SSRF Testing Methodology & Checklist

Eyad Islam El-Taher

February 28, 2026

Document Purpose

This document provides a comprehensive methodology and checklist for identifying, testing, and exploiting Server-Side Request Forgery (SSRF) vulnerabilities. It serves as a practical guide for penetration testers, security professionals, and developers to systematically assess SSRF risks in web applications.

Testing Methodology Overview

Phase 1: Reconnaissance & Attack Surface Identification

1. Identify SSRF Entry Points

- Features that fetch external resources (profile pictures, webhooks)
- URL parameters (`?url=`, `?path=`, `?dest=`)
- HTTP headers (`Referer`, `X-Forwarded-For`, `Origin`)
- File uploads that process external content (PDF generators, image processors)
- XML parsers (XXE to SSRF vectors)

2. Map Parameter Names

Common SSRF Parameters

`url`	`uri`	`path`	`dest`
`redirect`	`return`	`out`	`view`
`dir`	`show`	`file`	`document`
`location`	`src`	`source`	`image`
`hook`	`webhook`	`callback`	`notify`

3. Identify Internal Resources

- Cloud metadata endpoints: `169.254.169.254` (AWS, GCP, Azure)
- Internal IP ranges: `10.0.0.0/8`, `172.16.0.0/12`, `192.168.0.0/16`
- Localhost: `127.0.0.1`, `localhost`, `0.0.0.0`
- Common services: `redis`, `memcached`, `mongodb`, `elasticsearch`

Phase 2: Basic (In-Band) SSRF Testing

1. **Test Basic Localhost Access**
   - Submit `http://127.0.0.1`
   - Submit `http://localhost`
   - Submit `http://0.0.0.0`
   - Check for admin interfaces, internal files, or debug endpoints
2. **Test Internal Network Ranges**
   - Use Burp Intruder to fuzz internal IPs: `http://192.168.0.$1$:8080/admin`
   - Monitor response times and status codes
   - Look for `200 OK` responses indicating live services
3. **Test Alternative IP Representations**

IP Bypass Formats

Format	Example
Decimal	`http://2130706433/`
Octal	`http://017700000001/`
Hexadecimal	`http://0x7f000001/`
Shortened	`http://127.1/`
CIDR bypass	`http://127.127.127.127/`

Phase 3: Blind SSRF Detection

1. **Out-of-Band (OAST) Detection**
   - Set up Burp Collaborator or Interactsh
   - Inject unique domain in suspicious parameters
   - Monitor for DNS and HTTP interactions
2. **Collaborator Everywhere Workflow**
   (a) Install Collaborator Everywhere extension
   (b) Add target to scope
   (c) Browse application normally
   (d) Monitor Collaborator tab for callbacks
   (e) Identify which headers triggered interactions
3. **Response Time Analysis**
   - Compare response times between valid and invalid internal IPs
   - Longer responses may indicate live internal services
   - Timeouts may indicate blocked ports or firewalls
4. **Error Message Analysis**
   - Look for differences in error messages
   - Connection refused vs connection timeout
   - DNS resolution errors

Phase 4: Bypass Technique Testing

1. Blacklist Bypass Techniques

- Alternative IP representations (decimal, octal, hex)
- URL encoding and double encoding
- Case variation (`/ADMIN`, `/Admin`)
- DNS to localhost (`localtest.me` → `127.0.0.1`)
- Custom domain pointing to `127.0.0.1`

2. Whitelist Bypass Techniques

- Embedded credentials: `http://expected@evil-host`
- Fragment tricks: `http://evil-host#expected`
- DNS hierarchy: `http://expected.evil-host`
- Double encoding: `http://localhost:80%2523@expected.com`
- Open redirect chaining

3. DNS Rebinding Testing

- Use rebinding services: `lock.cmpxchg8b.com`
- Configure public IP + internal target IP
- Submit domain to SSRF endpoint
- Send multiple requests (10-20 attempts)
- Monitor for successful bypass

4. Protocol Smuggling

- Test `file://` for local file inclusion
- Test `gopher://` for attacking internal services
- Test `dict://` for service probing
- Test `ftp://` for internal FTP access

Phase 5: Exploitation & Impact Assessment

1. Cloud Metadata Exploitation

Cloud Metadata Endpoints

Cloud Provider	Metadata URL
AWS	`http://169.254.169.254/latest/meta-data/`
GCP	`http://169.254.169.254/computeMetadata/v1/`
Azure	`http://169.254.169.254/metadata/instance?api-version=2017`
DigitalOcean	`http://169.254.169.254/metadata/v1/`

2. Internal Service Exploitation

- Redis: `gopher://localhost:6379/_*2$4...`
- Memcached: `gopher://localhost:11211/_stats`
- Elasticsearch: `http://localhost:9200/_cat/indices`
- Internal admin panels: `http://192.168.1.1/admin`

3. Remote Code Execution Vectors

- Shellshock via headers in blind SSRF
- Vulnerable HTTP client libraries
- Internal service vulnerabilities (Redis RCE)
- PDF generator exploitation

SSRF Testing Checklist

Comprehensive SSRF Checklist

[1] Entry Point Identification

- URL parameters (dest, path, url, redirect)
- HTTP headers (Referer, X-Forwarded-For, Origin)
- File upload features (PDF, image, document processing)
- XML parsers (XXE to SSRF)
- API endpoints with external fetch capabilities
- Webhook configuration pages
- Profile picture URL import

[2] Basic Localhost Testing

- http://127.0.0.1
- http://localhost
- http://0.0.0.0
- http://[::1] (IPv6)
- http://127.0.0.1:8080 (different ports)
- http://127.0.0.1/admin
- http://127.0.0.1/secret
- http://127.0.0.1/internal

[3] Internal Network Probing

- 192.168.0.0/16 range scanning
- 10.0.0.0/8 range scanning
- 172.16.0.0/12 range scanning
- Common ports (22, 80, 443, 8080, 3306, 6379, 9200)
- Internal service discovery

[4] Cloud Metadata Testing

- AWS: http://169.254.169.254/latest/meta-data/
- AWS: http://169.254.169.254/latest/user-data/
- GCP: http://169.254.169.254/computeMetadata/v1/
- Azure: http://169.254.169.254/metadata/instance
- DigitalOcean: http://169.254.169.254/metadata/v1/
- Alibaba Cloud: http://100.100.100.200/latest/meta-data/

[5] IP Format Bypasses

- Decimal: 2130706433
- Octal: 017700000001
- Hex: 0x7f000001
- Short: 127.1
- CIDR variations: 127.127.127.127
- IPv6: [::1]
- IPv6 to IPv4 mapping: [::ffff:7f00:1]

Comprehensive SSRF Checklist

[1] DNS-Based Bypasses

- localhost domains: localtest.me, localh.st
- Custom domain pointing to 127.0.0.1
- Subdomain tricks: evil.com resolving to 127.0.0.1
- DNS rebinding attacks

[2] Encoding Bypasses

- URL encoding
- Double encoding
- Unicode encoding
- Mixed case (AdMiN)
- Null byte injection

[3] Protocol Testing

- file:///etc/passwd
- file:///c:/windows/win.ini
- gopher://localhost:6379/
- dict://localhost:11211/
- ftp://localhost:21/
- ldap://localhost:389/
- smb://localhost:445/

[4] Blind SSRF Detection

- Burp Collaborator injection
- Response time analysis
- Error message differences
- HTTP header injection (Referer, User-Agent)
- Collaborator Everywhere extension

[5] Advanced Exploitation

- Shellshock via headers
- Redis RCE via gopher

Prevention Checklist

SSRF Prevention Measures

1. **Allowlist Approach**
   - Whitelist allowed domains/IPs
   - Reject all by default
   - Strict protocol restriction (HTTP/HTTPS only)
2. **Network Isolation**
   - Firewall egress filtering
   - Block access to internal IP ranges
   - Disable cloud metadata access
   - Network segmentation
3. **URL Handling**
   - Resolve hostname once, validate IP
   - Connect directly to validated IP
   - Set Host header manually
   - Disable redirect following
4. **Input Validation**
   - Validate URL format
   - Reject private IP ranges
   - URL parsing with proper libraries
   - No blacklisting (easily bypassed)

— End of SSRF Testing Methodology & Checklist —