵	بخش ۶: ویژه‌ی سیستم عامل
۸۶۷	فصل ۶۴: روش‌های رد کردن آرگومان (قراردادهای فراخوانی)
۸۶۷	۶۴-۱ cdecl
۸۶۷	۶۴-۲ stdcall
۸۶۸	۶۴-۲-۱ توابع یا تعداد آرگومان‌های متغیر
۸۶۹	۶۴-۳ fastcall
۸۷۰	۶۴-۳-۱ GCC regparm
۸۷۰	۶۴-۳-۲ Watcom/OpenWatcom
۸۷۰	۶۴-۴-۴ thiscall
۸۷۱	۶۴-۵ x86-64
۸۷۱	۶۴-۵-۱ Windows x64
۸۷۳	۶۴-۵-۲ Windows: رد کردن this (در C/C++)
۸۷۴	۶۴-۵-۳ Linux x64
۸۷۴	۶۴-۶ بازگشت مقدار از نوع float و double
۸۷۵	۶۴-۷ اصلاح آرگومان‌ها
۸۷۶	۶۴-۸ گرفتن یک اشاره به آرگومان تابع
۸۷۹	فصل ۶۵: نخ‌کشی ذخیره‌ساز حلی
۸۷۹	۶۵-۱ بازبینی تولید شده جنس خطی
۸۸۰	۶۵-۱-۱ Win32
۸۸۵	۶۵-۱-۲ لینوکس
۸۸۷	فصل ۶۶: فراخوانی سیستم (SYSCALL-S)
۸۸۷	۶۶-۱ لینوکس
۸۸۸	۶۶-۲ ویندوز
۸۸۹	فصل ۶۷: لینوکس
۸۸۹	۶۷-۱ کد مستقل از موقعیت
۸۹۲	۶۷-۱-۱ ویندوز
۸۹۳	۶۷-۲ هک LD_PRELOAD در لینوکس
۸۹۷	فصل ۶۸: WINDOWS NT
۸۹۷	۶۸-۱ CRT (win32)
۹۰۱	۶۸-۲ Win32 PE
۹۱۱	۶۸-۳ Windows SEH
۹۳۹	۶۸-۴ Windows NT بخش Critical
۹۴۳	بخش ۷: ابزارها
۹۴۵	فصل ۶۹: DISASSEMBLER
۹۴۵	۶۹-۱ IDA
۹۴۷	فصل ۷۰: دیباگر
۹۴۷	۷۰-۱ OllyDbg
۹۴۷	۷۰-۲ GDB
۹۴۷	۷۰-۳ tracer
۹۴۹	فصل ۷۱: ردیابی SYSTEM CALL

۹۴۹	strace / dtruss	۷۱-۱
۹۵۱	DECOMPILER	فصل ۷۲: ۷۲
۹۵۳		فصل ۷۳: ابزارهای دیگر
۹۵۵		بخش ۸: نمونه‌هایی از تمرین‌های مهندسی معکوس در دنیای واقعی
۹۵۷	TASK MANAGER (WINDOWS VISTA)	فصل ۷۴: شوخی عملی با
۹۶۱	استفاده از LEA برای بارگذاری مقادیر	۷۴-۱
۹۶۳	COLOR LINES	فصل ۷۵: شوخی عملی با بازی
۹۶۷	MINESWEEPER (WINDOWS XP)	فصل ۷۶: ۷۶
۹۷۳	تمرین‌ها	۷۶-۱
۹۷۵	Z3 SMT + حل‌کننده دستی	فصل ۷۷: DECOMPILER دستی
۹۷۵	decompile دستی	۷۷-۱
۹۸۰	استفاده از حل‌کننده‌ی Z3 SMT	۷۷-۲
۹۸۷		فصل ۷۸: دنگل‌ها
۹۸۷	PowerPC و MacOS Classic	۷۸-۱
۹۹۶	SCO OpenServer	۷۸-۲
۱۰۱۰	MIPS	۷۸-۳
۱۰۱۹	"QR9": مکعب روبیک الهام‌بخش از یک الگوریتم رمزنگاری آماتور	فصل ۷۹: ۷۹
۱۰۵۹	SAP	فصل ۸۰: ۸۰
۱۰۵۹	درباره‌ی سرویس‌گیرنده فاش‌ده‌سازی ترافیک شبکه SAP	۸۰-۱
۱۰۷۴	SAP ۶.0	۸۰-۲
۱۰۸۱	ORACLE RDBMS	فصل ۸۱: ۸۱
۱۰۸۱	Oracle RDBMS در V\$VERSION	۸۱-۱
۱۰۹۱	Oracle RDBMS در X\$KSMRLU	۸۱-۲
۱۰۹۴	Oracle RDBMS در V\$TIMER	۸۱-۳
۱۰۹۹	کد اسمبلی که دستی نوشته شده است	فصل ۸۲: ۸۲
۱۰۹۹	EICAR	۸۲-۱
۱۱۰۱	DEMOS	فصل ۸۳: ۸۳
۱۱۰۱	PRINT CHR\$(205.5+RND(1)); : GOTO 10 10	۸۳-۱
۱۱۰۵	مجموعه‌ی مندلیو	۸۳-۲
۱۱۱۷	نمونه‌هایی از معکوس کردن قالب‌های اختصاصی فایل	بخش ۹: ۹
۱۱۱۹	رمزگذاری XOR: آغازین	فصل ۸۴: رمزگذاری XOR
۱۱۱۹	راهنمای نورتون: ساده‌ترین رمزگذاری XOR یک بایتی امکان‌پذیر	۸۴-۱
۱۱۲۱	XOR	۸۴-۲
۱۱۲۵	MILLENIUM	فصل ۸۵: فایل ذخیره مراحل مربوط به بازی
۱۱۳۱	ORACLE RDBMS: فایل‌های SYM	فصل ۸۶: ۸۶
۱۱۴۱	ORACLE RDBMS: فایل‌های MSB	فصل ۸۷: ۸۷
۱۱۴۵	خلاصه	۸۷-۱
۱۱۴۷	مطالب دیگر	بخش ۱۰: ۱۰
۱۱۴۹	NPAD	فصل ۸۸: ۸۸
۱۱۵۱	وصله کردن فایل‌های اجرایی	فصل ۸۹: ۸۹
۱۱۵۱	رشته‌های متنی	۸۹-۱

۱۱۵۱	۸۹-۲ کد x86
۱۱۵۳	فصل ۹۰: کامپایلر درونی
۱۱۵۵	فصل ۹۱: ناهنجاری کامپایلر
۱۱۵۷	فصل ۹۲: OPENMP
۱۱۵۹	MSVC ۹۲-۱
۱۱۶۲	GCC ۹۲-۲
۱۱۶۵	فصل ۹۳: ITANIUM
۱۱۶۹	فصل ۹۴: مدل حافظه‌ی 8086
۱۱۷۱	فصل ۹۵: مرتب‌سازی دوباره‌ی بلوک‌های اساسی
۱۱۷۱	۹۵-۱ Profile-guided بهینه‌سازی
۱۱۷۳	بخش ۱۱: کتب‌ها/بلاگ‌ها که ارزش خواندن دارند
۱۱۷۵	فصل ۹۶: کتب‌ها
۱۱۷۵	۹۶-۱ ویندوز
۱۱۷۵	۹۶-۱ C/C++
۱۱۷۵	۹۶-۳ x86 / 66-6
۱۱۷۵	۹۶-۴ ARM
۱۱۷۶	۹۶-۵ رمزنگاری
۱۱۷۷	فصل ۹۷: بلاگ‌ها
۱۱۷۷	۹۷-۱ ویندوز
۱۱۷۹	فصل ۹۸: سایر مراجع
۱۱۸۱	فصل ۹۹: پرسش
۱۱۸۳	پیوست A: X86
۱۱۸۳	A-1 اصطلاحات
۱۱۸۳	A-2 ثبات‌های همه منظوره
۱۱۸۴	RAX/EAX/AX/AL A-2-1
۱۱۸۴	RBX/EBX/BX/BL A-2-2
۱۱۸۴	RCX/ECX/CX/CL A-2-3
۱۱۸۵	R8/R8D/R8W/R8L A.2.7
۱۱۸۶	R9/R9D/R9W/R9L A.2.8
۱۱۸۶	R10/R10D/R10W/R10L A.2.9
۱۱۸۶	R11/R11D/R11W/R11L A.2.10
۱۱۸۷	R14/R14D/R14W/R14L A.2.13
۱۱۸۸	RIP/EIP/IP A.2.17
۱۱۸۸	A.2.19 ثبات پرچم
۱۱۹۰	A.3 ثبات FPU
۱۱۹۰	A.3.1 کلمه‌ی کنترلی
۱۱۹۱	A.3.2 وضعیت واژه
۱۱۹۲	A.3.3 برجسب واژه
۱۱۹۳	A.4 ثبات‌های SIMD
۱۱۹۳	A.4.1 ثبات‌های MMX
۱۱۹۳	A.4.2 ثبات‌های SSE و AVX

۱۱۹۳	A.5 ثبات‌های اشکال یابی
۱۱۹۴	DR6 A.5.1
۱۱۹۴	DR7 A.5.2
۱۱۹۶	A.6 دستورها
۱۱۹۶	A.6.1 پیشوندها
۱۱۹۷	A.6.2 دستورهایی که بیشتر استفاده می‌شوند
۱۲۰۵	A.6.3 دستورهایی که کمتر استفاده می‌شوند
۱۲۱۲	A.6.4 دستور FPU
۱۲۱۵	A.6.5 دستورهایی که آپ‌کد ASCII قابل چاپ دارند
۱۲۱۷	ARM : B پیوست
۱۲۱۷	B.۱ اصطلاحات
۱۲۱۷	B.2 نسخه
۱۲۱۸	B.3 32-bit ARM (AArch32)
۱۲۲۰	B.4 64-bit ARM (AArch64)
۱۲۲۱	B.5 دستورها
۱۲۲۳	MIPS : C پیوست
۱۲۲۳	C.1 ثبات‌ها
۱۲۲۴	C.2 دستورها
۱۲۲۷	D : برخی توابع کتابخانه‌ای GCC
۱۲۲۹	E : برخی توابع کتابخانه‌ای M'VC
۱۲۳۱	CHEATSHEET : F پیوست
۱۲۳۲	F.2 OllyDbg
۱۲۳۲	F.3 MSVC
۱۲۳۳	F.4 GCC
۱۲۳۳	F.5 GDB
۱۲۳۷	کلمات اختصاری استفاده شده
۱۲۴۳	واژه نامه

پیش‌گفتار

چندین معنی از «مهندسی معکوس» وجود دارد:

- (۱) مهندسی معکوس نرم‌افزار: کاوش کردن برنامه‌های کامپایل شده؛
- (۲) اسکن کردن ساختارهای 3D و دستکاری محصولات دیجیتالی پیشرو برای تکثیر و تولید آنها؛
- (۳) ایجاد دوباره‌ی ساختار (DBMS سیستم مدیریت پایگاه داده).

این کتاب درباره‌ی معنی نخست است. مباحثی که به صورت عمیق بحث خواهد شد:

x86/x64, ARM/ARM64, MIPS, Java/JVM.

مباحثی که به صورت گذرا بررسی خواهد شد:

Oracle RDBM، Itanium، فایل‌های حافظه از کپی، LD_PRELOAD، سرریز پشته، ELF 10، قالب فایل‌های قابل حمل و اجرایی 32-bit، ساختار x86-64، بخش‌های مهم فراخوان‌های سیستمی، TLS 11، کد مستقل از موقعیت (PIC 12)، بهینه‌سازی پروفایل هدایت شده C++ STL، OpenMP، SEH.

تمرین‌ها

<http://challenges.re>

همه‌ی تمرین‌ها در وب‌سایت جداگانه‌ای قرار گرفته است:



درباره‌ی نویسنده

دنيس پوريچو (Dennis Yurichev) يك برنامه‌نويس و مهندس معكوس باتجربه است. با آدرس ايميل dennis@yurichev.com يا اسكاپ dennis.yurichev مي‌توان با او ارتباط برقرار كرد.