绿盟科技应急响应中心安全研究员邓永凯：那些年，你怎么写总会出现的漏洞...

原创

csdn业界要闻 于 2017-12-01 16:25:41 发布 717 收藏
文章标签： 看雪安全开发者峰会 绿盟科技 邓永凯 安全

本文链接：https://blog.csdn.net/csdn_bang/article/details/80133056

11月18号，2017看雪安全开发者峰会在北京悠唐皇冠假日酒店举行。来自全国各地的开发人员、网络安全爱好者及相应领域顶尖专家，在2017看雪安全开发者峰会汇聚一堂，只为这场“安全与开发”的技术盛宴。

源代码作为软件的最初原始形态，其安全缺陷是导致软件漏洞的直接根源。与之相对的是，解决代码的完整性与安全性，实现安全编码也成为减少漏洞的根本解决办法。但让人无奈的是，在编写数量繁多且关系复杂的代码过程中，难免会出现各种各样的问题，即使是具有多年开发经验的老司机也是一样。这些问题就像是高考英语易错短语一样，成为每个程序员编写代码的“必经之路”。

本峰会上，绿盟科技应急响应中心安全研究员邓永凯对开发者在开发编码时最容易产生的漏洞以及意料之外的漏洞进行了分析，阐述其原因以及应该注意的地方。通过没有防御到防御绕过、错误的防御姿势、错误的使用方法、错误的修复方法、系统及语言自身特性、设计缺陷、二次漏洞、程序员的惰性导致的漏洞等方面进行实例讲解。

绿盟科技应急响应中心安全研究员 邓永凯

> 邓永凯，Web安全研究员。绿盟科技从事安全工作6年，主要负责Web与系统漏洞扫描产品的开发及攻防代表、Web漏洞挖掘及分析、Web安全研究工作。 现于绿盟科技应急响应中心从事Web安全研究工作。擅长代码审计、Web漏洞挖掘、安全自动化、渗透测试。曾创办《安全参考》、《书安》等免费电子安全杂志，国内多个漏洞提交平台及SRC核心白帽子，SSC安全大会演讲者。

以下为演讲速记：

邓永凯：大家下午好。很荣幸跟大家探讨关于安全开发的问题。为什么取这个名字，《那些年，你怎么写总会出现的漏洞》？今天是安全开发者大会，段总说在座各位基本上都是开发为主的，我们在开发的过程当中不仅仅要把功能完善好，也要注意在开发过程中的安全问题，是不是你就会在你的代码里面埋下一些雷。

自我介绍一下，我之前也是一个程序员，主要负责绿盟科技的系统和web漏扫开发工作，并担任攻防产品攻防代表，后来加入研究院做web和IoT安全研究的工作，之前基于爱好也办过一些电子杂志《安全参考》、《书安》，现在由于某种原因已经停办了，大家网上可以找得到。

今天主要从这三个方面给大家展开来讲，利用大量案例讲解初级程序员、高级程序员以及疯狂程序员，他们在开发编码的过程当中是怎样把漏洞写出来的。为什么我加了防御、做了过滤、漏洞已经修复还是不断被黑客攻击？

初级程序员开发的时候基本上不会考虑安全问题，因为他只要把任务完成了，功能开发OK了，上线运行OK就可以了，他不会考虑到安全漏洞，或者没有安全开发的概念。很简单，直接获取参数进到数据库操作里面，一个赤裸裸的注入。直接把参数拼接到系统命令里面，执行命令里面，很暴力的命令注入。另外文件上传，直接获取文件上传无任何判断。这些简单例子功能上来看没有问题，我可以查阅数据，可以执行命令，可以上传文件，但是有一个共同的缺陷就是，它获取数据之后根本没有考虑到用户数据是正常数据还是恶意数据，如果传入恶意数据那么这些地方就会被黑客利用，那么你的业务功能也就会存在问题。初级开发者就会说：“我把东西只要写完了，功能已经OK，上线正常运行了，你现在跟我说安全开发，什么是安全编码？我只是个写代码的啊，你要我怎样？”。

高级程序员经验非常丰富做过很多项目，有很多的经验。对用户的数据进行一些处理，或者是加一些过滤，给业务里面加一些软WAF等。但是，虽然做了这些安全措施，如果不到位或者是不完整或者是错误的方法，错误的修复或者是加过滤了，结果是一样的仍然存在漏洞。比如获取参数了，加入数据库之前进行了处理，这个时候把用户数据过滤了，但是在数据库操作时这个ID没有单双引号的保护，这个过滤有什么用？过滤跟没有过滤不是一个样子吗？一些经验的开发者在代码全域入口加上软WAF服务，通过一些请求包把获得的参数统一过滤了，都可转义了，而且变量的数据库进到数据里面加上一些引号保护，也就是说想来注入，必须绕过我的单双引号，这个时候就没有办法了，感觉很完美！别忘了获取数据不仅仅通过GET、POST、COOKIE，通过别的方式也可以获取，比如FILES、SERVER，前面就没有考虑。上传这个文件的考虑你的文件有单双引号吗，这样就不受全局软WAF的影响。通过count做一些信息带到库里面，虽然软AWF服务，但是这个里面Select根本不受你的控制。

很多时候程序员在写代码的时候会经过各种各样的处理，在进行核心操作的时候比如说查询数据库或者进入数据库的时候会给你一个反编码，因为他害怕里面有一些他不想要的东西，比如说不合法的东西，他先给你反编码一下，或者说他干脆给你一个反转义，比如说反转义，那么前面处理转义已经没有用了，而且进入数据库的时候跟他有一个反编码、反转义，那提交数据的时候编码不就行了吗，多来几次编码，编码之后就没有恶意数据了，提交数据后你再反转义反编码不照样恶意数据就还原回去了。还有其他的案例，它想到了安全问题，你要逃逸单引号，我把你变成两个单引号，单引号成对出现之后就没有逃逸功能，就没有办法注入了。我输入单引号你给我编两个单一号，我输入一个斜杠单引号，但是斜杠并没有处理，斜杠是一个转义的的功能。(如果数据库用的是postgresql，就没有办法绕过了，因为它每个这个反斜线。不同的数据库也有不同的效果。)

下面这个案例是一个意料之外的问题。这个都进入到那个prepare里面，把里面’%s’,”%s”统一替换为’%s’，要注入首先逃逸单引号，但是前面做了处理逃逸单引号是不可能的，所以是没有问题的。这样来看可能是没有问题，他所有的东西都包括进来了，而且对方想要逃逸单双引号是不可能，恰恰是格式化的时候就有一个问题，比如说这是一个合法格式化的东西，格式化的内容。第一次进入prepare函数之后进行一个替换，变成下面的一个，这还是’%s’，再替换一下，有一些数据库拼接起来，所以这个地方红色标入的地方都有一个单引号。但是那个单引号最后被吃掉了，为什么中间单引号不见了，这就是格式化字串的东西。这个东西1就是代表了格式化第一个类型，后面那个单引号是干什么，是格式化时候的一个附加值用来做padding的，比如说单引号后面必须有一个字符，如果这个字符不够会以单引号后面的另外一个字符附加进去。附加多少也是在后面一个数字，如果你不给数字的话就返回空，比如说这个里面单引号%S，就是不加任何东西反馈一个空，这个时候恰恰那个里面一个单引号就被当成Padding功能干掉了，这个时候就逃逸单引号进行注入了，这是一种绕过方法。另外一种绕过方法，可以传入空格+%s+空格，第一次替换变成这样，这样会不会有问题？如果第一个%S被拒了是不是就直接执行了？这个是合法，而且可以成功执行。看起来是非常正常而且考虑很多功能的代码，还是会产生很多的问题，也有很多办法可以绕过。

做项目的时候有这样一个案例，在前台找到注入管理的帐号，但是管理员没有办法登陆，因为找不到后台，前台的功能有限，所以做渗透测试的时候拿不到shell都是耍流氓。你找不到后台，前台没有办法让你登陆，你就只能看代码。前台限制登陆的时候，有这样一个代码。如果你登录的用户名是admin就直接退出了，前台登不了，后台也登不了，这样写大家看看是不是没有问题，但是这里存在一个登录绕过。

MySQL存储数据的时候有格式，首先是字符集用什么语言存储，哪种比对格式，Ci，大小写不敏感。那么ADMIN前面判断不等admin，但是查询出来这个结果是一样的。这样ADMIN就绕过了。

第二种情况，刚刚前面也有一句话代码set names utf8，这个功能是什么，将这个客户端所有的字符集都变为utf8，服务端模拟为latin1，这字符集有差异，有差异的时候就会进行这个字符集的转换，转换的过程中就有这个错误的编码字符。继续绕过！

获取一个用户名，获取一个密码，密码Md5加密，看你用户名密码对不对，这是很简单的几句话。刚刚开始看的时候我也没发现有什么问题，但是你注意到这个MD5有两个参数，一个是字符串，一个输出格式。第二个参数默认为False。如果为True就是以16位原始二进制字符返回，如果是False就是32位十六进制数。如果输入129等等一大串之后，返回的内容里面有一个单引号'Or'8其他字符，这个时候相当于一个万能密码。为什么呢，把后面那个密码带到那个密码之后，'Or'8其他字符就相当于万能密码了，而且这个O单引号的后面必须是数字开头，这个时候查询出来的数据是OK的，相当于一个万能密码绕过。如果'Or'8其他字符后面是字母或者非数字的话就不行了，因为'Or'的字符是数字开头则永远返回True，这个时候就绕过了。可以写一个脚本爆破一下，只要有一个单引号数字开头的就可以了，就可以绕过了。

前面都是各种数据传输过程中出现了缺陷，有一些比较聪明，把数据在传输过程中都签名一下，签名的过程中肯定有密钥，密钥拿不到，你没有办法伪造的签名，你有问题也没办法利用。但是签名并不是万能的。万一您的Key被泄露了怎么办，或者你的Key直接可以破解。你的Key通过各种各样的途径被泄露，或者是被破解。这些例子具体就不讲了，因为每一个例子都很多样式，我们博客（http://blog.nsfocus.com ）里面可以把这个东西搜索出来看一下，只要你的Key泄露之后就继续伪造数据进行注入，进行漏洞利用。

二次漏洞利用，大家经常听到这样的话"用户一切输入都是有害的"，这句话说得还不够完整，应该是"进入核心操作的一切数据都是有害的"。因为数据在系统里面核心操作的时候，这些数据不一定是用户直接传进来的，也有可能是从其他地方查出来的，比如说另外一个系统，另外一个存储介质或者另外一个数据库或者配置文件里面取出来的，你取出来之后在某一天某一个时刻，通过另外一个地方又传进去了，就是来回的处理的过程，这个里面会引发多次的漏洞利用。比如说用户先输入了一个数据，插入这个数据库里面了，或者插入到文件或者全局数组里面，跟踪了一大堆但是他没有利用这个数据，但是存到存储介质里面，某一天某一个时刻他从这个存储介质里面取出来了，因为你在想你是从数据库或者存储介质里面取出来东西，不是用户直接传给你，他就是安全的，你可以直接进行操作了。但是你取出来这个数据，是有隐患的数据，是不可信任的数据。比如说先插入一个用户名，进行操作的时候转义了。某一天取出来了，取出来了之后他想这是我取的东西，不是用户输给我的再进行一次操作，再继续注入，这是很简单的一个二次注入。

某知名摄像头0day：前面有一个设置数据的地方，第一次先设置进去，第二次再取出来，取出来之后没有进行处理直接命令执行，这样就可以利用。

还有把上面所有东西加起来一起用，有很多的方法来防御，我也有很多的方法绕过。

在文件上传的时候我给你限制各种能执行的后缀，但是有三种方法，第一个给bypass.php加一个X，这个x是%80-%99。第二个是在windows，小于号代表星号，星号可以代表任何字符，第一次传一个空文件，然后在覆盖之前的文件。第三种就是直接::$DASTA，就是这个数据流。另外一个通用方法就是直接在文件后面加/.，在打开这个文件先判断这个文件是不是目录，如果目录就报错，不是目录就会打开。怎么判断是不是目录，就要以斜杠，不是斜杠就是打开。获取最后打开文件长度的时候，把那个斜杠那个点去掉了，经过一切处理，就直接打开了，就导致任意代码写入了。

还有GD库的处理，虽然我们上传的图片被gd库处理了，但是如果我们上传一个特意构造好的图片，那么处理后，图片里面存在php代码，这个时候后缀可控那么就导致代码执行。
利用条件竞争，上传文件了，处理之后我给你删了，但是你在删了之后，再处理的过程中有一个时间差，利用那个时间差可以生成另外一个文件。上传了一个文件，有一堆处理，处理之后删除，但是再删除之前有一个时间差，利用这个时间差可以写多线程角本，不停请求再生成另外一个就可以了。这是一个国外知名厂商的安全设备的一个0day漏洞。

命令执行时使用easpaceshellarg和easpaceshellcmd，如果这两个函数没有用对的话，本来没有漏洞就造出一个漏洞，就有这样一个例子。你输入一个引号他给你转移，最后处理完了之后单引号不见了。

命令执行的时候会限制你的字符，比如说这里限制你的输入内容必须是数字字母，我们可以用这两个方法绕过。还有限制你的长度，根据输出命令只能是七个字符或者五个字符，或者只能是四个字符，这个时候怎么执行命令，拿不到shell？利用这种方法可以搞定，把你的命令分割了，切割之后创建成一个文件，再利用各种各样的构造，然后将命令输入到一个文件，构造完了之后再执行那个文件，那个文件里面包含你所需要的命令。

说来说去无论就是Bypass，Bypass真的是一门艺术，不同的人，不同的漏洞，不同的场景，具有不同的方法。架构层，资源层，规则层，人的层面都可以Bypass。你的代码写的再好，肯定也有薄弱的地方。只要你有薄弱的地方，我就可以绕过你。

程序员们“狠死你们这帮黑客”，我写了那么多的东西，做了各种防御你还各种绕，绕了还给我报到漏洞平台，我天天加班。

疯狂程序员，一个重置密码的功能就能有十几种方法绕过，我想不通一个重置密码功能这么多人开发出不同的逻辑，这些逻辑里面都有问题，都可以重置密码。是不是很疯狂？

这是某银行的项目，修改A用户信息时，可以添加一个字段，换成B用户的手机号码，正常情况下把A用户信息修改了，有漏洞的情况下把B用户信息修改了，它这儿不是，修改A用户的时候A用户信息被B用户信息覆盖了，而且把A用户信息删掉了，还进入到了B用户，如果你把B用户修成一个老板或者一个大BOSS，你瞬间进入到他的帐户里面去了，你就很有钱了。

还有这样一个项目只有一个登陆框，登陆不了什么也干不了，这是一个硬件设备。把这个固件下来，这里面也有一大堆的参数，也没有接口，访问一下这个文件，构造这个参数它真的可以访问，而且直接进入到后台了。想不通它这个是为什么，估计是传说中的后门。

还有指哪修哪，永远修不好，永远修不完。第四次绕过，第八次绕过，最后程序员被开除了。这是真实的案例。当时我们也提了很多的漏洞，他说他们已经把程序员开除了。

一切漏洞都来源于有缺陷的代码，但是一切代码都是可爱的程序员写的，谢谢大家！

> 注：本文根据大会主办方提供的速记整理而成，不代表CSDN观点。

2017看雪安全开发者峰会更多精彩内容：

- 2017看雪安全开发者峰会在京召开 共商网络安全保障之策
- 中国信息安全测评中心总工程师王军：用技术实现国家的网络强国梦
- 兴华永恒公司CSO仙果：Flash之殇—漏洞之王Flash Player的末路
- 中国婚博会PHP高级工程师、安全顾问汤青松：浅析Web安全编程
- 威胁猎人产品总监彭巍：业务安全发展趋势及对安全研发的挑战
- 启明星辰ADLab西南团队负责人王东：智能化的安全——设备&应用&ICS
- 自由Android安全研究员陈愉鑫：移动App灰色产业案例分析与防范
- 腾讯反病毒实验室安全研究员杨经宇：开启IoT设备的上帝模式
- 腾讯游戏安全高级工程师胡和君：定制化对抗——游戏反外挂的安全实践
- 绿盟科技网络安全攻防实验室安全研究员廖新喜：Java JSON 反序列化之殇
- 阿里安全IoT安全研究团队Leader谢君：如何黑掉无人机