Engineer

SSRF

↰ et angreb, der får en server til at hente angriberens URL

Hvad det er

SSRF, kort for server-side request forgery, er det angreb, hvor nogen giver en server en URL, og serveren henter den på deres vegne, fra det sted i netværket, hvor serveren nu engang står. Common Weakness Enumeration fører det som CWE-918 og beskriver fejlen ud fra det, koden undlader at gøre: den tager imod en URL fra en komponent længere oppe i kæden, henter det, URL'en peger på, og sikrer sig aldrig, at forespørgslen går til den destination, den var tiltænkt. Angriberen ændrer ikke en linje i serverens kode. Angriberen leverer en adresse, og serverens placering i netværket klarer resten.

Den placering er hele pointen. En server i et datacenter eller en cloudkonto kan som regel nå ting, som en besøgende på internettet ikke kan: et adminpanel bundet til en intern adresse, en databaseport, et overvågningsdashboard, en metadatatjeneste. CWE-posten gør det klart, hvad der følger. Serveren bliver proxy for angriberen, den kan sættes til at scanne porte på værter i det interne netværk, og URL'er, der begynder med `file://`, `gopher://` eller `tftp://`, kan læse dokumenter på selve maskinen eller give angriberen mere kontrol over, hvad forespørgslen indeholder. CWE placerer SSRF under en bredere svaghed, CWE-441, den utilsigtede proxy eller »confused deputy«: en komponent med rettigheder, der udfører arbejde for en, der ingen har.

Det, der åbner døren, er som regel en helt almindelig funktion. OWASP's cheat sheet om forebyggelse nævner to typiske tilfælde: et profilbillede, som applikationen henter fra en URL, brugeren har tastet ind, og webhook- eller callback-URL'er, som brugerne registrerer, så applikationen kan kalde dem senere. Forhåndsvisninger af links og knapper til import fra en URL hører til samme familie, ligesom enhver funktion, hvor serveren følger en adresse, som en anden har skrevet. CWE staver selv betegnelsen Server-Side Request Forgery med store forbogstaver, og posten nævner et ældre navn for det samme, XSPA, kort for Cross Site Port Attack.

Brug begrebet i en beslutning

Brug »Hvad er SSRF?« i en konkret beslutning. Vi kan knytte begrebet til det system, I bygger eller køber. Konti, adgangskoder og rettigheder sat op, så folk har præcis den adgang, jobbet kræver, og fratrådte mister den samme dag.

Se vores ydelser Kontakt os →

Undersøg antagelserne bag ordet

Brug forklaringen og det tilknyttede arbejde til at formulere spørgsmål til jeres projekt. Vi kan undersøge antagelserne sammen, før et værktøj vælges. Har skrevet 1.000 RBAC-regler til Linux-servere — 1,000 adgangsregler skrevet for at lukke et sikkerhedshul

Har skrevet 1.000 RBAC-regler til Linux-servere Kontakt os →

Hvorfor det betyder noget

OWASP tog SSRF med på sin Top 10 i 2021 som A10, og ikke på grund af mængden. Dataene viste en relativt lav forekomst, men kategorien var førstevalget i den undersøgelse, OWASP lavede blandt fællesskabet, og dens vurderinger for udnyttelse og konsekvens lå over gennemsnittet. OWASP forklarer det med, at det er blevet en rutinefunktion i webapplikationer at hente en URL, og at cloudtjenester og mere komplekse arkitekturer gør konsekvenserne alvorligere. I 2025-udgaven har SSRF ikke længere sin egen kategori: CWE-918 står blandt de nævneværdige svagheder under A01:2025, Broken Access Control, den kategori, der ligger øverst på listen. MITRE's rangliste peger samme vej, med CWE-918 på CWE Top 25 Most Dangerous Software Weaknesses hvert år fra 2021 til 2025.

Grunden er, at SSRF går lige igennem de forsvar, teams læner sig mest op ad. OWASP understreger, at angrebet virker, selv når applikationen står bag en firewall, en VPN eller en adgangskontrolliste på netværket, fordi forespørgslen går ud indefra. OWASP's eksempelscenarier er en rundtur i det, der gemmer sig bag den perimeter: at kortlægge interne værter og åbne porte ud fra, om forbindelser lykkes, eller hvor lang tid de tager, at læse lokale filer som `file:///etc/passwd`, at tale med interne tjenester som `http://localhost:28017/` og at misbruge de tjenester til at komme videre, helt til fjernkørsel af kode eller lammelsesangreb.

Cloud-metadata er det scenarie, der gør en irritation til et brud. OWASP bemærker, at de fleste cloududbydere har instansens metadata liggende på `http://169.254.169.254/`, en adresse i blokken 169.254.0.0/16, som IANA's register over adresser til særlige formål opfører som link-local. På Amazon EC2 er instansens metadatatjeneste på den adresse også stedet, hvor software på instansen henter AWS-legitimationsoplysninger for den rolle, der er knyttet til den. En server, der henter enhver URL for hvem som helst, kan blive bedt om at hente dem.

Mønstret bliver ved med at dukke op nye steder. Blandt de nyeste eksempler i CWE-posten er et LLM-værktøjssæt, der hentede vilkårlige billed-URL'er og blev aktivt udnyttet i april 2026 til portscanning (CVE-2026-33626), og et framework til LLM-applikationer, hvis URL-henter kunne styres mod lokale adresser med en udformet `Location` -header (CVE-2024-3095). En agent, der søger og læser på nettet for en bruger, er set fra netværkets side blot en server, der henter URL'er, som en anden har valgt.

Sådan ser det ud i praksis

Tag en chatapplikation på `app.example.com` , der viser et forhåndsvisningskort, når nogen indsætter et link. Browseren sender URL'en til serveren, og serveren henter siden for at læse dens titel og billede:

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

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

Sådan virker funktionen efter hensigten. Se så den samme forespørgsel med en anden URL, den, som AWS' dokumentation bruger til at 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 kører på EC2, og den ældre IMDSv1 stadig er tilladt, sender den forespørgslen indefra instansen, metadatatjenesten svarer, og det, forhåndsvisningskoden sender tilbage, ender hos angriberen. Intet blev sprøjtet ind, og intet gik ned. Serveren gjorde præcis det, den var bygget til, bare for den forkerte adresse.

En kontrol af destinationen begynder med, hvor forespørgslen faktisk ville ende, efter at navnet er slået op, og ikke med URL'ens tekst. I Python bruger standardbibliotekets `ipaddress` -modul IANA's registre over adresser til særlige

formål til at klassificere, om en adresse er global. Det udelukker adresser, der direkte tilhører loopback-, private, link-local-, delte eller dokumentationsområder. Hjælpefunktionen nedenfor illustrerer denne klassifikation, ikke et fuldstændigt værn mod SSRF. En udrulning skal også begrænse forespørgsler til tilladte unicastdestinationer og tage højde for adresseoversættelse som NAT64: på et netværk med en oversættende gateway kan en adresse klassificeret som global stadig nå en blokeret, indlejret IPv4-destination. Hjælpefunktionens klassifikationstest 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 slår op til, skal bestå, ikke kun den første. Selve hentningen forbinder derefter til en af de godkendte adresser, men sender stadig det oprindelige værtsnavn i `Host`-headeren og bruger det til at kontrollere TLS-certifikatet, og omdirigeringer er slået fra. Lader man HTTP-klienten slå navnet op en gang til, er tjekket spildt. Det er den form, OWASP's cheat sheet beskriver for det svære tilfælde, webhooks, hvor brugerne selv vælger destinationerne, og ingen liste over tilladte adresser kan nævne dem på forhånd. Hvor destinationerne er kendte, snævrer en liste over præcise værtsnavne feltet yderligere ind, men cheat sheetet slår fast, at de opslåede adresser stadig skal igennem samme test.

Applikationens tjek er ét lag. Vores beskrivelse af e-mail med failover og leveringslog nævner, at selve afsendelsen går gennem en SSRF-beskyttet klient med en timeout på 30 sekunder. Bag applikationen gælder OWASP's råd for

netværkslaget stadig: læg den kode, der henter eksterne ressourcer, i sit eget netværkssegment, og giv den firewallregler, der afviser alt, som ikke udtrykkeligt er tilladt, så et overset tjek møder en lukket dør.

Hvor det går galt

Blokeringslister og mønstergenkendelse. Et filter, der afviser URL'er med `127.0.0.1`, `localhost` eller `169.254.169.254`, har allerede tabt. POSIX-funktionen `inet_addr()` tager imod en IPv4-adresse i en til fire dele, hver skrevet decimalt, oktalt eller hexadecimalt, så for enhver parser, der følger de regler, er både `2852039166` og `0xa9.0xfe.0xa9.0xfe` metadataadressen. CWE-posten nævner en CVE, hvor oktale og hexadecimale former slap forbi et tjek skrevet til decimaltal og satte en SSRF-beskyttelse ud af spil (CVE-2016-4029). OWASP's Top 10 siger det ligeud: SSRF skal ikke afhjælpes med en blokeringsliste eller et regulært udtryk. Afgør det på den opslåede adresse, som ovenfor, og tillad frem for at forbyde.

Tjek nu, forbind senere. At validere værtsnavnet og så lade HTTP-klienten slå det op igen efterlader et hul. Med DNS rebinding svarer angriberens navn med en offentlig adresse ved tjekket og en intern ved forbindelsen. Forbind til præcis den adresse, du validerede, og brug samme politik ved genforsøg og reserveforbindelser, som OWASP's cheat sheet nævner ved navn.

Omdirigeringer, der bliver fulgt. Den godkendte URL svarer med en omdirigering til `http://169.254.169.254/` eller til en `file://`-URL, og klienten følger den uden at spørge. Blandt eksemplerne i CWE-posten er et bibliotek til at hente URL'er, der fulgte omdirigeringer til `file://`- og `scp://`-URL'er (CVE-2009-0037). Slå omdirigeringer fra, eller behandl hvert hop som en ny URL, der skal igennem hele tjekket igen.

Klienten taler mere end HTTP. Forstår den også `file`, `gopher` eller `dict`, kan en angriber læse lokale filer eller få langt mere kontrol over, hvad der når frem til en intern tjeneste, end en HTTP-forespørgsel giver. OWASP's cheat sheet nævner netop dem blandt de URL-typer, en anden forespørgsel kan bruge. Tillad kun `http` og `https` og kun de porte, funktionen har brug for.

To parsere, to værter. OWASP's cheat sheet viser, hvordan én streng kan pege på to forskellige værter. I `http://good.example\@bad.example` læser en parser, der følger WHATWG's URL-standard, backslashen som stiseparator og værten som `good.example`, mens Pythons `urlsplit` tager alt før det sidste `@` som

brugeroplysninger og returnerer `bad.example`. Når én komponent validerer, og en anden henter, kan de være uenige om, hvor forespørgslen går hen. Parse URL'en én gang, og afvis enhver URL, hvis vært ikke læses ens af alle de parsere, den kommer igennem.

Det rå svar sendes tilbage. Et endpoint til forhåndsvisninger, der sender det hentede indhold retur til kalderen, gør et angreb, der kun kunne gætte sig til, hvad der står indenfor, ud fra fejl og svartider, til et, der kan læse det. Returner kun det, funktionen har brug for, fx en titel og en henvisning til et billede, og aldrig det rå svar, sådan som OWASP's liste råder til.

IMDSv1 står stadig til. AWS' dokumentation siger, at begge versioner af instansens metadatatjeneste som standard kan bruges. IMDSv2 kræver først et sessionstoken, som hentes med en `PUT` til `/latest/api/token` og derefter sendes i headeren `X-aws-ec2-metadata-token`, og når IMDSv2 er påkrævet (`httpTokens=required`), holder IMDSv1 op med at virke. OWASP's cheat sheet kalder IMDSv2 et ekstra forsvarslag, der afbøder nogle tilfælde af SSRF, ikke dem alle. Kræv IMDSv2 på alle instanser, og ret applikationen alligevel.

Et fladt netværk. Når den tjeneste, der henter URL'er, kan nå databasen, adminpanelet og metadatatjenesten, er hvert overset tjek fatalt. Brug OWASP's kontroller på netværkslaget: segmentering, firewallregler, der som udgangspunkt afviser intern trafik, og logning af tilladte og blokerede forbindelser, så en sondering bliver synlig.

Læs mere

- CWE-918: Server-Side Request Forgery (SSRF) — svaghedsposten: konsekvenserne, »confused deputy«-forælderen og de observerede CVE'er.
- OWASP Top 10:2021: A10 Server Side Request Forgery (SSRF) — hvorfor kategorien kom med, kontroller på netværks- og applikationslaget og fire angrebsscenarier.
- OWASP Cheat Sheet Series: Server Side Request Forgery Prevention — lister over tilladte adresser, forbindelse kun til validerede adresser, uenige parsere og webhook-tilfældet.
- OWASP Top 10:2025: A01 Broken Access Control — hvor CWE-918 står i 2025-udgaven.
- Amazon EC2: Access instance metadata for an EC2 instance — IMDSv1 og IMDSv2, tokenforespørgslen og hvordan man ser, om IMDSv2 er påkrævet.

- Python documentation: ipaddress — `is_global` og de IANA-registre, den følger.

Ordliste

Vælg ud fra jeres faktiske vilkår

Medbring udrulningsforløbet, driftsrisiciene og dem, der har vagten. Vi kan afgrænse en ændring med kontroller, en vej tilbage og tydeligt ansvar.

Se, hvad det koster Kontakt os →

Gør begreberne praktiske

Spørg, hvordan »Hvad er SSRF?« gælder i jeres organisation. Et konkret eksempel fra jeres nuværende system er nok til at begynde samtalen.

Se vores ydelser

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

Tag denne side med dig

Visitkort

Lavet med kærlighed af D.

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

Copyright © 2025 – nu · Engineer ApS · Alle rettigheder forbeholdes.

Udskrevet fra https://engineer.company/da/glossary/ssrf/