

面向信息安全专业的程序设计课程教学改革研究*

从立钢^{1, 2} 张宇昕^{1, 2} 王杨惠^{1, 2} 刘旭^{1, 2}

(1. 长春理工大学 计算机科学技术学院, 吉林省 长春市 130022;

2. 长春理工大学 国家级计算机实验教学示范中心, 吉林省 长春市 130022)

摘要 随着网络安全产业格局的变化, 企事业单位对于信息安全人才的能力要求已从“会用工具”转向“能改底层代码、能做攻防对抗”。然而, 当前部分高校信息安全专业的程序设计课程仍沿袭计算机大类的通用教学模式, 编程训练与安全能力培养之间明显脱节。本文以长春理工大学信息安全专业的教学实践为基础, 梳理了程序设计课程教学过程中存在的内容通用化、课程体系割裂、实验缺乏对抗性等突出问题, 并以OBE理念为框架, 提出了“安全导向、层次递进、技码融合”的改革路径。具体措施包括: 按学年递进重构课程内容、将CTF赛题机制引入日常实验、以“安全健壮性”为标尺调整程序设计课程考核导向等。经过近两年的试点, 学生的底层代码审计能力和安全工具开发意识均有提升, 改革取得了初步成效。

关键词 信息安全; 程序设计课程; 案例改造; 教学改革; OBE理念

Research on the Teaching Reform of Programming Courses for the Information Security Major*

CONG Ligang^{1,2} ZHANG Yuxin^{1,2} WANG Yanghui^{1,2} LIU Xu^{1,2}

(1. College of Computer Science and Technology, Changchun University of Science and Technology, Changchun 130022, China;

2. National Demonstration Center for Computer Experimental Education, Changchun University of Science and Technology, Changchun 130022, China)

Abstract—With the evolution of the cybersecurity industry, the capability requirements of employers for information security professionals have shifted from “being able to use tools” to “being able to modify low-level code and conduct offensive and defensive operations”. However, the programming courses of the information security programs in some universities still follow the generic teaching model of the computer science discipline, and programming training is obviously disconnected from the cultivation of security capabilities. Based on the teaching practice of the information security program at Changchun University of Science and Technology, this paper sorts out the prominent problems in the teaching of programming courses, such as generic content, a fragmented curriculum system, and the lack of adversarial training in experiments. Guided by the OBE philosophy, a reform path of “security orientation, progressive layering, and technology-code integration” is proposed. Specific measures include restructuring the course content progressively by academic year, introducing the CTF competition mechanism into routine experiments, and adjusting the assessment orientation of programming courses with “security robustness” as the yardstick. After nearly two years of pilot implementation, students’ capability of low-level code auditing and their awareness of security tool development have both improved, and preliminary results have been achieved.

Keywords—information security; programming course; case redesign; teaching reform; OBE philosophy

1 引言

近年来, 网络信息安全行业的用人需求发生了巨大的变化, 相关研究指出, 网络空间安全学科的发展对人才提出了“理论扎实、工程能力强、攻防兼备”的复合型要求^[1]。然而, 部分高校的信息安全专业毕业生虽然能熟练操作Burp Suite、Nmap等主流安全工具, 但是在遇到需要阅读源码定位漏洞、或者自行编写检测引擎的场景, 就显得力不从心了。相关研究也

印证了这一问题: 企业对“工具使用者”的需求趋于饱和, 而对具备底层研发能力的“安全工程师”的缺口却持续扩大^[2]。

笔者在开展长春理工大学信息安全专业课程建设过程中, 逐渐意识到一个长期被忽视的环节——程序设计课程。编程能力不仅是写代码的技能, 更是理解计算机系统运行逻辑的认知方式。脱离了编程能力的支撑, 学生对安全技术的理解往往停留在表层, 难以深入到底层机理。已有教学改革实践表明, 如果编程课程不融入安全视角, 学生很容易形成“功能优先、安全缺位”的编码习惯^[3]。因此, 如何让程序设计课

*基金资助: 吉林省高等教育教学改革研究课题 (JLJY202324733000)

程真正服务于信息安全专业的人才培养目标，而不是仅仅作为一门“通识工具课”，是本文试图探讨的核心问题。

2 信息安全专业程序设计课程教学现状与痛点

结合笔者所在教学团队的经验，以及与国内多所兄弟院校同行的交流，以下三个痛点是比较普遍的。

2.1 教学内容通用化，安全导向不足

目前国内大多数高校的信息安全专业，其程序设计类课程直接沿用计算机科学与技术专业的教学大纲和教材，这种情况相当普遍^[4]。教学内容围绕类与对象的封装继承、算法复杂度分析等传统软件工程范式展开，对代码安全性的关注几乎为零。例如，在面向对象程序设计课程中，strcpy和gets这两个典型的不安全函数，很多教材只是简单提一句“不推荐使用”，却鲜少从内存布局的角度剖析其危险性——指针越界后相邻内存中的数据会被怎样覆盖，攻击者又如何利用这一点劫持程序执行流。又如Python课程，实验项目多半是爬虫或者数据可视化，很少涉及用Scapy构造自定义网络协议包、用socket编写端口扫描器这类安全场景下的编程训练。可见，当前程序设计课程教学对系统底层理解的重视程度远远不够。值得注意的是，已有院校开始在C语言程序设计课程中系统融入安全思维^[5]，但这类尝试多集中于单门课程内部，尚未与信息安全专业的整体课程体系形成呼应。

2.2 基础课与专业课之间存在“断层”

基础课与专业课之间存在“断层”这一问题在其他专业也存在，但是在信息安全专业显得尤为突出。学生在低年级学完程序设计课程后，进入大三学习《网络攻防技术》、《软件安全技术》等专业核心课时，往往会遭遇严重的知识衔接困难。以堆溢出漏洞分析为例，专业课要求学生理解glibc的ptmalloc内存管理机制，但前期编程课对动态内存分配的讲解仅限于“malloc分配、free释放”的语法层面，根本没涉及堆块结构、空闲链表组织方式这些底层细节。结果就是，专业课教师不得不花大量课时“补课”，重讲内存布局和汇编基础，严重挤占了本应用于前沿安全技术的教学时间。从学生的反馈来看，他们在学习安全理论时普遍感到“抽象”，很难把课本上的安全机制跟实际的代码实现对应起来。这种“两层皮”现象，归根到底是前期编程课程没有为后续安全课程铺设足够的认知基础。

2.3 实验模式偏重验证，缺乏对抗性训练

传统的程序设计课程实验有一个共同特点：程序输入和输出确定，实现步骤详尽。实验指导书对每一

步均作了详尽规定，学生按部就班地敲代码、运行、比对结果，整个过程几乎不存在不确定性。这种模式对于初学编程语法具有一定合理性，但问题在于，这种模式训练出的是一种“确定性思维”，即学生习惯于在已知条件下写出正确答案，却缺乏在不确定环境中搜寻代码脆弱性的逆向思维。这一点在CTF竞赛中体现得尤为明显：不少参赛队伍在偏重逻辑推理的Web安全赛道尚能应付，但在二进制漏洞利用和逆向工程赛道表现却很差，而这两个赛道恰恰高度依赖汇编语言功底和C/C++底层调试能力。

3 编程能力在复合型安全人才培养中的基础性作用

上一节分析了现状，本节通过几个具体的与编程能力相关的安全案例来说明其对于信息安全专业的重要性。

3.1 Heartbleed漏洞：不理解内存就无法理解漏洞

2014年爆发的“心脏滴血”漏洞是一个典型的切入点。如果只是工具层面的操作，运行一下检测脚本就能判断目标服务器是否存在漏洞，整个过程不需要理解任何底层原理。但问题在于，仅会使用工具检测漏洞与能够分析漏洞成因、编写防御补丁，在工程能力上存在本质区别。“心脏滴血”漏洞的成因是OpenSSL中TLS心跳扩展的实现代码里，memcpy函数没有对用户输入的payload长度做边界校验，导致攻击者可以通过构造超长的心跳请求，让服务器把内存中相邻区域的数据原样返回，而这些数据中可能包含私钥、密码等敏感信息。要理解这个漏洞，至少需要具备C语言编程经验和对内存管理机制的认知。缺乏此基础，漏洞分析只能停留在套用既有利用代码的层面。

3.2 格式化字符串漏洞：栈帧知识是理解前提

格式化字符串漏洞是二进制安全教学中的经典案例，也是学生普遍反映“难懂”的知识点。如果学生只会输入%x或%n观察程序崩溃，则难以触及深层原理。但如果具备了C语言变参函数的编程知识，理解了函数调用时参数在栈帧中的压栈顺序和寻址方式，就能明白%x会导致栈上数据泄露的原因，因为printf会按照格式化字符串的指示，从栈上依次读取数据并输出；而%n之所以能实现任意地址写，是因为它会把当前已经输出的字节数写入栈上某个位置指向的地址。程序设计课程应当培养学生“透过代码看本质”的思维能力^[6]，而格式化字符串漏洞恰好是一个理想的教学载体。

3.3 网络协议安全防御：离不开网络编程功底

长春理工大学在光电与军工领域有较深的学科积累,嵌入式系统和工控设备的安全防护是本校信息安全专业的一个重要方向。以ARP欺骗攻击的防御为例,如果仅仅依赖第三方安全软件,则学生的角色仍局限于“工具使用者”。但如果要求学生用C/C++或Python深入数据链路层,自己编写ARP请求/应答包的解析与过滤代码,用来解析以太网帧头、提取源/目的MAC地址、维护网关IP-MAC绑定表、对异常ARP广播进行实时告警,那这就不仅仅是一个编程练习,而是一次完整的安全工程实践,对于提高学生安全实践能力大有好处。

4 “技码融合”的程序设计课程教学改革路径

基于上述分析,笔者所在教学团队以OBE(Outcome-Based Education,成果导向教育)理念为框架^[7],尝试从学生最终应具备的专业能力出发,反向设计程序设计课程的教学目标、内容体系和评价标准,以产出为导向,从社会需求倒推人才培养方案的设计^[8]。具体而言,改革路径包含以下三个层面。

4.1 按学年递进重构课程内容

改革的核心思路是打破基础课与专业课之间的壁垒,依据安全能力的成长规律来组织程序设计课程的内容体系,这一做法与近年来计算机类课程螺旋式能力进阶的改革探索相互印证^[9]。结合长春理工大学信息安全专业22版培养方案的课程设置,并借鉴三阶项目驱动的程序设计改革经验^[10],具体安排如下:

大一阶段:在《面向对象程序设计》中引入安全编码规范,将网络探测脚本替代传统的学生信息管理系统作为核心实验项目,并在配套的课程设计中融入内存安全实践。

大二阶段:在《计算机组成原理》与《操作系统》课程中穿插内存布局、权限控制、栈帧结构等安全知识点,为后续安全课程搭建底层认知框架。

大三阶段:《Python程序设计》、《密码学》、《网络攻防技术》、《软件安全技术》等课程联动,开展漏洞挖掘与安全工具开发的实战训练。

大四阶段:依托毕业设计和生产实习,锤炼工程级代码能力。

需要说明的是,这项改革并非一蹴而就,目前在大一和大二阶段推进得相对顺利,大三、大四阶段的跨课程联动仍在逐步完善中。

4.2 将CTF赛题机制引入日常实验

这是改革中投入精力最多的一个环节。传统的验证性实验模式需要被打破,至少在部分实验项目中,

应当引入“对抗性”元素,竞赛机制对激发学生主动学习的积极性有显著效果^[11],课赛一体也已成为网络安全实战化人才培养中被广泛验证的做法^[12]。具体做法是:在面向对象程序设计实验中,采用“问题驱动”的项目化模式^[10],提供一段存在漏洞的服务端源码,要求学生完成“代码审计→漏洞定位→编写利用代码→提交补丁”的全流程。与传统的“按步骤实现功能”相比,这种模式迫使学生在解决安全问题的过程中提升编程技能,而非将编程与安全视为互不关联的两个领域。

4.3 以“安全健壮性”为标尺调整考核导向

考核方式的改革往往是最难推进的,但也是最关键的。如果考试仍然只考语法知识和算法实现,则学生自然缺乏在安全编码上投入精力的动力。因此,我们尝试在课程考核中降低客观题的比重,增加项目实战考核的权重。同时,评价标准也从单一的“功能是否正确”扩展为“功能与安全并重”。具体来说,如果学生提交的代码存在输入校验缺失、异常处理不当等可能引发安全风险的问题,即使功能本身运行正常,也会被扣减一定分数。

5 教学改革实践与初步成效

长春理工大学信息安全专业依托国家级计算机实验教学示范中心、国家级计算机信息安全与网络攻防虚拟仿真实验教学中心等平台^[13],自2022年起逐步推进上述改革思路。受限于人力和课时,改革并非全面铺开,而是选择了几门核心课程作为试点,将重点放在典型教学案例的深度改造上。以下介绍三个已经实施并取得初步反馈的案例。

5.1 典型教学案例改造

(1) 基于堆块结构的内存管理器与溢出模拟

这个案例替代了《面向对象程序设计》课程设计中的“学生信息管理系统”链表实验,其改造思路与近年来网络安全综合课程设计的改革方向一致^[14]。原实验只关注指针的增删改查,学生经常因为指针操作不当导致程序崩溃,但崩溃之后往往无从判断问题所在。改造后的实验要求学生用C/C++实现一个基于隐式空闲链表的堆内存管理器,管理一块连续的内存空间。关键环节在于测试阶段:学生需要故意构造超出分配大小的写入数据,观察相邻内存块的控制结构如何被覆盖,并解释程序随后的崩溃或异常行为。这个案例把枯燥的指针操作变成了对堆溢出漏洞成因的直观体验,学生反馈表明,其对指针操作的风险性有了更直观的认识。

(2) SQL盲注自动化利用工具开发

这个案例替代了《Python程序设计》中基于

Requests库的爬虫实验。原实验侧重HTTP请求和正则表达式解析，与安全场景基本没有交集。改造后，实验升级为“SQL盲注自动化脚本开发”：针对预先搭建的存在注入漏洞的靶场（依托虚拟机环境部署），不使用SQLMap等现成工具，要求学生手写代码实现基于布尔的盲注逻辑。学生需要处理HTTP响应的差异性判断，设计二分查找算法来加速数据猜解，还要考虑网络延迟和异常重试的处理。这个案例的难度明显高于原实验，但从学生反响来看，大多数学生能够完成基本功能，个别学生还主动加入了多线程优化和日志记录功能。

（3）基于Scapy的ARP欺骗与防御系统

这个案例设在《网络攻防技术》课程中，替代了传统的Socket聊天室实验。学生首先需要用Scapy库手动构造ARP请求和应答包，实现局域网内的中间人流量劫持，通过该步骤让学生直观感受到ARP协议“无状态、不验证”的设计缺陷。随后，在攻击代码的基础上，要求学生开发防御模块：实时嗅探网络中的ARP广播包，维护一份网关IP-MAC绑定表，对异常的ARP应答进行告警。这个案例的亮点在于，它不是单纯的攻击实验或防御实验，而是“先攻后防”的完整闭环，迫使学生在攻击实践中理解防御的必要性。

5.2 实践成效观察

改革试点以来，教学团队从学科竞赛表现、课外安全工具研发、毕业设计选题和课程实验考核四个维度，对改革前后的主要教学观测量进行了跟踪统计：以改革前三届学生（2020—2022年）的数据为基线，与试点后（2023—2025年）的观测值进行对比，结果见表1。需要说明的是，试点按自然班级滚动推进，未设置严格意义的对照组，以下数据反映的是纵向变化趋势，对其解读还需结合更大样本的后续检验。

表 1 改革前后主要教学观测量对比

观测指标	单位	改革前	试点后
网络安全学科竞赛PWN/Reverse赛道年均解题数	题	7	19
学生自研安全工具与脚本原型	件/年	6	18
毕业设计工程实践类选题占比	%	17	41
课程实验“安全健壮性”达标率	%	58	86

从表1可以看出，产出类指标的改善幅度较为明显：竞赛PWN和Reverse赛道的年均解题数与自研安全工具原型的数量均增至改革前的3倍左右，毕业设计中工程实践类选题的占比从不足两成升至四成以上；课程实验“安全健壮性”达标率的提高（提升约28个百分点）则更直接地反映了课堂层面的变化。

其一，底层代码实践能力有所提升。图1给出了试点班与往届班在课程实验安全能力四个维度上的实验报告平均得分对比，试点班各维度得分均高于往届班，其中“安全编码规范”维度的提升幅度最大。结合竞赛表现看，参赛队伍在PWN和Reverse赛道的解题数量较改革前明显增加，部分学生已经能够独立完成基础堆溢出利用题目的求解。虽然与顶尖队伍仍有差距，但已初步改变了以往“仅能应对Web方向、难以涉足二进制方向”的局面。

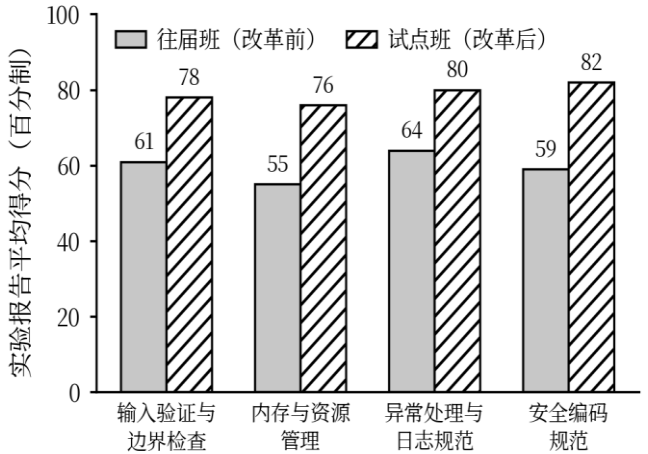


图 1 试点班与往届班课程实验安全能力维度得分对比

其二，学生面对安全验证需求时，越来越多地倾向于自主编写脚本而非依赖图形化工具。在课程实验和课外科技活动中，学生提交的安全工具原型数量有所增长，虽然质量参差不齐，但“愿意动手编写”本身即是一个积极信号。

其三，毕业设计选题方向出现了一些可喜的变化。选择安全工具原型开发、底层安全机制实现等工程实践类题目的毕业生比例较以往有所上升。部分用人单位反馈，近几届毕业生的代码工程规范性和底层调试能力有所增强，适应研发岗位的周期有所缩短。当然，这些反馈样本量较小，尚不足以作为严格的统计结论。

5.3 反思与不足

以上工作虽然取得了一定成绩，但仍存在明显不足，主要表现在：首先是试点范围有限，案例改造目前仅在部分课程和班级中推行，尚未形成覆盖全专业的系统性方案，不同教师之间的执行力度也存在差异；其次是评价机制的细化程度不够，“安全健壮”的考核标准尚未完全量化，在实际操作中存在一定的主观性。上述不足将在后续教学实践中逐步改进，计划引入基于OBE理念的课程达成度评价体系，对改革效果进行更严谨的量化评估。

6 结束语

网络信息安全领域的技术深度，在很大程度上取

决于从业者的代码实现能力。缺乏代码实现能力的安全从业者，很难真正理解漏洞的本质，更谈不上开发有效的安全防护方案。从这个意义上说，程序设计课程不应该是信息安全专业的“边缘工具课”，而应当被提升为支撑整个专业能力体系的核心基础课程。

长春理工大学的教改实践初步表明，通过技码融合与实战引领，特别是依托内存管理、协议栈构造、自动化利用等底层教学案例的深度改造，能够在一定程度上扭转学生过度依赖现成工具的倾向，提升其漏洞挖掘与安全工具研发的意识及能力。但是，这些改革举措仅仅是开始，课程内容的深度整合、实战场景的持续构建、定量评估体系的建立，都需要在后续的教学实践中不断迭代。复合型信息安全人才的培养是一项系统工程^[15]，程序设计课程的改革只是其中一个环节，但它是一个不可忽视的基础性环节，开展针对该问题的研究非常必要。

参 考 文 献

- [1] 李建华. 多元化多层次网络空间安全人才培养创新与实践[J]. 信息安全研究, 2018, 4(12): 1073-1082.
- [2] 翁健, 魏林锋, 张悦. 网络空间安全人才培养探讨[J]. 网络与信息安全学报, 2019, 5(3): 44-53.
- [3] 周宇, 杨晴雯, 周鲁梅, 等. 面向安全编程训练的Web程序设计教学改革与实践[J]. 计算机教育, 2024(12): 124-128+133.
- [4] 黄继平, 卢玲, 黄贤英. 新工科背景下以“模糊角色、赋能学生”理念驱动的程序设计课程教学改革与创新[J]. 计算机教育, 2024(02): 55-60.
- [5] 彭聪, 王慧, 冯琦, 等. 面向安全思维的C语言程序设计课程教学改革与实践[J]. 计算机技术与教育学报, 2025, 13(5): 72-76.
- [6] 尚俊平, 刘合兵, 张丽. 基于能力导向的程序设计基础课程教学改革实践——以河南农业大学为例[J]. 河南教育(高教), 2020(9): 41-44.
- [7] 顾佩华, 胡文龙, 林鹏, 等. 基于“学习产出”(OBE)的工程教育模式——汕头大学的实践与探索[J]. 高等教育研究, 2014(1): 27-37.
- [8] 吴爱华, 侯永峰, 杨秋波, 等. 加快发展和建设新工科主动适应和引领新经济[J]. 高等工程教育研究, 2017(01): 1-9.
- [9] 胡亚慧, 胡俊, 刘亮, 等. 计算机类课程螺旋式能力进阶探索与教学实践[J]. 计算机技术与教育学报, 2025, 13(7): 127-130.
- [10] 戴晓东, 谢鑫, 梁卫芳, 等. 数智赋能的三阶项目驱动程序设计教学改革与实践[J]. 计算机技术与教育学报, 2025, 13(5): 21-27.
- [11] 宋晓锋, 倪林, 韩鹏, 等. CTF竞赛融入网络安全人才培养过程的探索与实践[J]. 计算机教育, 2021(11): 1-5.
- [12] 詹东阳, 叶麟, 王莘, 等. 面向网络安全实战化人才培养的体系化课赛一体教学方法[J]. 计算机技术与教育学报, 2025, 13(3): 193-196.
- [13] 张振锋, 陈杰. 基于云平台的信息安全专业云实验教学系统构建实践[J]. 实验技术与管理, 2021, 38(9): 251-255.
- [14] 刘静, 赖英旭, 杨胜志, 等. 新工科背景下的网络安全综合课程设计教学改革探索[J]. 计算机教育, 2020(15): 135-139.
- [15] 陶琳, 李小勇. 高校培养“五育”网络安全人才的路径研究[J]. 网络空间安全, 2022, 13(3): 88-93.