Engineer

SSRF

Angriff: Ein Server ruft eine vom Angreifer gewählte URL ab

Was es ist

SSRF steht für Server-Side Request Forgery. Bei diesem Angriff übergibt jemand einem Server eine URL, die der Server dann stellvertretend von seinem eigenen Standort im Netzwerk abruft. Die Common Weakness Enumeration führt dies als CWE-918 und beschreibt den Fehler anhand der ausgelassenen Prüfung: Der Code übernimmt eine URL von einer vorgelagerten Komponente, ruft ihr Ziel ab und stellt nie sicher, dass die Anfrage tatsächlich zum vorgesehenen Ziel geht. Der Angreifer ändert nichts am Servercode. Er liefert eine Adresse, und die Position des Servers im Netzwerk erledigt den Rest.

Um diese Position geht es. Ein Server im Rechenzentrum oder in einem Cloud-Konto erreicht gewöhnlich Dinge, die ein Besucher aus dem Internet nicht erreichen kann: ein an eine interne Adresse gebundenes Administrationspanel, einen Datenbankport, ein Monitoring-Dashboard oder einen Metadatendienst. Der CWE-Eintrag beschreibt die Folgen. Der Server wird zum Proxy des Angreifers, kann zum Scannen von Ports interner Hosts gebracht werden, und URL-Schemata wie `file://`, `gopher://` und `tftp://` können Dokumente auf dem Rechner selbst lesen oder dem Angreifer mehr Kontrolle über den Anfrageinhalt geben. CWE ordnet SSRF einer umfassenderen Schwachstelle unter: CWE-441, dem unbeabsichtigten Proxy oder „Confused Deputy“, einem privilegierten Baustein, der für jemanden ohne diese Rechte arbeitet.

Die Tür öffnet meist eine gewöhnliche Funktion. Das OWASP-Cheat-Sheet zur Vorbeugung nennt zwei typische Fälle: ein Avatarbild, das die Anwendung von einer vom Benutzer eingegebenen URL herunterlädt, sowie von Benutzern registrierte Webhook- oder Callback-URLs, die die Anwendung später aufrufen soll. Linkvorschauen und Schaltflächen zum Importieren von Inhalten über eine URL gehören ebenso dazu wie jede Funktion, bei der ein Server einer von jemand

anderem geschriebenen Adresse folgt. Der CWE-Titel schreibt den Begriff mit Großbuchstaben als Server-Side Request Forgery und nennt einen älteren Namen für dasselbe Problem: XSPA, kurz für Cross Site Port Attack.

Das Konzept für eine Entscheidung nutzen

Nutzen Sie „Was ist SSRF?“ für eine konkrete Entscheidung. Wir können helfen, das Konzept mit dem System in Verbindung zu bringen, das Sie entwickeln oder einkaufen. Konten, Passwörter und Berechtigungen werden so geordnet, dass alle genau den beruflich erforderlichen Zugriff haben und ihn am Tag ihres Ausscheidens verlieren.

Unsere Leistungen entdecken Kontakt aufnehmen →

Die Annahmen hinter dem Begriff prüfen

Nutzen Sie die Erklärung und die verlinkten Arbeiten, um Fragen für Ihr Projekt zu formulieren. Wir können die Annahmen gemeinsam mit Ihnen prüfen, bevor ein Werkzeug ausgewählt wird. 1.000 RBAC-Regeln für sichere Linux-Server — 1,000 Zugriffsregeln zum Schließen einer Sicherheitslücke

1.000 RBAC-Regeln für sichere Linux-Server Kontakt aufnehmen →

Warum es wichtig ist

OWASP nahm SSRF 2021 als A10 in seine Top 10 auf, nicht wegen der Fallzahl. Die Daten zeigten eine relativ geringe Häufigkeit, aber in der Community-Umfrage lag die Kategorie vorn, und die Bewertungen für Ausnutzbarkeit und Auswirkungen lagen über dem Durchschnitt. OWASP begründet das damit, dass URL-Abrufe zu einer alltäglichen Funktion von Webanwendungen geworden sind und Cloud-Dienste sowie komplexere Architekturen schwerere Folgen ermöglichen. In der Ausgabe 2025 hat SSRF keine eigene Kategorie mehr: CWE-918 gehört zu den hervorgehobenen Schwachstellen von A01:2025, Broken Access Control, an der Spitze der Liste. MITREs Rangliste weist in dieselbe Richtung. CWE-918 stand von 2021 bis 2025 in jedem Jahr in den CWE Top 25 Most Dangerous Software Weaknesses.

SSRF durchquert die Verteidigungslinien, auf die sich Teams besonders stark verlassen. OWASP betont, dass der Angriff selbst dann funktioniert, wenn die Anwendung hinter einer Firewall, einem VPN oder einer Zugriffskontrollliste im Netzwerk liegt: Die Anfrage kommt von innen. Die Beispielszenarien führen durch das, was hinter dieser Grenze liegt: interne Hosts und offene Ports anhand von Verbindungsergebnissen oder Verbindungszeiten ermitteln, lokale Dateien wie `file:///etc/passwd` lesen, interne Dienste wie `http://localhost:28017/` ansprechen und diese für weitere Schritte missbrauchen, bis hin zu entfernter Codeausführung oder Denial of Service.

Cloud-Metadaten machen aus einer lästigen Schwachstelle einen Einbruch. Laut OWASP stellen die meisten Cloud-Anbieter Instanzmetadaten unter `http://169.254.169.254/` bereit. Die Adresse liegt im Block 169.254.0.0/16, den das IANA-Register für Sonderzwecke als linklokal führt. Auf Amazon EC2 bezieht Software auf der Instanz dort auch die AWS-Zugangsdaten für die der Instanz zugewiesene Rolle. Ein Server, der jede URL für jeden abruft, lässt sich auch zum Abruf dieser Zugangsdaten auffordern.

Das Muster findet immer neue Einsatzorte. Zu den jüngeren Beispielen im CWE-Eintrag zählen ein LLM-Werkzeugsatz, der beliebige Bild-URLs abrief, eine Schwachstelle, die Angreifer im April 2026 bereits zum Portscannen nutzten (CVE-2026-33626), sowie ein LLM-Anwendungsframework, dessen URL-Abruf sich mit einem präparierten `Location` -Header auf lokale Adressen lenken ließ (CVE-2024-3095). Ein Agent, der für einen Benutzer im Web recherchiert, ist aus Sicht des Netzwerks ein Server, der von jemand anderem gewählte URLs abruft.

Wie es in der Praxis aussieht

Nehmen wir eine Chatanwendung unter `app.example.com` , die beim Einfügen eines Links eine Vorschaukarte anzeigt. Der Browser schickt die URL an den Server, und dieser ruft die Seite ab, um Titel und Bild auszulesen:

```
POST /api/link-preview HTTP/1.1
Host: app.example.com
Content-Type: application/json

{"url": "https://news.example.org/2026/10/harbour-reopens"}
```

So arbeitet die Funktion wie vorgesehen. Nun dieselbe Anfrage mit einer anderen URL, die die AWS-Dokumentation zum Auflisten von Instanzmetadaten verwendet:

```
POST /api/link-preview HTTP/1.1
Host: app.example.com
Content-Type: application/json

{"url": "http://169.254.169.254/latest/meta-data/"}
```

Läuft der Server auf EC2 und ist das ältere IMDSv1 noch erlaubt, sendet er die Anfrage aus der Instanz heraus. Der Metadatendienst antwortet, und alles, was der Vorschaucode zurückgibt, landet beim Angreifer. Nichts wurde injiziert, nichts ist abgestürzt. Der Server hat genau die vorgesehene Arbeit ausgeführt, nur für die falsche Adresse.

Eine Zielprüfung beginnt damit, wohin die Anfrage nach der Namensauflösung tatsächlich gehen würde, statt beim Text der URL. In Python verwendet das Standardbibliotheksmodul `ipaddress` die IANA-Register für Sonderzwecke, um Adressen als global oder nicht global einzuordnen. Dabei werden Adressen ausgeschlossen, die selbst zu Loopback-, privaten, linklokalen, gemeinsam genutzten oder Dokumentationsbereichen gehören. Die folgende Hilfsfunktion veranschaulicht diese Einordnung und ist keine vollständige SSRF-Abwehr. Eine Bereitstellung muss Anfragen außerdem auf geeignete Unicast-Ziele begrenzen und Adressübersetzung wie NAT64 berücksichtigen: In einem Netz mit übersetzendem Gateway kann eine als global eingestufte Adresse trotzdem ein gesperrtes eingebettetes IPv4-Ziel erreichen. Die Einordnungsprüfung der Hilfsfunktion lautet:

```
import ipaddress
import socket
from urllib.parse import urlsplit

ALLOWED_SCHEMES = {"http", "https"}


def resolve_public(url: str) -> tuple[str, int, list[str]]:
    parts = urlsplit(url)
    if parts.scheme not in ALLOWED_SCHEMES or not parts.hostname:
        raise ValueError("scheme or host not allowed")
    port = parts.port or (443 if parts.scheme == "https" else 80)
    infos = socket.getaddrinfo(parts.hostname, port, type=socket.SOCK_STREAM)
    addresses = sorted({info[4][0] for info in infos})
    for address in addresses:
        if not ipaddress.ip_address(address).is_global:
            raise ValueError(f"{address} is not a public address")
    return parts.hostname, port, addresses
```

Jede durch die Namensauflösung ermittelte Adresse muss die Prüfung bestehen, nicht nur die erste. Der Abruf verbindet sich dann mit einer dieser geprüften Adressen, sendet aber weiterhin den ursprünglichen Hostnamen im `Host` -Header und verwendet ihn zur TLS-Zertifikatsprüfung. Weiterleitungen sind abgeschaltet. Den HTTP-Client den Namen ein zweites Mal auflösen zu lassen

würde die Prüfung zunichtemachen. Diesen Aufbau beschreibt das OWASP-Cheat-Sheet für den schwierigen Fall der Webhooks: Die Benutzer wählen die Ziele, die keine Positivliste im Voraus aufzählen kann. Bei bekannten Zielen grenzt eine Positivliste genauer Hostnamen die Möglichkeiten weiter ein. Das Cheat-Sheet stellt aber klar, dass die aufgelösten Adressen dieselbe Prüfung trotzdem brauchen.

Anwendungsprüfungen sind eine Schicht. Der Bericht über E-Mail mit Ausweichversand und Zustellprotokollierung hält fest, dass der Versandweg über einen gegen SSRF geschützten Client mit einem Timeout von 30 Sekunden läuft. Hinter der Anwendung gilt weiterhin OWASPs Empfehlung für die Netzwerkschicht: Platzieren Sie den Code zum Abrufen entfernter Ressourcen in einem eigenen Netzwerksegment und verwenden Sie Firewallregeln, die Zugriffe standardmäßig verweigern. Eine ausgelassene Prüfung trifft dann auf eine geschlossene Tür.

Wo es schiefgeht

Sperrlisten und Mustersuche. Ein Filter, der URLs mit `127.0.0.1`, `localhost` oder `169.254.169.254` ablehnt, hat bereits verloren. POSIX-`inet_addr()` akzeptiert IPv4-Adressen mit einem bis vier Teilen, jeweils dezimal, oktal oder hexadezimal. Für jeden Parser, der diesen Regeln folgt, bezeichnen deshalb sowohl `2852039166` als auch `0xa9.0xfe.0xa9.0xfe` die Metadatenadresse. Der CWE-Eintrag nennt eine CVE, bei der oktale und hexadezimale Formen eine auf Dezimaldarstellung ausgelegte Prüfung umgingen und einen SSRF-Schutz aushebelten (CVE-2016-4029). OWASPs Top-10-Eintrag formuliert es deutlich: „Verwenden Sie zur Abwehr von SSRF weder eine Sperrliste noch einen regulären Ausdruck.“ Entscheiden Sie wie oben anhand der aufgelösten Adresse und erlauben Sie ausdrücklich, statt einzelne Fälle zu sperren.

Einmal prüfen und später verbinden. Den Hostnamen zu validieren und dann den HTTP-Client erneut auflösen zu lassen schafft eine Lücke. Bei DNS Rebinding liefert die Auflösung des vom Angreifer kontrollierten Hostnamens bei der Prüfung eine öffentliche Adresse, bei der Verbindung dagegen eine interne. Verbinden Sie sich mit genau der geprüften Adresse. Wenden Sie dieselbe Regel auch auf Wiederholungen und Ausweichverbindungen an, die das OWASP-Cheat-Sheet ausdrücklich nennt.

Weiterleitungen folgen. Die geprüfte URL antwortet mit einer Weiterleitung an `http://169.254.169.254/` oder eine `file://` -URL, und der Client folgt ihr ohne Nachfrage. Zu den CWE-Beispielen gehört eine Bibliothek für URL-Downloads, die Weiterleitungen zu `file://` - und `scp://` -URLs verfolgte (CVE-2009-0037). Schalten Sie Weiterleitungen ab oder behandeln Sie jeden Schritt als neue URL, die die vollständige Prüfung erneut durchläuft.

Jedes Schema zulassen. Ein Client, der auch `file` , `gopher` oder `dict` versteht, gibt einem Angreifer lokale Dateien oder wesentlich mehr Kontrolle über das, was einen internen Dienst erreicht, als eine HTTP-Anfrage erlaubt. Das OWASP-Cheat-Sheet nennt diese Schemata unter denen, die eine zweite Anfrage verwenden kann. Erlauben Sie nur `http` und `https` , auf den von der Funktion benötigten Ports.

Zwei Parser, zwei Hosts. Das OWASP-Cheat-Sheet zeigt, wie eine Zeichenfolge zwei verschiedene Hosts bezeichnen kann. In `http://good.example\@bad.example` behandelt ein Parser nach dem WHATWG URL Standard den Rückwärtsschrägstrich wie einen normalen Schrägstrich. Dort beginnt dann der Pfad, und der Host ist `good.example` . Pythons `urlsplit` deutet hingegen alles vor dem letzten `@` als Benutzerinformation und liefert `bad.example` . Prüft eine Komponente und ruft eine andere ab, können sie das Ziel unterschiedlich verstehen. Parsen Sie einmal und lehnen Sie jede URL ab, deren Host nicht von allen durchlaufenen Parsern gleich gelesen wird.

Die ungefilterte Antwort zurückgeben. Ein Vorschau-Endpunkt, der den abgerufenen Inhalt an den Aufrufer weiterreicht, macht aus einem Angriff, der anhand von Fehlern und Laufzeiten nur auf interne Inhalte schließen konnte, einen lesenden Zugriff. Geben Sie nur zurück, was die Funktion braucht, etwa einen Titel und einen Bildverweis, nie die ungefilterte Antwort, wie OWASPs Liste empfiehlt.

IMDSv1 eingeschaltet lassen. Laut AWS-Dokumentation können standardmäßig beide Versionen des Instanzmetadatendienstes verwendet werden. IMDSv2 verlangt zunächst ein Sitzungstoken, das mit `PUT` an `/latest/api/token` angefordert und anschließend im Header `X-aws-ec2-metadata-token` gesendet wird. Sobald IMDSv2 vorgeschrieben ist (`httpTokens=required`), funktioniert IMDSv1 nicht mehr. Das OWASP-Cheat-Sheet beschreibt IMDSv2 als Defense-in-Depth-Maßnahme, die einige SSRF-Fälle abschwächt, nicht alle. Schreiben Sie IMDSv2 für jede Instanz vor und beheben Sie trotzdem den Anwendungsfehler.

Ein flaches Netzwerk. Kann der abrufende Dienst die Datenbank, das Administrationspanel und den Metadatendienst erreichen, wirkt sich jede ausgelassene Prüfung umfassend aus. Wenden Sie OWASPs Netzwerkkontrollen an: Segmentierung, Firewallregeln, die internen Verkehr standardmäßig verweigern, und Protokolle zugelassener sowie blockierter Datenströme, damit eine Erkundung sichtbar wird.

Weiterführende Lektüre

- CWE-918: Server-Side Request Forgery (SSRF) — der Schwachstelleneintrag mit Folgen, der übergeordneten Confused-Deputy-Kategorie und beobachteten CVEs.
- OWASP Top 10:2021: A10 Server Side Request Forgery (SSRF) — warum die Kategorie aufgenommen wurde, Kontrollen auf Netzwerk- und Anwendungsebene und vier Angriffsszenarien.
- OWASP Cheat Sheet Series: Server Side Request Forgery Prevention — Positivlisten, Verbindungen nur zu geprüften Adressen, abweichende Parser und der Webhook-Fall.
- OWASP Top 10:2025: A01 Broken Access Control — die Einordnung von CWE-918 in der Ausgabe 2025.
- Amazon EC2: Access instance metadata for an EC2 instance — IMDSv1 und IMDSv2, die Tokenanforderung und das Erkennen einer vorgeschriebenen IMDSv2-Nutzung.
- Python documentation: ipaddress — `is_global` und die zugrunde liegenden IANA-Register.

Glossar

Für Ihre tatsächlichen Rahmenbedingungen auswählen

Beschreiben Sie den Bereitstellungsweg, die Betriebsrisiken und die Bereitschaftsverantwortlichen. Wir können eine Änderung mit Prüfungen, einem Wiederherstellungsweg und klaren Zuständigkeiten abgrenzen.

Preise ansehen Kontakt aufnehmen →

Die Begriffe praktisch nutzbar machen

Fragen Sie, wie „Was ist SSRF?“ auf Ihre Organisation zutrifft. Ein konkretes Beispiel aus Ihrem bestehenden System reicht aus, um das Gespräch zu beginnen.

Unsere Leistungen entdecken

welcome@engineer.company +45 50 10 61 84 Kopenhagen, Dänemark

Nehmen Sie diese Seite mit

Kontaktkarte

Mit Sorgfalt und Liebe gestaltet von D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Copyright © 2025 – heute · Engineer ApS · Alle Rechte vorbehalten.

Gedruckt von https://engineer.company/de/glossary/ssrf/