Engineer

SSRF

et angrep som får en server til å hente angriperens URL

Hva det er

SSRF, kort for server-side request forgery, er angrepet der noen gir en server en URL, og serveren henter den på deres vegne, fra det stedet i nettverket der serveren tilfeldigvis står. Common Weakness Enumeration fører det som CWE-918 og beskriver feilen ut fra det koden lar være å gjøre: den tar imot en URL fra en komponent lenger opp i kjeden, henter det URL-en peker på, og forsikrer seg aldri om at forespørselen går til det målet den var ment for. Angriperen endrer ikke en linje i serverens kode. Angriperen leverer en adresse, og serverens plass i nettverket gjør resten.

Den plassen er hele poenget. En server i et datasenter eller en skykonto når som regel ting en besøkende på internett ikke når: et adminpanel bundet til en intern adresse, en databaseport, et overvåkingsdashbord, en metadatatjeneste. CWE-oppføringen sier rett ut hva som følger. Serveren blir proxy for angriperen, den kan settes til å skanne porter på verter i det interne nettverket, og URL-er som begynner med `file://`, `gopher://` eller `tftp://`, kan lese dokumenter på selve maskinen eller gi angriperen mer kontroll over hva forespørselen inneholder. CWE plasserer SSRF under en bredere svakhet, CWE-441, den utilsiktede proxyen eller «confused deputy»: en komponent med rettigheter som gjør arbeid for noen som ikke har dem.

Det som åpner døren, er som regel en helt vanlig funksjon. OWASPs jukselapp om forebygging (cheat sheet) nevner to typiske tilfeller: et profilbilde som applikasjonen laster ned fra en URL brukeren har skrevet inn, og webhook- eller callback-URL-er som brukerne registrerer slik at applikasjonen kan kalle dem senere. Forhåndsvisninger av lenker og knapper for import fra URL hører til samme familie, det samme gjør enhver funksjon der serveren følger en adresse som noen andre har skrevet. CWE staver selv betegnelsen Server-Side Request Forgery med store forbokstaver, og oppføringen nevner et eldre navn på det samme, XSPA, kort for Cross Site Port Attack.

Bruk begrepet i en beslutning

Bruk «Hva er SSRF?» i en konkret beslutning. Vi kan knytte begrepet til systemet dere bygger eller kjøper. Kontoer, passord og tilganger ordnet slik at folk har nøyaktig den tilgangen jobben krever, og de som slutter, mister den samme dag.

Se tjenestene våre Ta kontakt →

Undersøk antakelsene bak ordet

Bruk forklaringen og det tilknyttede arbeidet til å formulere spørsmål for prosjektet. Vi kan undersøke antakelsene sammen før et verktøy velges. Skrev 1 000 RBAC-regler for å sikre Linux-servere — 1,000 tilgangsregler skrevet for å tette et sikkerhetshull

Skrev 1 000 RBAC-regler for å sikre Linux-servere Ta kontakt →

Hvorfor det er viktig

OWASP tok SSRF med på Topp 10 i 2021 som A10, og ikke på grunn av mengden. Dataene viste en relativt lav forekomst, men kategorien var førstevalget i undersøkelsen OWASP gjorde blant fellesskapet, og vurderingene for utnyttelse og konsekvens lå over gjennomsnittet. OWASP forklarer det med at henting av URL-er er blitt en rutinefunksjon i webapplikasjoner, og at skytjenester og mer komplekse arkitekturer gjør konsekvensene alvorligere. I 2025-utgaven har SSRF ikke lenger en egen kategori: CWE-918 står blant de nevneverdige svakhetene under A01:2025, Broken Access Control, kategorien som ligger øverst på listen. MITREs rangering peker samme vei, med CWE-918 på CWE Top 25 Most Dangerous Software Weaknesses hvert år fra 2021 til 2025.

Grunnen er at SSRF går rett gjennom de forsvarene team lener seg mest på. OWASP understreker at angrepet virker selv når applikasjonen står bak en brannmur, en VPN eller en tilgangskontrolliste i nettverket, fordi forespørselen går ut innenfra. OWASPs eksempelscenarier er en omvisning i det som skjuler seg bak den perimeteren: å kartlegge interne verter og åpne porter ut fra om tilkoblinger lykkes eller hvor lang tid de tar, å lese lokale filer som `file:///etc/passwd`, å snakke med interne tjenester som `http://localhost:28017/` og å misbruke de tjenestene for å komme videre, helt til fjernkjøring av kode eller tjenestenekt.

Sky-metadata er scenariet som gjør en irritasjon til et innbrudd. OWASP påpeker at de fleste skyleverandører har instansens metadata på `http://169.254.169.254/`, en adresse i blokken 169.254.0.0/16 som IANAs register over adresser til spesielle formål oppfører som link-local. På Amazon EC2 er instansens metadatatjeneste på den adressen også stedet der programvare på instansen henter AWS-legitimasjon for rollen som er knyttet til den. En server som henter hvilken som helst URL for hvem som helst, kan bli bedt om å hente den også.

Mønsteret dukker stadig opp nye steder. Blant de nyeste eksemplene i CWE-oppføringen er et LLM-verktøysett som hentet vilkårlige bilde-URL-er og ble aktivt utnyttet i april 2026 til portskanning (CVE-2026-33626), og et rammeverk for LLM-applikasjoner der URL-henteren kunne styres mot lokale adresser med en konstruert `Location` -header (CVE-2024-3095). En agent som leter og leser på nettet for en bruker, er sett fra nettverkets side bare en server som henter URL-er noen andre har valgt.

Slik ser det ut i praksis

Ta en chatteapplikasjon på `app.example.com` som viser et forhåndsvisningskort når noen limer inn en lenke. Nettleseren sender URL-en til serveren, og serveren henter siden for å lese tittelen og bildet:

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

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

Slik virker funksjonen etter hensikten. Se så den samme forespørselen med en annen URL, den som AWS-dokumentasjonen bruker for å liste instansens metadata:

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

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

Hvis serveren kjører på EC2 og den eldre IMDSv1 fortsatt er tillatt, sender den forespørselen innenfra instansen, metadatatjenesten svarer, og det forhåndsvisningskoden sender tilbake, havner hos angriperen. Ingenting ble injisert, og ingenting krasjet. Serveren gjorde akkurat det den var bygd for, bare for feil adresse.

En kontroll av måladressen begynner med hvor forespørselen faktisk ville havnet etter at navnet er slått opp, ikke med teksten i URL-en. I Python bruker standardbibliotekets `ipaddress` -modul IANAs registre over adresser til spesielle

formål til å klassifisere om en adresse er global. Det utelukker adresser som direkte tilhører loopback-, private, link-local-, delte eller dokumentasjonsområder. Hjelpefunksjonen nedenfor illustrerer denne klassifiseringen, ikke et fullstendig vern mot SSRF. En installasjon må også begrense forespørsler til tillatte unicastmål og ta høyde for adresseoversetting som NAT64: på et nettverk med en oversettende gateway kan en adresse som er klassifisert som global, fortsatt nå et blokkert, innebygd IPv4-mål. Hjelpefunksjonens klassifiseringstest er:

```
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
```

Hver eneste adresse navnet slås opp til, må bestå, ikke bare den første. Selve hentingen kobler seg deretter til en av de godkjente adressene, men sender fortsatt det opprinnelige vertsnavnet i `Host`-headeren og bruker det til å kontrollere TLS-sertifikatet, og omdirigeringer er slått av. Lar man HTTP-klienten slå opp navnet en gang til, er kontrollen bortkastet. Det er den formen OWASPs jukselapp beskriver for det vanskelige tilfellet, webhooks, der brukerne selv velger målene og ingen liste over tillatte adresser kan nevne dem på forhånd. Der målene er kjent, snevrer en liste over eksakte vertsnavn inn feltet ytterligere, men jukselappen er klar på at de oppslåtte adressene fortsatt må gjennom den samme testen.

Kontrollene i applikasjonen er ett lag. Beskrivelsen vår av e-post med failover og leveringslogging nevner at selve sendeveien går gjennom en SSRF-beskyttet klient med et tidsavbrudd på 30 sekunder. Bak applikasjonen gjelder OWASPs råd

for nettverkslaget fortsatt: legg koden som henter eksterne ressurser, i sitt eget nettverkssegment, og gi den brannmurregler som avviser alt som ikke er uttrykkelig tillatt, slik at en oversett kontroll møter en lukket dør.

Hvor det går galt

Blokkeringslister og mønstergjenkjenning. Et filter som avviser URL-er med `127.0.0.1`, `localhost` eller `169.254.169.254`, har allerede tapt. POSIX-funksjonen `inet_addr()` godtar en IPv4-adresse i én til fire deler, hver skrevet desimalt, oktalt eller heksadesimalt, så for enhver parser som følger de reglene, er både `2852039166` og `0xa9.0xfe.0xa9.0xfe` metadataadressen. CWE-oppføringen nevner en CVE der oktale og heksadesimale former slapp forbi en kontroll skrevet for desimaltall og satte en SSRF-beskyttelse ut av spill (CVE-2016-4029). OWASPs Topp 10 sier det rett ut: SSRF skal ikke motvirkes med en blokkeringsliste eller et regulært uttrykk. Avgjør det på den oppslåtte adressen, som over, og tillat heller enn å forby.

Kontroll nå, tilkobling senere. Å validere vertsnavnet og så la HTTP-klienten slå det opp på nytt etterlater et hull. Med DNS rebinding svarer angriperens navn med en offentlig adresse ved kontrollen og en intern ved tilkoblingen. Koble til nøyaktig den adressen du validerte, og bruk samme policy for nye forsøk og reserveforbindelser, som OWASPs jukselapp nevner ved navn.

Omdirigeringer som blir fulgt. Den godkjente URL-en svarer med en omdirigering til `http://169.254.169.254/` eller til en `file://`-URL, og klienten følger den uten å spørre. Blant eksemplene i CWE-oppføringen er et bibliotek for nedlasting av URL-er som fulgte omdirigeringer til `file://`- og `scp://`-URL-er (CVE-2009-0037). Slå av omdirigeringer, eller behandle hvert hopp som en ny URL som må gjennom hele kontrollen igjen.

Klienten snakker mer enn HTTP. Forstår den også `file`, `gopher` eller `dict`, kan en angriper lese lokale filer eller få langt mer kontroll over hva som når fram til en intern tjeneste, enn en HTTP-forespørsel gir. OWASPs jukselapp nevner nettopp disse blant URL-typene en ny forespørsel kan bruke. Tillat bare `http` og `https`, og bare portene funksjonen trenger.

To parsere, to verter. OWASPs jukselapp viser hvordan én streng kan peke på to ulike verter. I `http://good.example\@bad.example` leser en parser som følger WHATWGs URL-standard, omvendt skråstrek som baneskille og verten som `good.example`, mens Pythons `urlsplit` tar alt før den siste `@` som

brukeropplysninger og returnerer `bad.example`. Når én komponent validerer og en annen henter, kan de være uenige om hvor forespørselen går. Parse URL-en én gang, og avvis enhver URL der ikke alle parserne den passerer, leser verten likt.

Råsvaret sendes tilbake. Et endepunkt for forhåndsvisninger som sender det hentede innholdet tilbake til den som kalte det, gjør et angrep som bare kunne gjette seg til hva som finnes innenfor, ut fra feil og svartider, til et som kan lese det. Returner bare det funksjonen trenger, for eksempel en tittel og en henvisning til et bilde, og aldri råsvaret, slik OWASPs liste råder til.

IMDSv1 står fortsatt på. AWS-dokumentasjonen sier at begge versjonene av instansens metadatatjeneste kan brukes som standard. IMDSv2 krever først et sesjonstoken, som hentes med en `PUT` til `/latest/api/token` og deretter sendes i headeren `X-aws-ec2-metadata-token`, og når IMDSv2 er påkrevd (`httpTokens=required`), slutter IMDSv1 å virke. OWASPs jukselapp kaller IMDSv2 et ekstra forsvarslag som demper noen tilfeller av SSRF, ikke alle. Krev IMDSv2 på alle instanser, og rett applikasjonen likevel.

Et flatt nettverk. Når tjenesten som henter URL-er, kan nå databasen, adminpanelet og metadatatjenesten, er hver oversett kontroll fatal. Bruk OWASPs kontroller på nettverkslaget: segmentering, brannmurregler som i utgangspunktet avviser intern trafikk, og logging av tillatte og blokkerte forbindelser, slik at et søk etter åpninger blir synlig.

Les mer

- CWE-918: Server-Side Request Forgery (SSRF) — svakhetsoppføringen: konsekvensene, «confused deputy»-forelderen og de observerte CVE-ene.
- OWASP Top 10:2021: A10 Server Side Request Forgery (SSRF) — hvorfor kategorien kom med, kontroller på nettverks- og applikasjonslaget og fire angrepsscenarier.
- OWASP Cheat Sheet Series: Server Side Request Forgery Prevention — lister over tillatte adresser, tilkobling bare til validerte adresser, uenige parsere og webhook-tilfellet.
- OWASP Top 10:2025: A01 Broken Access Control — hvor CWE-918 står i 2025-utgaven.
- Amazon EC2: Access instance metadata for an EC2 instance — IMDSv1 og IMDSv2, tokenforespørselen og hvordan man ser om IMDSv2 er påkrevd.

- Python documentation: ipaddress — `is_global` og IANA-registrene den følger.

Ordliste

Velg ut fra deres faktiske rammer

Ta med utrullingsløpet, driftsrisikoene og dem som har vakt. Vi kan avgrense en endring med kontroller, en vei tilbake og tydelig ansvar.

Se hva det koster Ta kontakt →

Gjør begrepene praktiske

Spør hvordan «Hva er SSRF?» gjelder i organisasjonen deres. Et konkret eksempel fra dagens system er nok til å starte samtalen.

Se tjenestene våre

welcome@engineer.company +45 50 10 61 84 København, Danmark

Ta med deg denne siden

Visittkort

Laget med kjærlighet av D.

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

Opphavsrett © 2025 – nå · Engineer ApS · Alle rettigheter forbeholdt.

Skrevet ut fra https://engineer.company/nb/glossary/ssrf/