远程文件包含漏洞实验

一【实验目标】

- 了解远程文件包含攻击的方式
- 掌握远程文件包含攻击防范方法。

二【实验环境】

- Windows 10 操作系统
- xampp

三【实验原理】

"远程文件包含漏洞"是在通过 PHP 的函数引入文件时，由于传入的文件名没有经过合理的校验，从而操作了预想之外的文件，导致意外的文件泄露甚至恶意的代码注入。

程序开发人员通常会把可重复使用的函数写到单个文件中，在使用某些函数时，直接调用此文件，而无须再次编写，这种调用文件的过程一般被称为包含。程序开发人员都希望代码更加灵活，所以通常会将被包含的文件设置为变量，用来进行动态调用，但正是由于这种灵活性，从而导致客户端可以调用一个恶意文件，造成文件包含漏洞。

文件包含漏洞在 PHP Web Application 中居多。PHP 常见的导致文件包含的函数如下：include()、include_once()、require()、require_once()、fopen()、readfile()，当使用前 4 个函数包含一个新的文件时，只要文件内容符合 PHP 语法规范，那么任何扩展名都可以被 PHP 解析，包含非 PHP 语法规范源文件时，将会暴露其源代码。后 2 个函数会造成敏感文件被读取。

要防范目录穿越与远程文件调用攻击，可以使用下列方法。

(1)不要使用使用者提供的文件名。

(2)检查使用者输入的文件名中是否有“..”的目录级层的字符。

(3) php.ini 文件中设置 open basedir 来指定可以打开文件的目录。

(4) php.ini 文件中设置 allow_url_fopen 为 Off，让 Web 应用程序不能打开远程文件。

(5) realpath 与 basename 函数来处理使用者输入的文件名。

四【实验步骤】

实验具体操作步骤如下：

1. 进入系统，输入密码“Admin123456”，如图 1 所示。

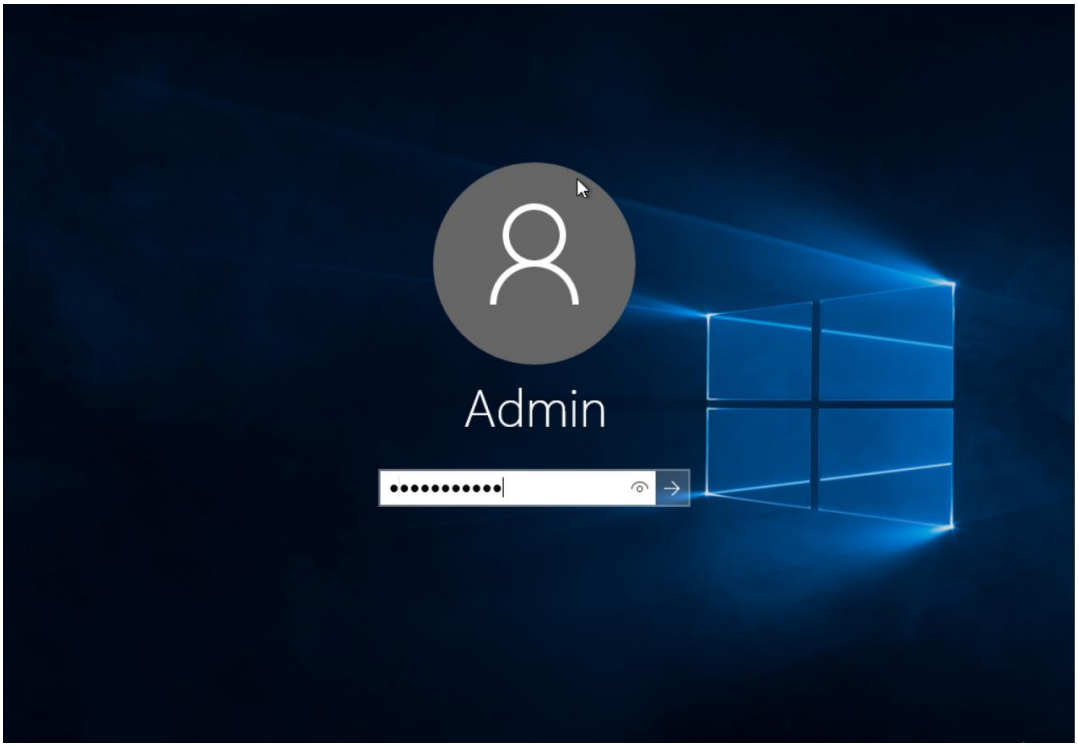

图 1

2. 点击“文件资源管理器”进入 C 盘“xampp”文件夹,找到 php 文件夹下的“php.ini”文件,如图 2 所示。

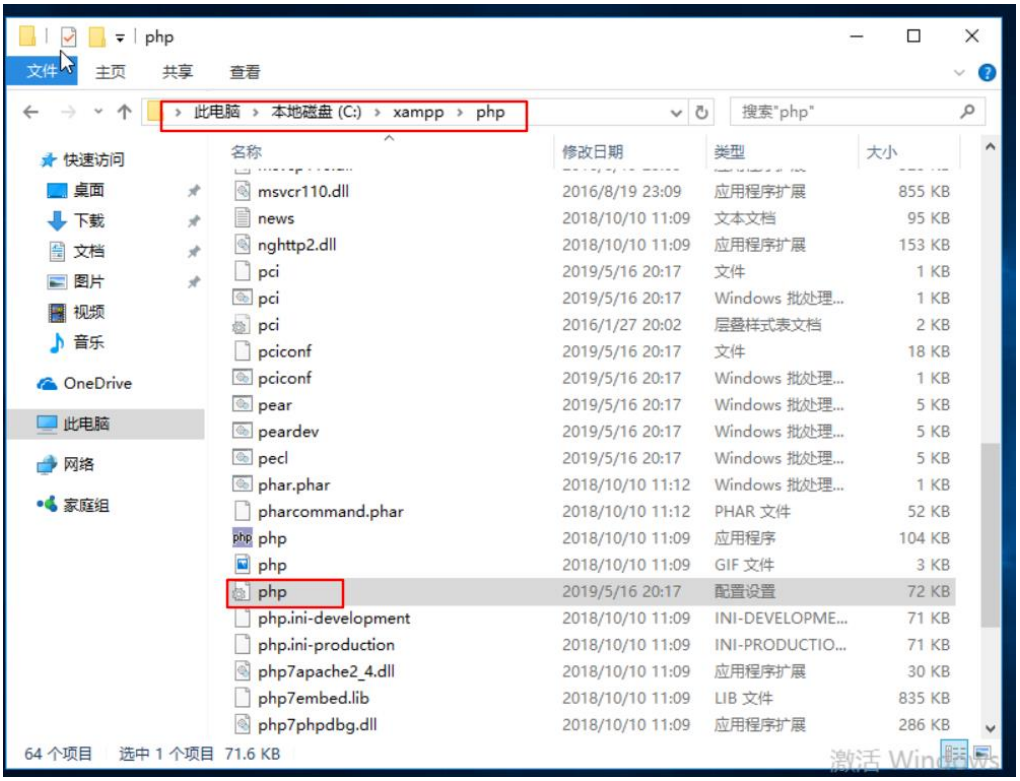

图 2

3. 将 php.ini 文件中的 allow_url_fopen 设置为 On，就可以将远程文件的 URL 当做是本机的文件来处理。并且可以将远程文件的 URL 发送给 readfile，fopen 等函数来处理。如图 3 所示。

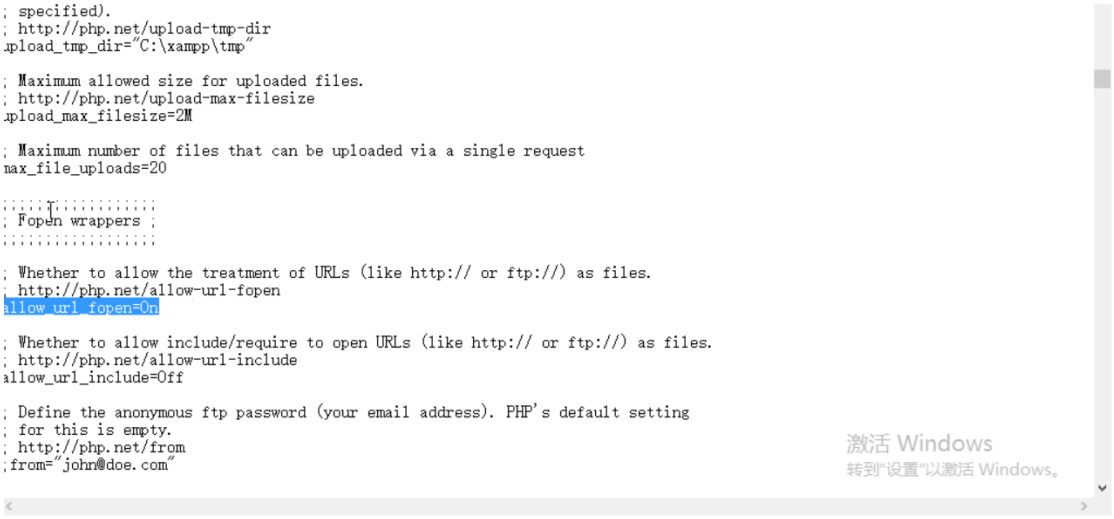

图 3

4. 双击打开桌面上的【XAMPP Control Panel】,开启 Apache 和 Mysql 服务，如图 4 所示。

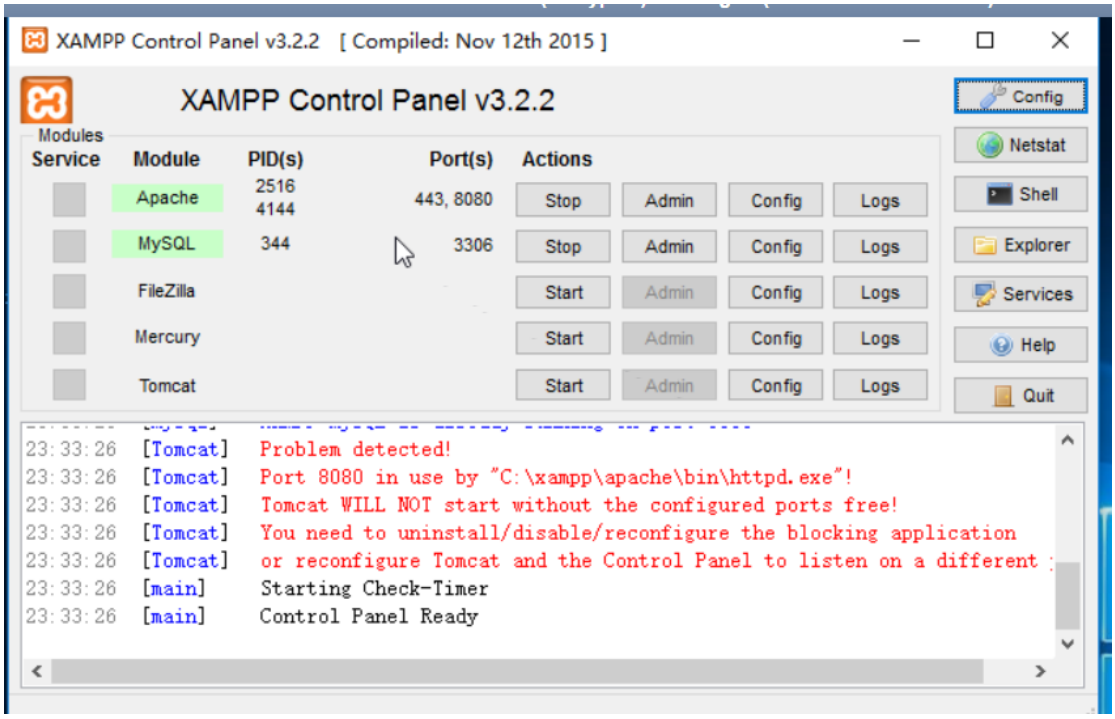

图 4

5. 打开【C:\Tools:\远程文件包含】找到 example_code 右键【复制】粘贴到【C:\xampp\htdocs】。。如图 5 所示

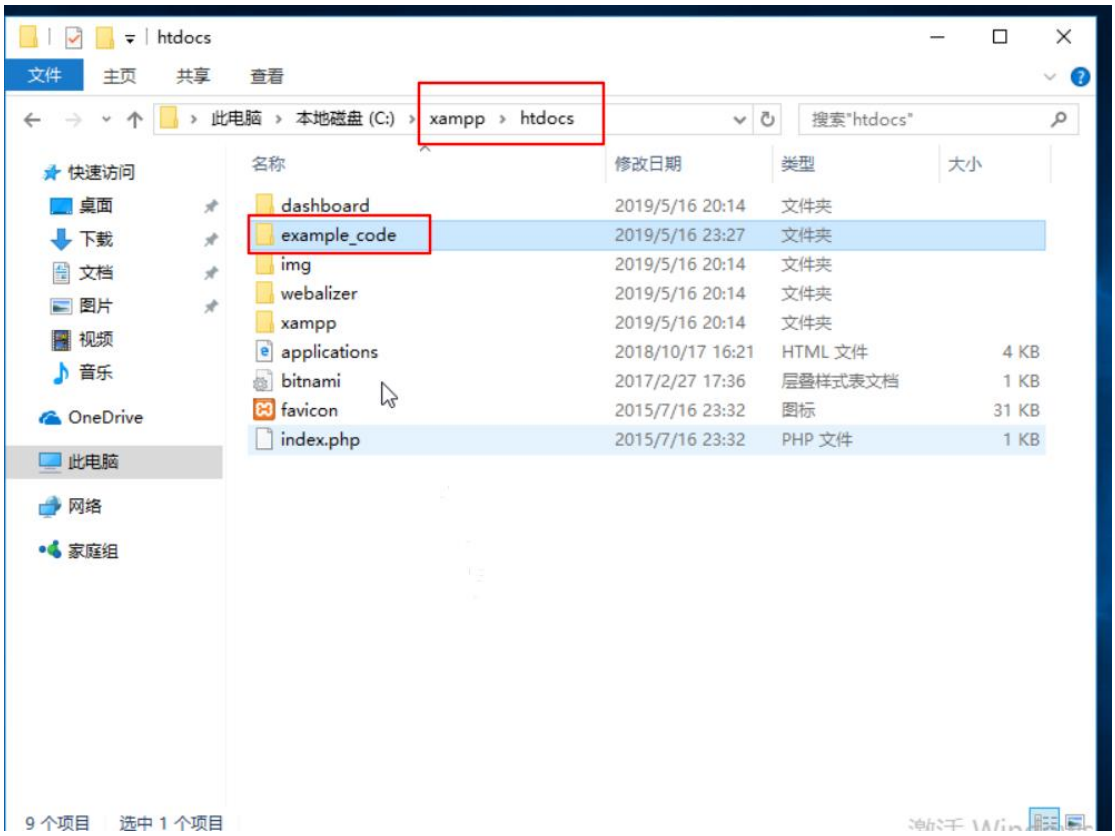

图 5

6. 打开浏览器，输入“http://localhost:8080/example_code/”如图 6 所示。

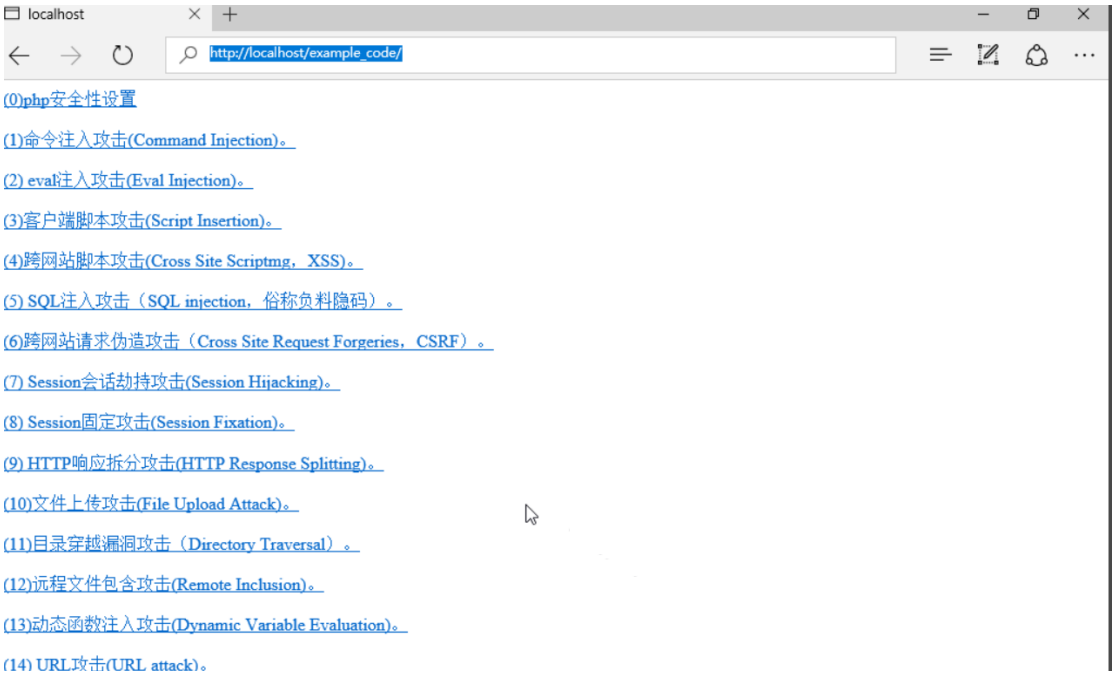

图 6

7. 鼠标单击打开【(12)远程文件包含攻击(Remote Inclusion)】。如图 7 所示。

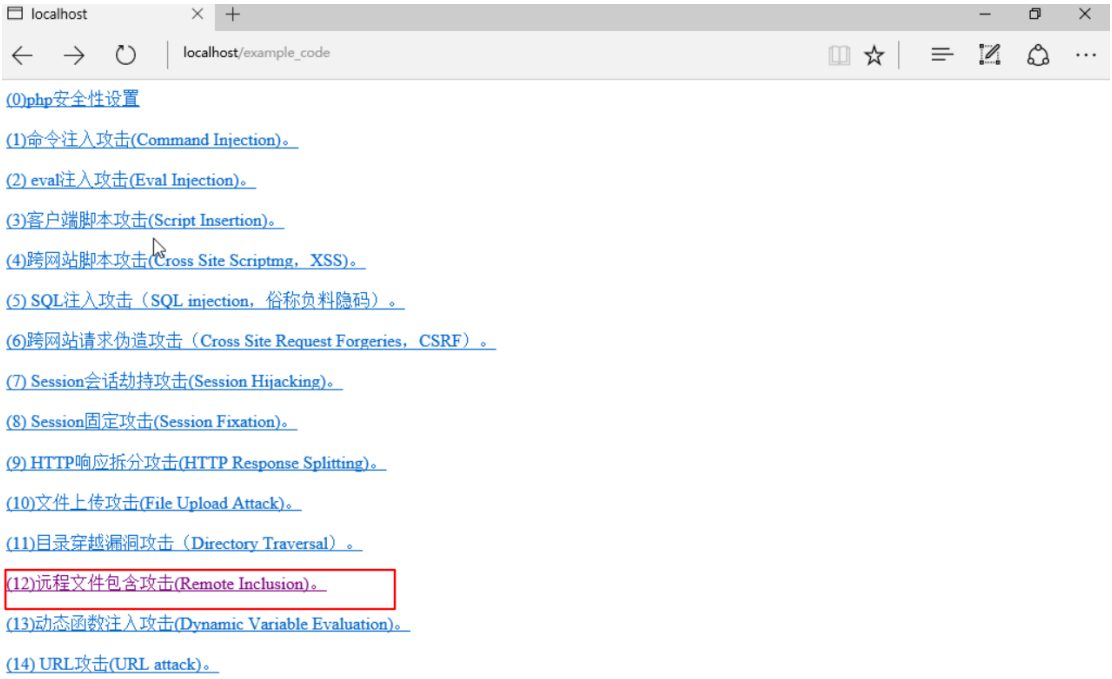

图 7

8. 如何攻击。点击【演示 1】，在演示 1 中，文件会读取一个文件名称，这个文件名称由 URL 参数 file 所提供。include 语句会引入 URL 参数 file 所提供的文件名称，readfile 函数会输出文件的内容，@用来屏蔽错误信息。如图 8 所示。

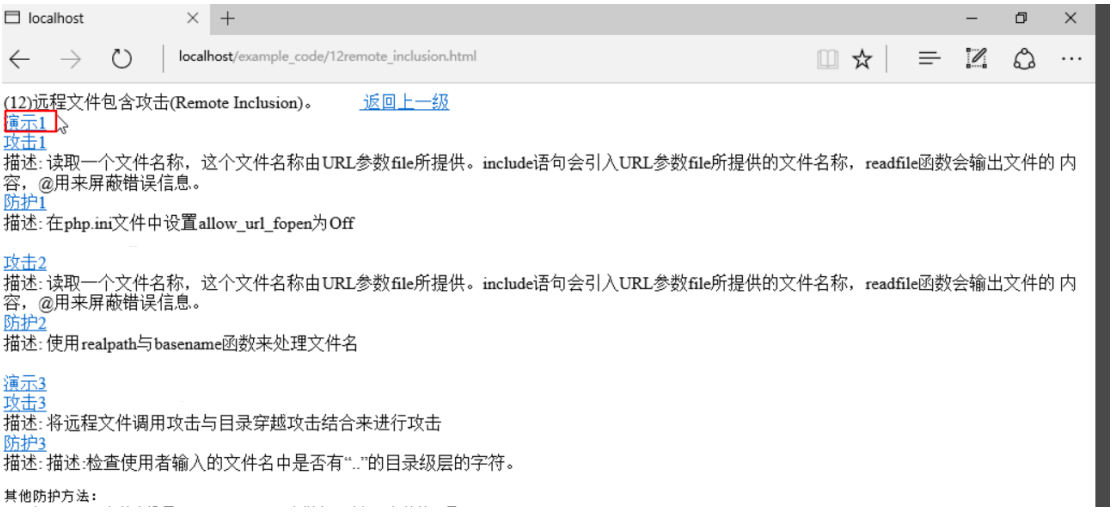

图 8

9. 返回点击【攻击 1】。如图 9 所示。

图 9

10..ex12-2-inclusion.php 是一个 PHP 文件，用来输出一个字符串。如图 10 所示。

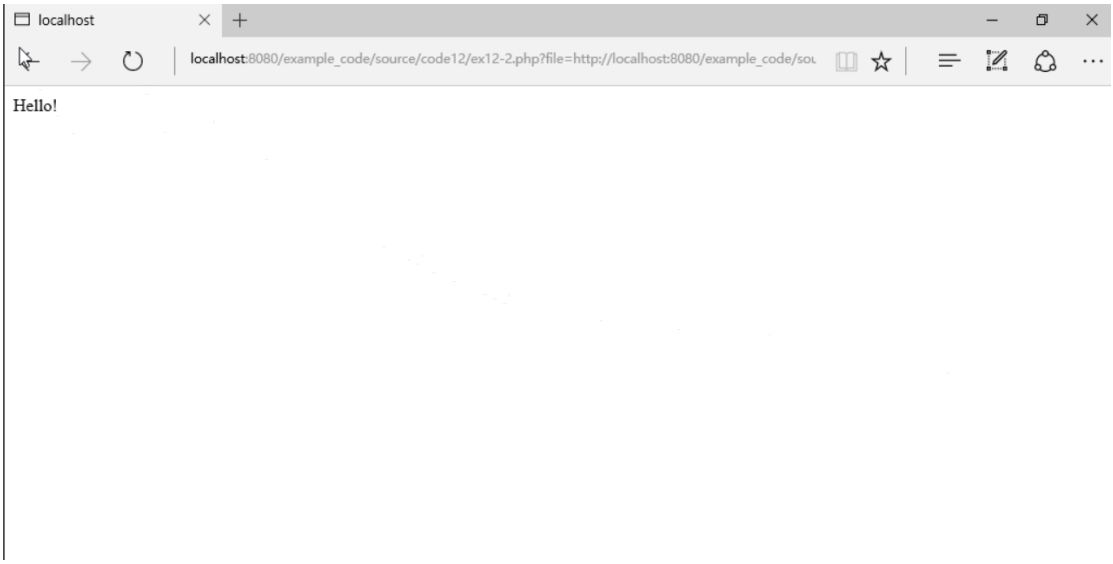

图 10

11.此时我们将看到浏览器地址栏中的值“http://localhost:8080/example_code/source/code12/ex12-2-inclusion.phpex12.php”引入并执行了 ex12-2-inclusion.php 文件，所以我们将 php.ini 里的 allow_url_fopen 设为 Off 后，打开【防护 1】，你将会看到 ex12-2.php 的 URL 参数就无法等于 ex12-2-inclusion.php。

12.打开 php.ini 文件。如图 11 所示。

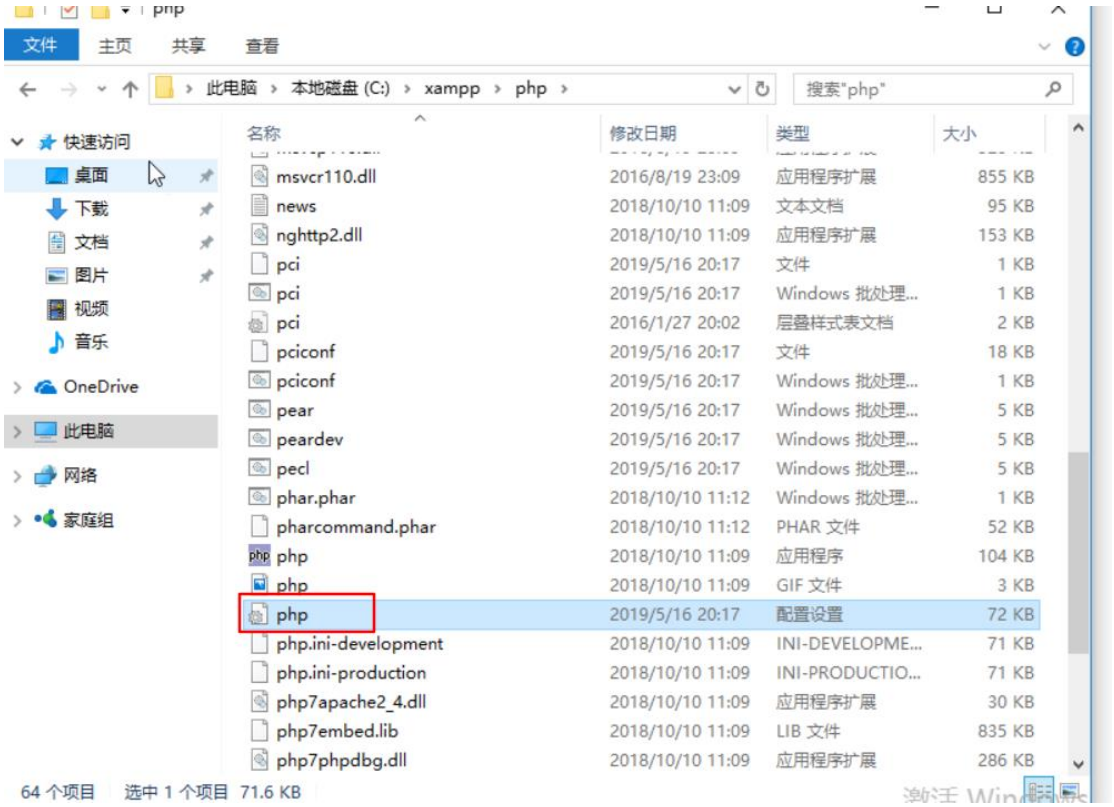

图 11

13. 将 allow_url_fopen 设置为 Off。如图 12 所示。

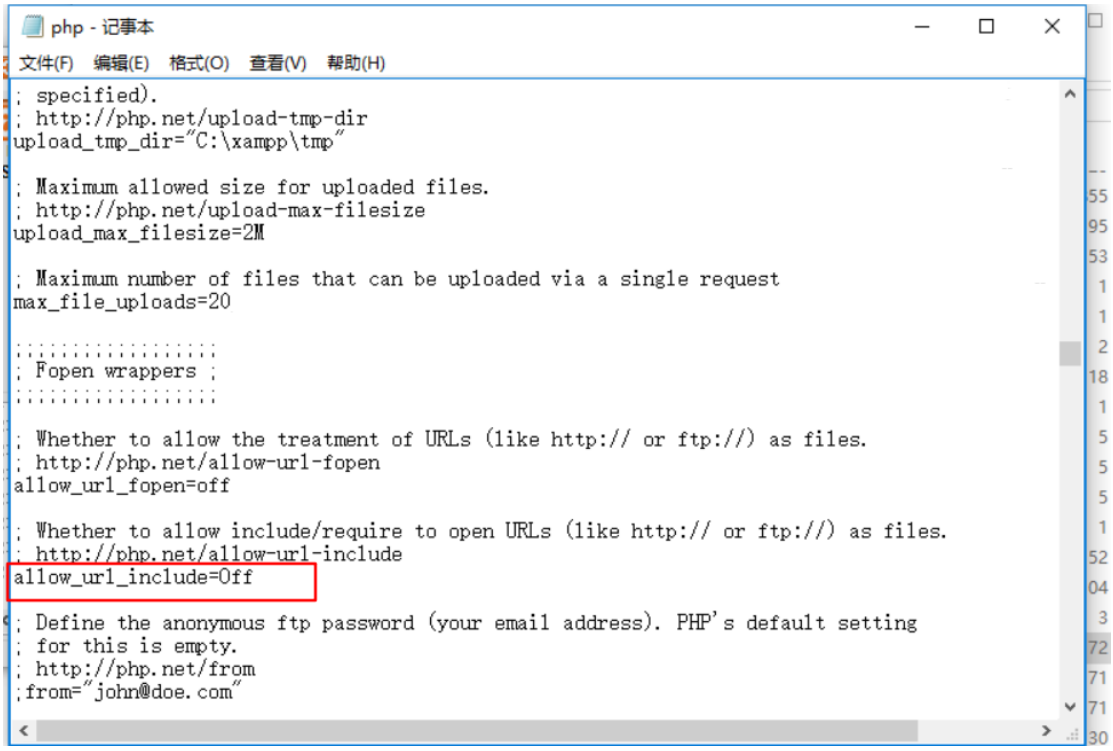

图 12

14. 重启 Apache 服务。如图 13 所示。

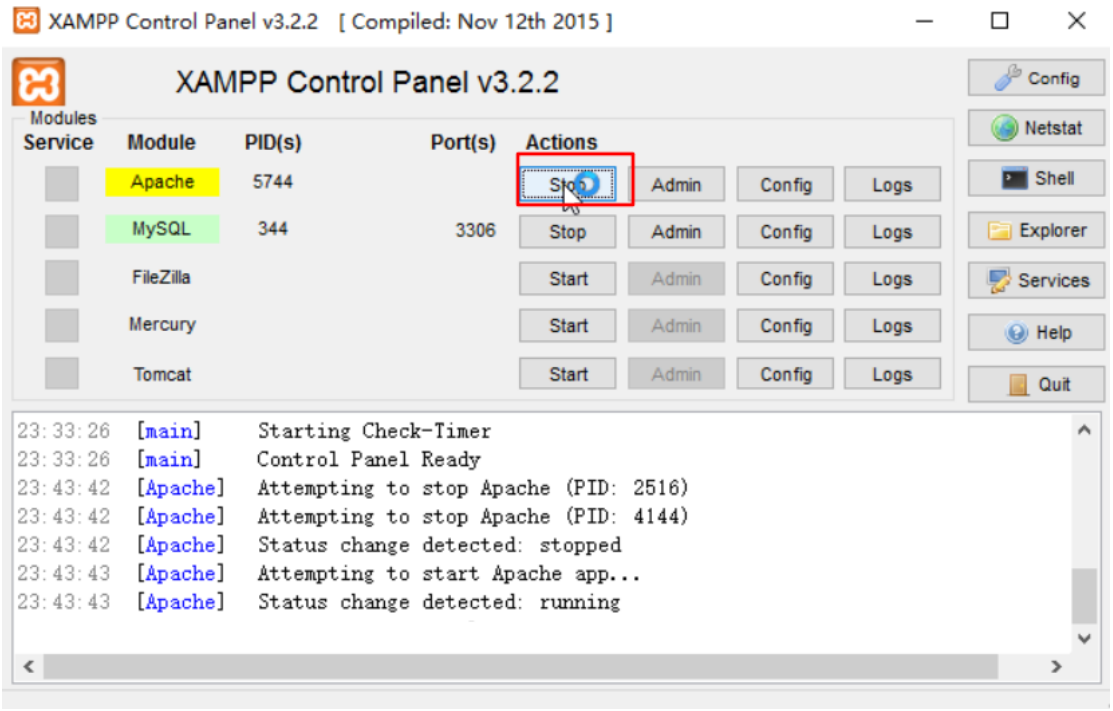

图 13

15. 点击【防护 1】。如图 14 所示。

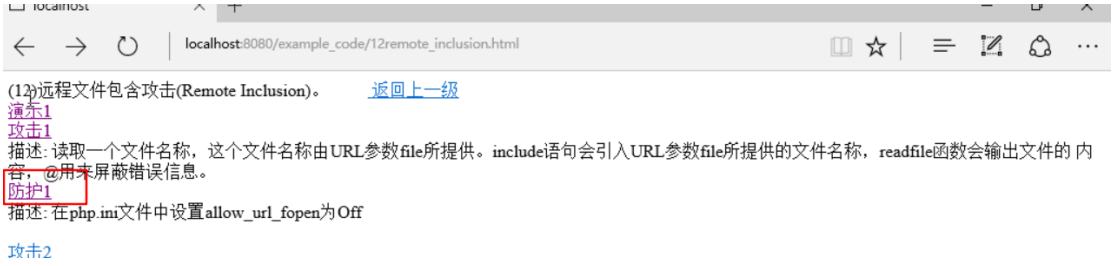

图 14

16. 可以看到页面出现 Warning，如图 15 所示。

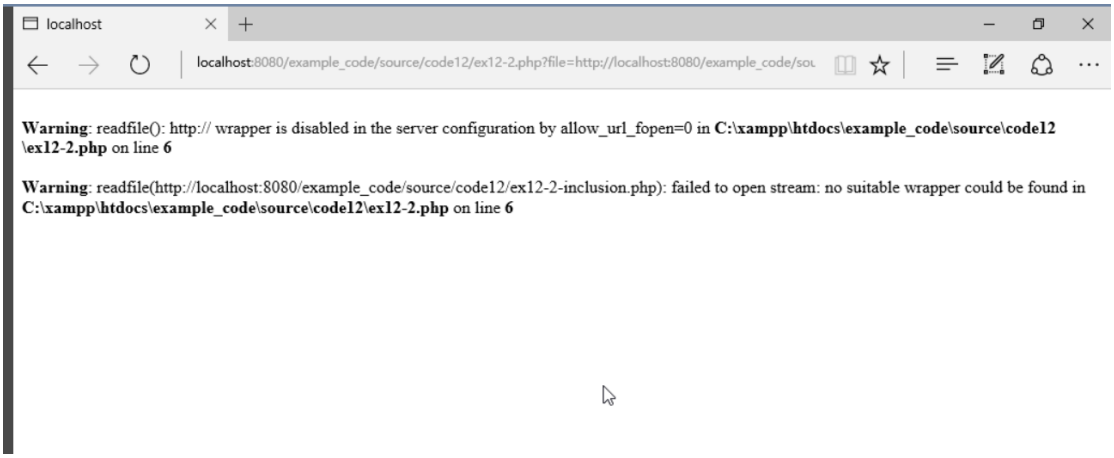

图 15

17. 进行攻击 2，将 php.ini 文件的 allow_url_fopen 修改为 on，然后重启 Apache 服务，这一步骤与上面修改 php.ini 一致。点击【攻击 2】。如图 16 所示

图 16

18. 此地址会引入到"显示 php 服务器的配置信息"页面并且执行", 如图 17 所示。

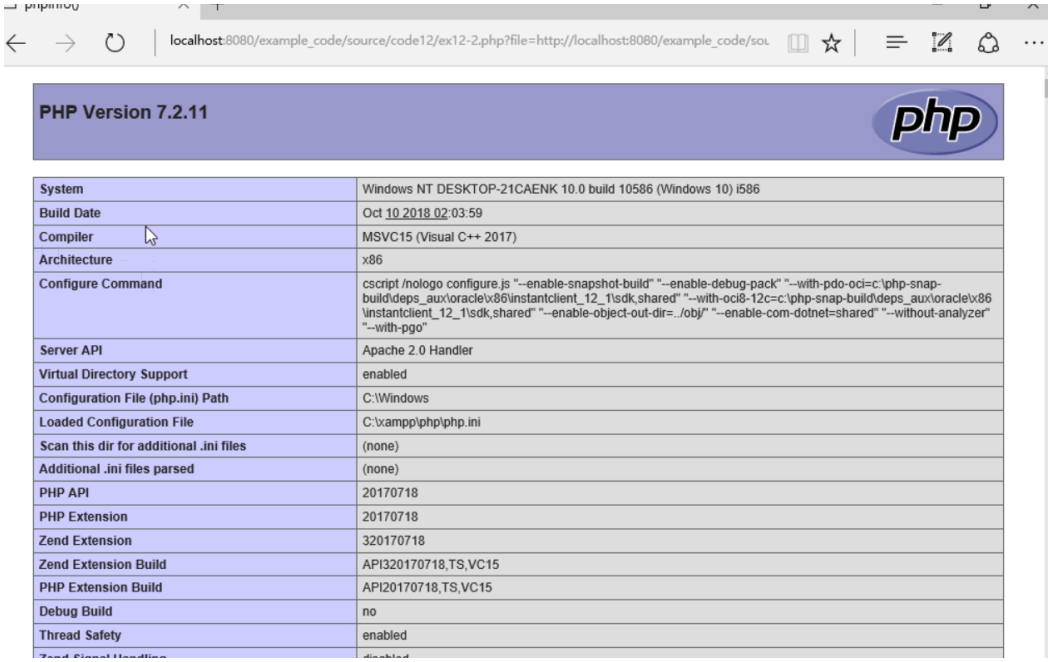

图 17

19. 用远程文件引入攻击。黑客要执行远程文件引入攻击时，他会在自己的网站内建立具有攻击性的 PHP 文件 index.php。然后使用“http://localhost:8080/example_code/source/code12/ex12-2.php?file=http://localhost:8080/example_code/source/code12/index.php”来让目标网站加载黑客的 attack.php 文件并且执行。

20. 与目录穿越结合攻击。黑客可以将远程文件调用攻击与目录穿越攻击结合，来产生更大的破坏力。请参考 ex12-3.php，程序源码在【C:\xampp\htdocs\example_code\source\code12】下。如图 18 所示。

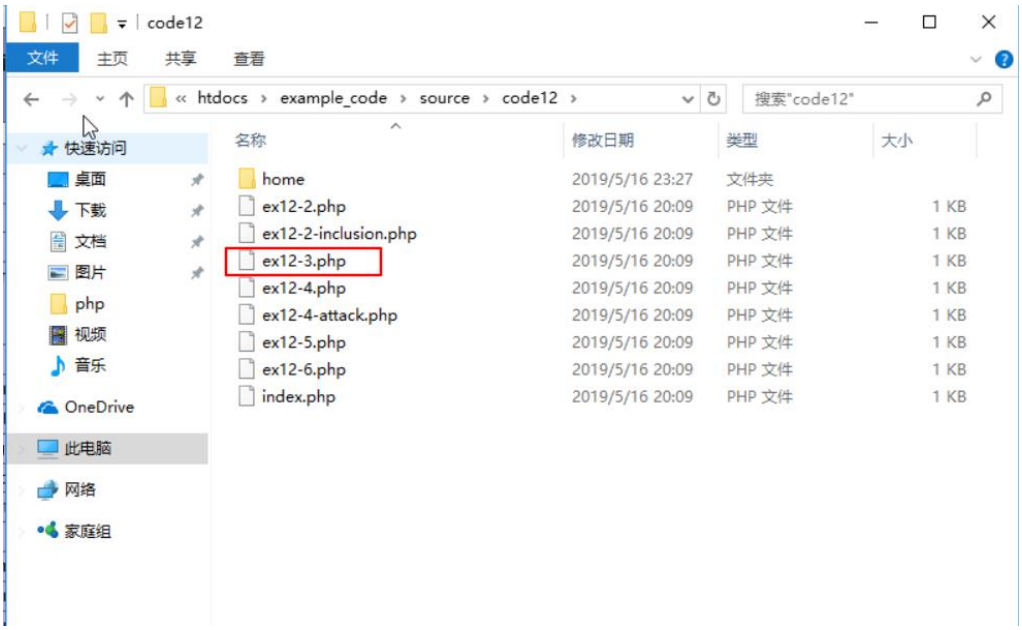

图 18

21. ex12-3.php 文件会读取一个文件名称，这个文件名称由 URL 参数 file 所提供。include 语句会加载 URL 参数 fle 所提供的文件名称，@用来屏蔽错误信息。

    将这个文件名称与目录字符串“home/users/”进行结合后，就是在 Web 应用程序所在的根目录下的文件路径。假设使用者输入的文件名称是 data.txt，其路径为：

    【C:\xampp\htdocs\example_code\source\code12\home\user\data.txt】

22. 点击“攻击 3”，如图 19 所示。

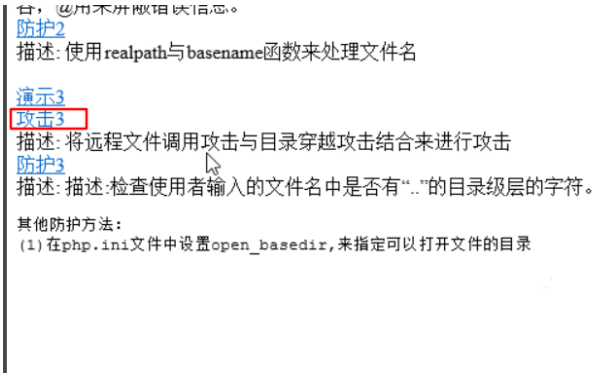

图 19

23. 结果如图 20 所示。

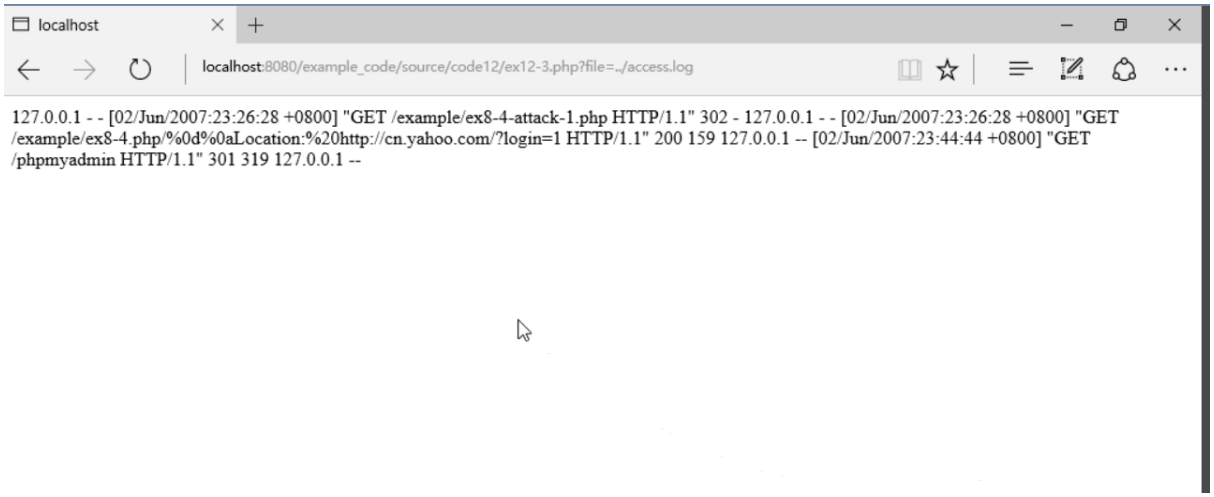

图 20

24. 在这个地址中“../access.log”使用 1 个“../”记号移动到上一层目录，所以会到达【C:/xampp/apache/logs】的目录。将【C:/xampp/apache/logs】与“../”记号后面的地址【/access.log】进行结合后，就是【C:/xampp/apache/logs/access.log】

    这表示要打开 Apache 服务器内的记录文件 access.log，这是将远程文件调用攻击与目录穿越攻击结合来进行攻击的例子。远程文件调用攻击所用的手法是```@include("home/users/".$_GET["file"])```,而目录穿越攻击所用的手法是```http://localhost:8080/example_code/source/code12/ex12-3.php?file=../access.log```两者结合来进行攻击，更能产生强大的破坏力。

25. 防范方法。检查输入的文件名。点击【防护 3】。如图 21 所示。

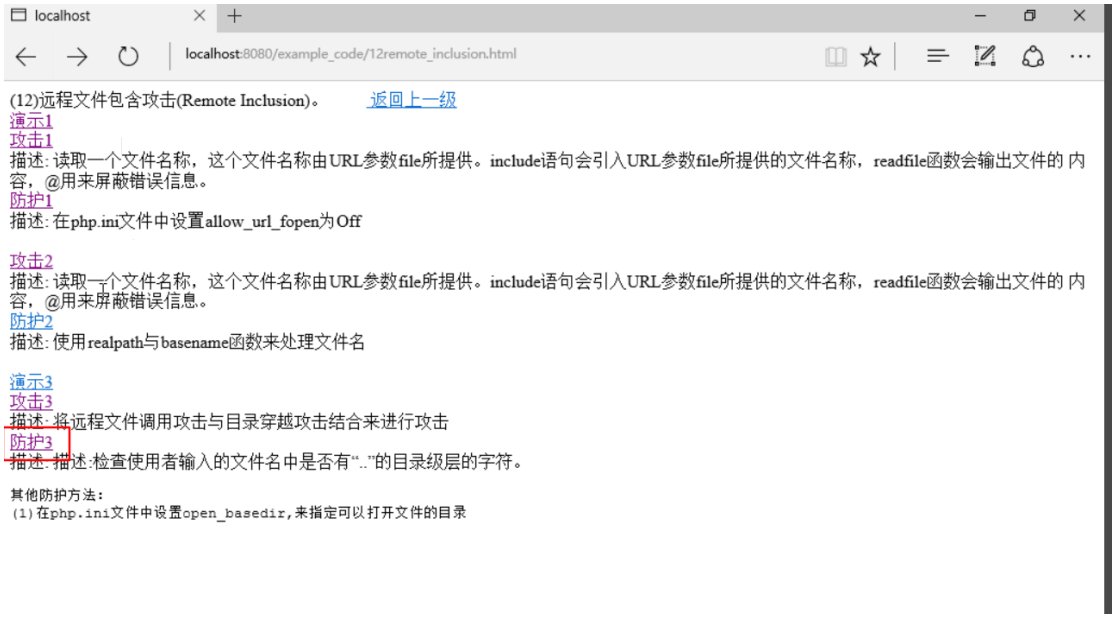

图 21

26. 通过检查使用者输入的文件名发现了目录穿越攻击。如图 22 所示。

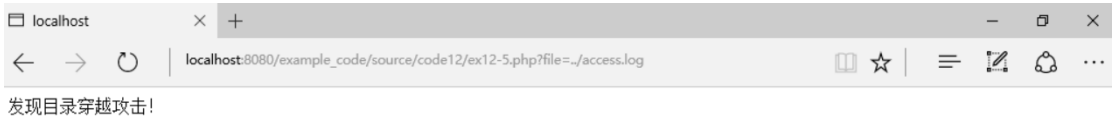

图 22

27. 实验只介绍了基本的防范方法希望学员自行阅读“其他防护方法”，并自行完成远程文件包含防范。

五【实验思考】

● 要想成功利用文件包含漏洞，需要满足哪些条件？。