Engineer

Onion-tjeneste

en adresse som bare virker gjennom Tor

Hva det er

En onion-tjeneste er en server leserne når inne i Tor-nettverket via et navn bygd av serverens egen offentlige nøkkel, i stedet for et DNS-navn som peker på en IP-adresse. Mange forkorter det til en onion, én tor-daemon kan kjøre flere onion-tjenester side om side, og spesifikasjonen bruker fortsatt det eldre navnet hidden service, skjult tjeneste.

Navnet er 56 tegn i base32 fulgt av `.onion`. Dekodet blir det 35 byte: tjenestens 32 byte store ed25519-identitetsnøkkel, en sjekksum på to byte hentet fra en SHA3-256-hash av den nøkkelen, og en versjonsbyte med verdien 3. Fordi den byten kommer sist, slutter alle gjeldende adresser på bokstaven d. Ingen velger navnet; det er nøkkelen skrevet ut. Navnene i versjon 2, som kom før, var 16 tegn kuttet fra en SHA-1-hash av en RSA-1024-nøkkel, og tor fjernet dem i utgave 0.4.6.1-alpha, i 2021.

Å nå tjenesten krever to møtesteder. Tjenesten velger noen reléer som introduksjonspunkter, holder kretser åpne til dem og laster opp en signert beskrivelse (descriptor) som lister dem, til katalogreléer. Beskrivelsen arkiveres under en blindet nøkkel som skifter for hver periode, og innholdet er kryptert, så et relé som lagrer den uten å kjenne adressen, kan ikke lese listen. En klient som har adressen, henter beskrivelsen, sjekker signaturen mot nøkkelen i navnet, velger et relé som møtepunkt (rendezvous) og ber tjenesten, via et introduksjonspunkt, om å møtes der. Tjenesten bygger sin egen krets til det reléet og fullfører håndtrykket, og reléet kobler de to kretsene sammen. Veien går gjennom seks reléer, tre valgt av hver side, og ingen av endene får vite den andres IP-adresse. En valgfri modus, som nå heter restricted discovery og tidligere het client authorization, krypterer i tillegg beskrivelsen for nøkkelen til hver autoriserte klient, så bare de klientene får vite hvor introduksjonspunktene er.

Bruk begrepet i en beslutning

Bruk «Hva er en onion-tjeneste?» i en konkret beslutning. Vi kan knytte begrepet til systemet dere bygger eller kjøper. Wi-Fi, brannmurer og fjerntilgang satt opp slik at de ansatte kan jobbe hvor som helst, og bare de rette personene slipper inn.

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. Administrerte 30 Ubuntu Linux-VPS-instanser — 30 Linux-servere administrert, med gjenoppretting planlagt på forhånd

Administrerte 30 Ubuntu Linux-VPS-instanser Ta kontakt →

Hvorfor det er viktig

Navnet gjør jobben som DNS og en sertifikatutsteder deler på for et vanlig nettsted, og trenger ingen av dem. Når klienten har verifisert beskrivelsen og fullført håndtrykket, vet den at den snakker med den som har den private nøkkelen bak adressen. RFC 7686, som reserverer `.onion` som et domenenavn til spesiell bruk, beskriver slike navn som en kombinasjon av sted og autentisering, og trekker konsekvensen: et onion-navn kan ikke registreres, tildeles, overføres eller trekkes tilbake. Å eie et slikt navn betyr å kontrollere nøkkelparet, verken mer eller mindre.

Krypteringen følger av den samme konstruksjonen. Trafikk til en onion-tjeneste forlater aldri Tor-nettverket, så det finnes ikke noe utgangsrelé i den andre enden og ingen siste strekning i klartekst. Vanlig `http://` mot en onion gir derfor en side som er kryptert hele veien, fra en vert som klienten har sjekket identiteten til mot navnet som ble tastet inn.

Tjenesten ringer bare ut, til introduksjonspunktene og til hvert møtepunkt, så den trenger ingen åpen innkommende nettverksport og virker bak NAT. Å legge til en onion endrer ikke listen over åpne nettverksporter på verten.

Leserne må likevel få vite adressen. Et nettsted kan kunngjøre den med svarheaderen `Onion-Location`, og Tor Browser viser da en lilla pille i adressefeltet som fører til onion-kopien av siden. Tor-prosjektet stiller tre vilkår: verdien er en `http:` - eller `https:` -URL med en onion-vert, siden som sender den, kom over HTTPS, og den siden ligger ikke selv på en onion. En meta-tagg kan bære den samme kunngjøringen der headere ikke kan settes, men uten stien, så hver side peker på én fast URL.

Slik ser det ut i praksis

Hvordan selskapets onion-speil ble publisert på en lesbar adresse er beskrevet steg for steg. Daemonen kommer fra distribusjonens egen pakke, med nettverksporten for utgående proxy slått av. Den private nøkkelen bak adressen ble generert utenfor verten og har aldri havnet i repoet eller på en bærbar PC; rollen nekter å overskrive en identitet den ikke selv genererte, så en nøkkel som er plassert for hånd, brukes som den er. Adressen er en deklarert verdi i inventaret, og hver converge-kjøring kontrollerer at identitetsfilen på verten stemmer med den, så en byttet nøkkel med en utdatert deklarasjon stopper kjøringen. Begge hemmelighetsvaktene fanger nøkkelfilen, etter at en måling viste at det opplagte mønsteret for nøkkelfiler traff en deploy-nøkkel og bommet på denne.

Et lesbart prefiks kan finnes, men aldri velges. Hvert ekstra tegn ganger det forventede søket med 32, så de åtte bokstavene i `engineer` krever i snitt ca. 1,1 billioner nøkkelpar, altså rundt 1 100 milliarder. Å pensjonere den tilfeldige adressen speilet først ble levert med, var en trinnvis overgang, så den publiserte adressen svarte hele tiden. Ett steg så endelig ut uten å være det: nøkkelen kom på plass, converge-kjøringen meldte ingen endring, og den nye adressen ble aldri publisert, fordi ingenting navnga katalogen dens. Rollen feiler nå på enhver identitet på disk som ingen tjeneste navngir.

Kolofonen skriver ut hele adressen, `engineer2ezpeji52ipphljcflzyhetvy5r6tnqubtq45nzlxxv6rsqd.onion`, ved siden av adressene for Gemini og Gopher. Onion-tjenesten leser de samme filene som de tre andre inngangsdørene, så onion-adressen holder takt med dem etter hver produksjonssetting. Det vanlige nettstedet sender `Onion-Location`, så Tor Browser tilbyr onion-adressen uten at leseren må be om den. Siden forklarer også hva leserens eget nettverk ser gjennom hver dør: over Gopher hvilken side som leses, over Gemini hvilken adresse det ble spurt etter, over onion ingen av delene. Den kaller prefikset en bekvemmelighet og ikke et bevis, og ber leserne sammenligne hele adressen.

Adressens oppbygning kan sjekkes uten nett. Disse linjene dekoder den og regner ut sjekksummen på nytt, og begge sammenligningene skriver ut `True`:

```
import base64
import hashlib

label = "engineer2ezpeji52ipphljcflzyhetvy5r6tnqubtq45nzlxxv6rsqd"
```

```
raw = base64.b32decode(label.upper())
key, checksum, version = raw[:32], raw[32:34], raw[34:]
digest = hashlib.sha3_256(b".onion checksum" + key + version).digest()
print(version == b"\x03", digest[:2] == checksum)
```

En gyldig sjekksum viser at adressen er riktig formet, ikke hvem den tilhører. Det får du vite ved å lese den på selskapets egne HTTPS-sider.

Hvor det går galt

Nøkkelen er det eneste skjøtet. Uten en registrar bak navnet betyr en tapt privat nøkkel en tapt adresse, og en kopiert nøkkel lar noen andre svare på den. Tor-prosjektets oppsettsveiledning sier at nøkkelfilene skal holdes private, og at nøkkelen bør sikkerhetskopieres hvis tjenesten skal vare. Behandle den som en signeringsnøkkel, ikke som konfigurasjon.

Bare de første bokstavene blir lest. RFC 7686 advarer om at en angriper kan finne nøkler med navn som ligner det tiltenkte, og at risikoen er større fordi navnene ikke betyr noe for et menneske. Den regner lesbare navn, kjøpt med betydelig regnekraft, som ett mottiltak, men et prefiks på åtte tegn lar de andre 48 være like uleselige som før. Sammenlign hele adressen, eller kom dit via et bokmerke eller en lenke på nettstedets egne HTTPS-sider.

Serveren røper seg selv. Tor skjuler hvor tjenesten kjører, men programvaren bak den kan fortsatt fortelle det. RFC 7686 regner en server som lekker identiteten sin i en melding på applikasjonsnivå blant måtene vernet svikter på, og en feilside som skriver ut maskinens virkelige vertsnavn, er et eksempel. Oppsettsveiledningen anbefaler også å sende trafikken til webserveren over en Unix-socket i stedet for TCP, så tjenesten ikke lekker ut på det lokale nettverket:

```
HiddenServiceDir /var/lib/tor/my-website/
HiddenServicePort 80 unix:/var/run/tor/my-website.sock
```

Det betyr mest der selve plasseringen er det som skal holdes skjult.

Navnet havner hos DNS. Programvare som ikke snakker Tor, spør den vanlige resolveren om et onion-navn, og forteller dermed det lokale nettverket og DNS-leverandøren hvilken onion noen ville besøke. RFC 7686 sier at slik programvare

skal feile uten å gjøre oppslaget, og at DNS-servere skal svare at navnet ikke finnes. Gi navnet til Tor-klienten i stedet.

En `Onion-Location` -pille som aldri dukker opp. En header som bare settes på omdirigeringen over vanlig HTTP, bryter HTTPS-vilkåret, og en med feilskrevet adresse sender leserne ingen steder. Sjekk headeren på det endelige HTTPS-svaret.

Les mer

- Tor Rendezvous Specification, version 3 — hele protokollen: identitetsnøkler og blindede nøkler, beskrivelser, introduksjon og møtepunkt.
- Encoding onion addresses — formelen som gjør en offentlig nøkkel om til et navn på 56 tegn, med eksempeladresser.
- How do Onion Services work? — en gjennomgang steg for steg av introduksjonspunkter, beskrivelser og møtet mellom klient og tjeneste.
- Set up Your Onion Service — linjene i torrc, filene tor lager, og råd om nøkler og om lekkasjer av plassering.
- Onion-Location — oppsett med header og meta-tagg for Apache, Nginx og Caddy, og vilkårene headeren må oppfylle.
- RFC 7686 — hvorfor `.onion` er reservert, og hva resolvere, DNS-operatører og registrarer må gjøre med det.

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 en onion-tjeneste?» 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/onion/