

浅谈数学对于网安方向的重要性

原创

傲娇的天秤座先森OvO 于 2020-07-04 02:04:35 发布 284 收藏 1

分类专栏: [日常学习](#) 文章标签: [算法](#) [安全](#) [程序人生](#) [经验分享](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: <https://blog.csdn.net/Lemon1006/article/details/107118789>

版权



[日常学习](#) 专栏收录该内容

3 篇文章 0 订阅

订阅专栏

0x00 简单介绍

个人博客: <https://blog.gxzhang.cn>

还请大家都多关照呀~

首先声明, 本人在数学和算法上没什么造诣, 只是发表一下自己的观点吧, 最近一段时间在重新学习c语言。写这篇文章, 主要是最近忙着重新学习一遍c的基础上, 深入学习着重数据结构和算法方向, 另外是一个小学弟私聊我, 说在学习计算机相关领域数学需不需要特别好。(PS: 他着重跟我说的将来要从事网安方向) 由感而发写下这篇文章。简单说一下个人见解, 大佬勿喷, MCM参加两次分别H、M奖, 国赛运气好拿到一个国二。

CTF和ACM是我的下一个阶段的目标!!!

附一个参加MCM的经历吧: <https://blog.gxzhang.cn/20200225/2738.html>

0x01 正文

简单来说各位数学从幼儿园就开始, 一直读到大都有接触, 也是密不可分的。从1+1到高等数学...大学计算机专业学生都有感触, 计算机专业课程中最难的几门课程莫过于离散数学、编译原理、数据结构, 当然像组合数学、密码学、计算机图形学等课程也令许多人学起来相当吃力, 很多自认为数据库学得很好的学生在范式、函数依赖、传递依赖等数学性比较强的概念面前感到力不从心, 这些都是因为数学基础或者说数学知识的缺乏所造成的。数学是计算机的基础, 这也是为什么考计算机专业研究生数学都采用最难试题(数学)的原因, 当然这也能促使一些新的交叉学科如数学与应用软件、信息与计算科学专业等飞速发展。

0x02 究竟哪里重要?

在大学的C语言中我们学习认真看书的同学们都知道, 编程=算法+数据结构, 出自于谭浩强的C语言程序设计。在这里我梳理一下:

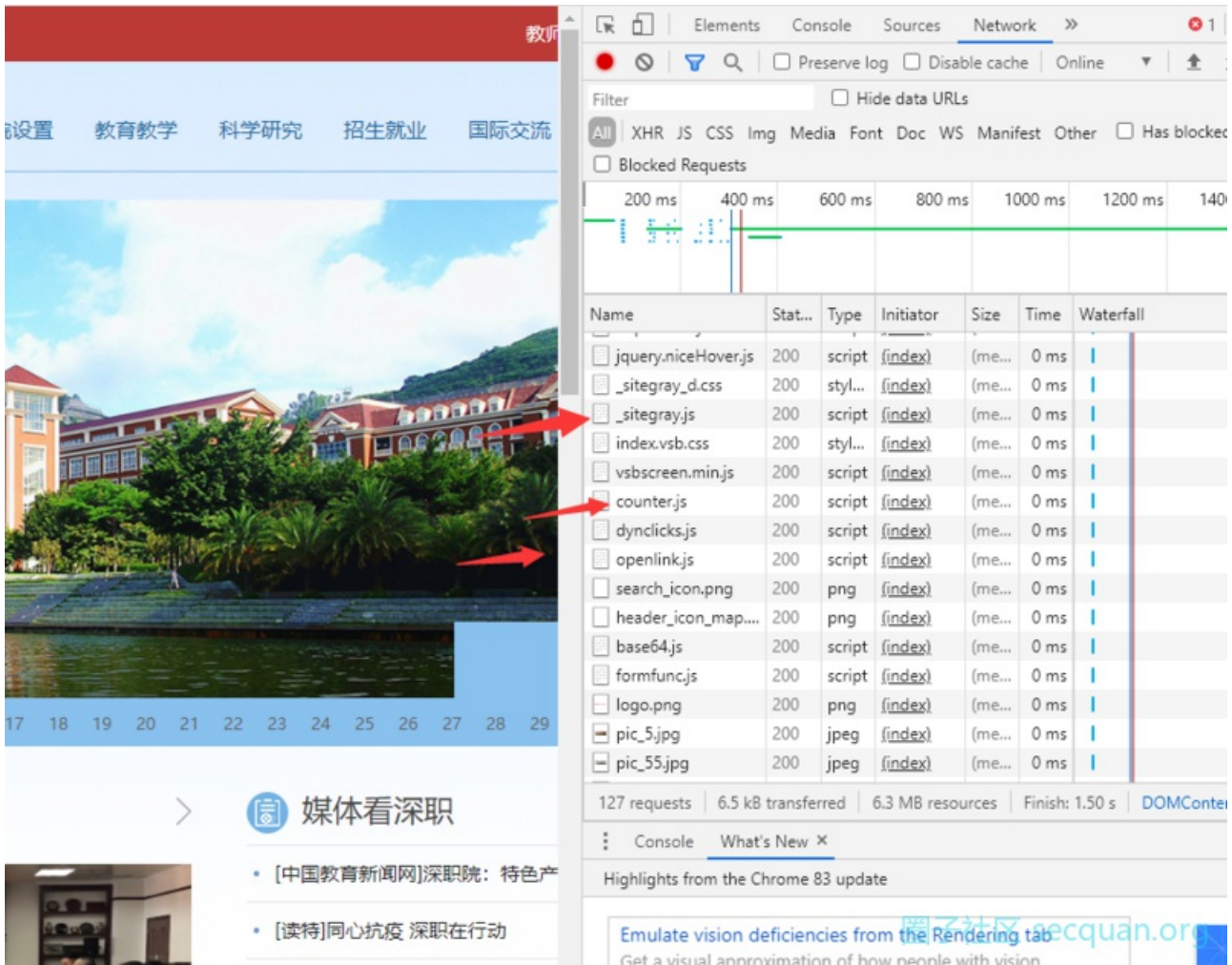
- 数学好的人逻辑好
- 逻辑好的人算法好
- 算法好的人学习数据结构上手就快

所以简单来说, 数学好的人编程不会差, 当然也会有特例, 反正我认识的朋友数学好的, 写的程序简洁明了, 算法优秀, 建模能力也很强。

0x03 那么对于网安方向的重要性

Github上的poc基本上都是python来写的，那么你懂poc编写规则也很容易，那么为什么大牛们挖到0day编写了poc来检测漏洞，编写了exp来利用。对于当下这个时代来说，不会是手工去完成渗透检测、利用。有时间可以看看大厂的招聘要求，基本上都是需要自己编写自动化程序，那么自动化的意义是什么呢？就是说可以用工具来代替我们人工检测的这个过程。

之前一个表哥给我说的挖洞思路，在访问站点的过程中打开F12中的Network模块，去看代码的过程很重要。



包括国光小姐姐的一篇代码审计：<https://www.sqlsec.com/2020/01/sinsiu.html>

代码审计的过程，对于你能否编写出这么一套cms可能难度大，当你的逻辑性好，数学能力强，你去阅读代码，去看他的函数和算法，Bypass,上传漏洞，逻辑漏洞。都是靠自己去挖掘，去审计他的代码。其实很多同学们（PS：可能只有我这个半吊子）喜欢去看一些实战性的文章，从而疏忽了对于基础的深入挖掘，大多数实战文章，没有不从代码入手的，包括你去了解sqlmap的原理也好，去做逆向安全也好，去分析APP也好，学习安全方向很多。还有大佬们发布的CTF的writeup，重点都是对于代码的审计，那么在这个过程中，函数你看不懂，逻辑运算看不懂，算法看了一脸懵逼。希望大家在学习的过程中要注意的就是脚踏实地，从基础一步一步的往上爬，不要做一个脚本小子，大佬们写的poc能不能看懂就拿来用，都说sql注入那么简单，可注入毕竟在于owasp的TOP10中排名第一吧？

T10 OWASP Top 10 应用安全风险- 2017 6	
A1:2017-注入	将不受信任的数据作为命令或查询的一部分发送到解析器时，会产生诸如SQL注入、NoSQL注入、OS注入和LDAP注入的注入缺陷。攻击者的恶意数据可以诱使解析器在没有适当授权的情况下执行非预期命令或访问数据。
A2:2017-失效的身份认证	通常，通过错误使用应用程序的身份认证和会话管理功能，攻击者能够破译密码、密钥或会话令牌，或者利用其它开发缺陷来暂时性或永久性冒充其他用户的身份。
A3:2017-敏感数据泄露	许多Web应用程序和API都无法正确保护敏感数据，例如：财务数据、医疗数据和PII数据。攻击者可以通过窃取或修改未加密的数据来实施信用卡诈骗、身份盗窃或其他犯罪行为。未加密的敏感数据容易受到破坏，因此，我们需要对敏感数据加密，这些数据包括：传输过程中的数据、存储的数据以及浏览器的交互数据。
A4:2017-XML 外部实体 (XXE)	许多较早的或配置错误的XML处理器评估了XML文件中的外部实体引用。攻击者可以利用外部实体窃取使用URI文件处理器的内部文件和共享文件、监听内部扫描端口、执行远程代码和实施拒绝服务攻击。
A5:2017-失效的访问控制	未对通过身份验证的用户实施恰当的访问控制，攻击者可以利用这些缺陷访问未经授权的功能或数据，例如：访问其他用户的帐户、查看敏感文件、修改其他用户的数据、更改访问权限等。
A6:2017-安全配置错误	安全配置错误是最常见的安全问题，这通常是由于不安全的默认配置、不完整的临时配置、开源云存储、错误的HTTP标头配置以及包含敏感信息的详细错误信息所造成的。因此，我们不仅需要所有的操作系统、框架、库和应用程序进行安全配置，而且必须及时修补和升级它们。
A7:2017-跨站脚本 (XSS)	当应用程序的新网页中包含不受信任的、未经恰当验证或转义的数据时，或者使用可以创建HTML或JavaScript的浏览器API更新现有的网页时，就会出现XSS缺陷。XSS让攻击者能够在受害者的浏览器中执行脚本，并劫持用户会话、破坏网站或将用户重定向到恶意站点。
A8:2017-不安全的反序列化	不安全的反序列化会导致远程代码执行，即使反序列化缺陷不会导致远程代码执行，攻击者也可以利用它们来执行攻击，包括：重播攻击、注入攻击和特权升级攻击。
A9:2017-使用含有已知漏洞的组件	组件（例如：库、框架和其他软件模块）拥有和应用程序相同的权限。如果应用程序中含有已知漏洞的组件被攻击者利用，可能会造成严重的后果或数据丢失或服务器接管。同时，使用含有已知漏洞的组件的应用程序和API可能会破坏应用程序防御、造成各种攻击并产生严重影响。
A10:2017-不足的日志记录和监控	不足的日志记录和监控，以及事件响应缺失或无效的集成，使攻击者能够进一步攻击系统、保持持续性或转向更多系统，以及篡改、提取或销毁数据。大多数缺陷研究显示，缺陷被检测出的时间超过200天，且通常通过外部检测方检测，而不是通过内部流程或监控检测。

真正的原理懂了嘛？sqlmap使用手册百度一大把，原理可以懂了嘛，绕waf插件是否会用，插件自己能写的出来吗？其实大家在实战能获得更大的提升，但是希望大家一定要打好基础。数学思维可以培养一个人的逻辑，基础课要认真学于此之外，工具的合理运用要知道原理。

写在最后

其实希望大家越来越好，挖的洞也越来越多，从基础开始分析，真正的去理解这个漏洞的原理，工具的原理。如果说没有什么方向的话从owasp top10的漏洞开始看，数学的思维也不是一天两天能明白的，希望大家都可以写出自己工具，早早有自己的0day！

本文原文地址：<https://blog.gxzhong.cn/20200703/31153.html>



[创作打卡挑战赛](#) >

[赢取流量/现金/CSDN周边激励大奖](#)