畸形解析漏洞

漏洞产生原因：配置文件php.ini中开启了cgi.fix_pathinfo
/etc/php5/fpm/pool.d/www.conf中不正确配置security.limit_extensione，导致允许将其他格式文件作为php文件格式执行。

在nginx .8.03环境中，新建一个文件夹，其内容为 < ? php phpinfo()?>然后将其迷名字修改为test.jpg。

之后在浏览器中访问http://test.com/test.jpg 显示图片解析错误，然后再次访问http://test.com/test.jpg/test.jpg/test.png没有报错，而显示了Access denied。

原因在于，Nginx拿到了文件路径或者说是URL，/test.jpg/test.png/然后服务器发现是php，之后就认为了他是php文件，便把test.jpg当成了执行文件，有因为后缀为.jpg不是php文件，于是返回了 Access denied

这其中设计到了一个php的其中一个选项是cgi.fix_pathinfo，这个值默认的是1,表示开启，这个选项的用处就是对文件目录进行处理。

比如当文件Php遇到文件路径为 /aa.aa/bb.bb/cc.cc/ 如果CC不存在则pass掉cc，然后判断bb，如果没有则继续pass，以此类推。

该选项配置在pho.ini中，若是关闭此选项，访问 http://test.com/test.jpg/test.png 只会出现找不到文件。

如果关闭了这个选项则会到处出现一些小小的问题，所以这个选项是默认开启的。

但是这一段代码并没有与寻行，而是返回了一个Access denied，因为新版本的pho引入了securitu.limit_extenstons，限制了其可执行文件的后缀，默认之允许执行php文件。

这一个漏洞是因为Nginx配置不当而造成的一个与Nginx版本无关，但是高于php版本的。

但是由于这个漏洞因为引入了 security.limit_extensions，使得该漏洞难以运行成功

为什么是Nginx中的Php才会有的问题？
因为Nginx只要是一看只要是URL结尾为.php的，便不管该文件是否存在，直接交给Php进行处理

而Apache和一些其他程序，是会先看这个是否存在，才而作出处理的。

cgi.fix_pathinfo是php所具有的，如果在php前便开始判断暖，则cgi.fix_pathinfo就会拍不上用场了，这个漏洞自然也不会存在。

而iis在这一点都是同样的，都会存在这问题。