<<Python灰帽子>>

图书基本信息

书名：<<Python灰帽子>>

13位ISBN编号：9787121129018

10位ISBN编号：7121129019

出版时间：2011-3

出版时间：电子工业出版社

作者：[美] Justin Seitz

页数：216

译者：丁赟卿 译,崔孝晨 审校

前言

前言　　“搞定了吗？
”，这可能是在Immunity公司出现频率最高的一句话了。
你也许会在类似以下的场景中听到这样的发问：“我正要给ImmunityDebugger开发一个新的ELF加载器”，片刻停顿之后，“搞定了吗？
”或者，“我刚发现了IE浏览器的一个Bug!”又一片刻的沉寂之后，“那个漏洞利用程序搞定了吗？
”在日常的安全项目中我们几乎无时无刻地须要创建或者改写自己的安全工具，并在这些频繁的活动中始终保持高速的开发节奏，这使得Python逐渐成为了这个舞台上的明星。
你可以在下一个安全项目中选择Python作为自己的开发工具，也许你将会用它来创建一个特殊的反编译器或者开发一个完整的调试器。
　　当我走进位于南迈阿密海滩的AceHardware（美国的一家连锁五金店），沿着摆放着螺丝刀的通道走过时，常常会感到目眩。
你会看到接近50多种不同规格的螺丝刀以整齐的顺序陈列在货架上。
每一种规格的螺丝刀都与紧邻的螺丝刀有着微小却又十分重要的区别。
我不算一个合格的修理能手，因此无法准确地说出每一种螺丝刀最为理想的使用场合，但是我很确信类似的情况同样适用于我们的安全工具软件。
尤其是当你在对Web类型或者其他类型的高度定制化的应用程序进行安全审计时，你会发现每一次的审计任务都会需要一把特殊的“螺丝刀”来解决问题。
要知道能够及时地拼凑出一些类似SQLAPI函数钩子之类的安全小工具已经不止一次地拯救了Immunity的工作团队。
当然这些工具并不仅仅适用于安全审计任务，一旦你能够使用钩子函数对SQLAPI进行拦截，你就可以轻易地编写出一个工具用于实时检测可疑的异常SQL查询，并及时向你的客户公司提供修复方案，以抵御那些来自顽固黑客们的攻击。
　　众所周知，要让你的每一个安全研究人员真正成为团队的一部分是一件棘手的事情。
很多安全研究人员无论在面对何种类型的问题时，都怀揣着白手起家式的过度热情，企图将需要借助的工具库完全重写。
比如说Immunity发现了某个SSLDaemon的一个安全漏洞，接下来很有可能发生的一件事就是，你突然发现你的某个安全研究人员居然正在试图从头开始编写一个SSL客户端。
而他们对此通常给出的解释是“我能找到的SSL库都丑陋不堪”。
　　你需要尽力避免这种情况发生。
事实情况并不是现有的SSL库丑陋不堪——它只是没有按照某个安全研究人员的特别偏好风格来设计而已。
而我们真正需要做的是能够深入分析大量的现有代码，快速地发现问题所在，并对其进行修改以适应自身所需，这才是及时地搭建出一个可用的SSL库，并用其开发出一个尚处于保鲜期内的漏洞利用程序的关键。
而要做到这一点，你需要使你的安全研究员们能够像一个真正的团队一样去工作。
一个熟练掌握了Python的安全研究人员就有了一个强大的武器，也许就像那些掌握了Ruby的安全研究人员一样。
然而Python真正的与众不同之处显现在那些Python狂热分子们协同工作时，他们将犹如一个高速运转的超个体①一样战斗力惊人。
正如你家厨房中的蚂蚁大军一样，当它们的数量足够组成一只大乌贼时，要杀死它们将比杀死一只乌贼棘手得多。
而这正是本书极力告诉你的一个事实。
　　你也许已经为自己想做的事找到了一些工具。
你也许会问：“我已经有了一套VisualStudio，里面附带了一个调试器，为什么还要去编写一个供

自己专用的调试器。
”或者“WinDbg不是有一个插件接口了吗？
”答案是肯定的。
WinDbg的确提供了插件接口，你可以通过那些API慢慢地拼凑出一些有用的东西。
直到某一天你很可能又会说：“Heck，如果我能和5000个WinDbg使用者互联该有多好啊，这样我们就可以互通各自的调试结果了”。
如果你从一开始就选择了Python，你只要写100行左右的代码就可以构建一个XMLRPC客户端与服务端，接下来整个团队可以同步地进行工作并使每个人及时地享有他人的成果和信息。
　　黑客绝不等同于逆向工程——你的目标并不是还原出整个应用程序的源码。
你的目标是对软件系统获得比系统开发者自身更加深入的理解。
一旦你能做到这一点，无论目标以何种形式出现，你将最终成功地渗透它，获得炙手可热的漏洞利用（exploit）。
这也意味着你需要成为可视化、远程同步、图论、线性方程求解、静态分析技术以及其他很多方面的专家。
因此，Immunity决定将这些都标准化实现在Python平台上，这样一旦我们编写了一个图论算法，这个算法将在我们所有的工具中通用。
　　在第6章中，Justin向你演示了如何使用一个钩子窃取Firefox浏览器中输入的用户名与密码。
这正是一个恶意软件作者所做的事——从之前的一些相关报道中可以看出，恶意软件作者通常使用一些更为高级语言来编写此类程序
（http://philosecurity.org/2009/01/12/interviewwithanadwareauthor）。
然而你同样可以使用Python在15分钟内编写出一个样例程序，用于向你的开发人员演示，让他们明白他们对自己的产品所做的安全假设并不成立。
现在的一些软件公司出于他们所声称的安全考虑，在保护软件内部数据方面的投资花费不菲。
而实际上他们所做的往往只是实现了一些版权保护和数字版权管理机制而已。
　　这正是本书试图教你的东西：快速创建安全工具的能力。
你应当能够借助这种能力为你个人或者整个团队带来成功。
而这也是安全工具开发的未来：快速实现、快速修改，以及快速互联。
我想，最后你唯一剩下的问题也许就是：“搞定了吗？
”　　ImmunityIne的创始人兼CTODaveAitel　　2009年2月于美国佛罗里达州，迈阿密海滩

<<Python灰帽子>>

内容概要

本书是由知名安全机构ImmunityInc的资深黑帽JustinSeitz先生主笔撰写的一本关于编程语言Python如何被广泛应用于黑客与逆向工程领域的书籍．老牌黑客，同时也是Immunity Inc的创始人兼首席技术执行官(CT0)Dave Aitel为本书担任了技术编辑一职。

本书的绝大部分篇幅着眼于黑客技术领域中的两大经久不衰的话题：逆向工程与漏洞挖掘，并向读者呈现了几乎每个逆向工程师或安全研究人员在日常工作中所面临的各种场景，其中包括：如何设计?构建自己的调试工具，如何自动化实现烦琐的逆向分析任务，如何设计与构建自己的fuzzing工具，如何利用fuzzing测试来找出存在于软件产品中的安全漏洞，一些小技巧诸如钩子与注入技术的应用，以及对一些主流Python安全工具如PyDbg、Immunity Debugger、Sulley、IDAPython、PyEmu等的深入介绍。

作者借助于如今黑客社区中备受青睐的编程语言Python引领读者构建出精悍的脚本程序来——应对上述这些问题。

出现在本书中的相当一部分Python代码实例借鉴或直接来源于一些优秀的开源安全项目，诸如Pedram Amini的Paimei，由此读者可以领略到安全研究者?是如何将黑客艺术与工程技术优雅融合来解决那些棘手问题的。

本书适合热衷于黑客技术，特别是与逆向工程与漏洞挖掘领域相关的读者，以及所有对Python编程感兴趣的读者阅读与参考。

作者简介

作者：（美国）塞兹（Justin Seitz） 译者：丁赟卿 注释 解说词：崔孝晨

<<Python灰帽子>>

书籍目录

章节摘录

版权页：插图：缓冲区溢出是最为普遍的一种软件漏洞。
所有那些看似无辜纯良的内存管理函数、字符串处理例程、甚至是某些编程语言所固有的内建函数都有可能成为引发缓冲区溢出的潜在元凶。
简而言之，当你试图向某一内存区域中存入超负荷量的数据时，便会发生缓冲区溢出。
用一个形象的比喻来解释这个现象，你可以将数据缓冲区想象成一个容积为一加仑的水桶。
如果你只是向这个桶内倒上两滴或者半加仑水，甚至是直接满上，你和水桶都还会相安无事。
然而我们都清楚如果向水桶内倾倒上两加仑的水会导致什么后果，你恐怕得劳烦自己清理溅溢到地板上的那额外一加仑的水了。
本质上相同的事情也会发生在软件程序中，当你试图向一块数据缓冲区（水桶）写入过量的数据（水）时，额外的写入数据将越过缓冲区的边界并覆盖那些位于边界之外的数据。
如果这些被殃及的内存区域的内容最终能被恶意攻击者完全控制，那么这就意味着恶意攻击者获得了一条潜在渠道用以控制程序内部的代码执行路径，并以某种方式最终成功渗透主机。
你需要了解两种最基本形式的缓冲区溢出：基于栈的溢出与基于堆的溢出。
这两种形式的溢出在形成机理上存在着本质区别，然而却往往招致相同的后果——攻击者接管目标程序内部的代码执行流。
栈溢出的一个标志性特征就是会引发栈上数据遭受污染，这为攻击者接管后续的代码执行流向创造了绝佳机会。
恶意攻击者可能会通过重写当前函数栈帧中的返回地址，或者改写存于栈上的函数指针，或者篡改栈上变量的取值，或者修改当前的异常处理例程执行链等各种手段来设法掌控后续的代码执行方式。
由栈溢出所引发的错误内存数据访问距离溢出事件本身往往仅几步之遥，因此通常在栈溢出发生后不久，系统就会抛出一个非法内存访问异常，这使得测试人员在一轮Fuzzing测试过后要找出事发源头相对容易一些。

编辑推荐

《Python灰帽子:黑客与逆向工程师的Python编程之道》：安全技术大系

版权说明