Engineer

Onion-tjeneste

en adresse, der kun virker gennem Tor

Hvad det er

Hvor et almindeligt website findes via et DNS-navn, der peger på en IP-adresse, nås en onion-tjeneste inde i Tor-netværket under et navn, der er afledt af tjenestens egen offentlige nøgle. I daglig tale siger man ofte bare en onion, én tor-dæmon kan køre flere onion-tjenester side om side, og specifikationen bruger stadig det ældre navn hidden service.

Selve navnet består af 56 tegn i base32 efterfulgt af `.onion`. Afkodet giver det 35 bytes: tjenestens ed25519-identitetsnøgle på 32 bytes, en kontrolsum på to bytes taget fra en SHA3-256-hash af nøglen og en versionsbyte med værdien 3. Den byte står sidst, og derfor ender alle nuværende adresser på bogstavet d. Ingen vælger etiketten; den er nøglen skrevet ud. Version 2-navnene, der gik forud, var 16 tegn skåret ud af en SHA-1-hash af en RSA-1024-nøgle, og tor droppede dem i udgave 0.4.6.1-alpha i 2021.

For at to parter kan mødes, uden at nogen af dem afslører, hvor de er, bruger protokollen to slags mødesteder. Tjenesten udpeger nogle relæer som introduktionspunkter, holder kredsløb åbne til dem og lægger en underskrevet beskrivelse (descriptor), der lister dem, op på relæer, der fungerer som katalog. Beskrivelsen arkiveres under en blindet nøgle, der skifter for hver periode, og indholdet er krypteret, så et relæ, der gemmer den uden at kende adressen, ikke kan læse listen. En klient, der kender adressen, henter beskrivelsen, kontrollerer underskriften mod nøglen i navnet, vælger selv et relæ som rendezvous-punkt og beder gennem et af introduktionspunkterne tjenesten om at møde op dér. Tjenesten bygger sit eget kredsløb til rendezvous-punktet og gør håndtrykket færdigt, og relæet kobler de to kredsløb sammen. Hele vejen går over seks relæer, tre valgt af hver side, og ingen af enderne lærer den andens IP-adresse. Med en valgfri tilstand, der nu hedder restricted discovery og tidligere hed client authorization, krypteres beskrivelsen desuden til hver godkendt klients nøgle, så kun de klienter får at vide, hvor introduktionspunkterne er.

Brug begrebet i en beslutning

Brug »Hvad er en onion-tjeneste?« i en konkret beslutning. Vi kan knytte begrebet til det system, I bygger eller køber. Wi-Fi, firewalls og fjernadgang sat op, så medarbejdere kan arbejde hvor som helst, og kun de rette kommer ind.

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 administreret 30 Ubuntu Linux VPS-instanser — 30 Linux-servere driftet, med en plan for gendannelse

Har administreret 30 Ubuntu Linux VPS-instanser Kontakt os →

Hvorfor det betyder noget

For et almindeligt website deler DNS og en certifikatudsteder opgaven med at fortælle læseren, hvem hun eller han taler med. En onion-tjeneste klarer sig uden dem begge, for navnet gør arbejdet selv: når klienten har kontrolleret beskrivelsens underskrift og gennemført håndtrykket, ved den, at modparten er den, der har den private nøgle bag adressen. RFC 7686, som reserverer `.onion` som et domænenavn til særlig brug, beskriver sådanne navne som en sammensmeltning af placering og autentificering og drager konsekvensen: et onion-navn kan hverken registreres, tildeles, overdrages eller tilbagekaldes. At eje et betyder at råde over nøgleparret, hverken mere eller mindre.

Krypteringen følger af samme konstruktion. Trafik til en onion-tjeneste forlader aldrig Tor-netværket, så der sidder intet exit-relæ for enden og ingen sidste strækning i klartekst. Almindelig `http://` på en onion bærer derfor en side, der er krypteret hele vejen, til en vært, hvis identitet klienten har tjekket mod det navn, den fik.

Set fra driftens side er retningen på forbindelserne lige så vigtig. Tjenesten ringer kun ud, til sine introduktionspunkter og til hvert rendezvous-punkt, så den behøver ingen åben indgående port og virker bag NAT. Værtens liste over åbne porte er den samme bagefter som før.

Læserne skal stadig lære adressen at kende. Et website kan annoncere den med en `Onion-Location` -header i svaret, og Tor Browser viser så en lilla knap i adresselinjen, der fører til onion-udgaven af siden. Tor-projektet stiller tre betingelser: værdien er en `http:` - eller `https:` -URL med en onion-vært, siden, der sender den, er hentet over HTTPS, og den side ligger ikke selv på en onion. Et meta-tag kan bære samme besked, hvor man ikke kan sætte headers, men uden stien, så hver side peger på én fast URL.

Sådan ser det ud i praksis

Virksomhedens onion-spejl på en læsbar adresse er beskrevet trin for trin. Dæmonen kommer fra distributionens egen pakke, med den udgående proxyport slået fra. Den private nøgle bag adressen blev genereret uden for hosten og har aldrig været i repositoriet eller på en laptop; rollen nægter at overskrive en identitet, den ikke selv har lavet, så en nøgle lagt på plads i hånden bliver brugt, som den er. Adressen står som en erklæret værdi i inventaret, og hver konvergens kontrollerer, at identitetsfilen på hosten matcher den, så en udskiftet nøgle med en forældet erklæring stopper kørslen. Begge hemmelighedsvagter fanger nøglefilen, efter at en måling viste, at det oplagte mønster for nøglefiler ramte en deploy-nøgle og missede denne.

Et læsbart præfiks kan man lede efter, men aldrig vælge. Hvert ekstra tegn koster toogtredive gange det forrige, så de otte bogstaver i `engineer` kræver i gennemsnit omkring 1.100 milliarder nøglepar. Den tilfældige adresse, spejlet først kørte med, blev pensioneret ved en trinvis omlægning, så den udgivne adresse virkede hele vejen igennem. Et trin lignede det sidste uden at være det: nøglen landede, konvergensen meldte ingen ændring, og den nye adresse blev aldrig udgivet, fordi intet navngav dens mappe. Nu fejler rollen på enhver identitet på disken, som ingen tjeneste navngiver.

Kolofonen gengiver hele adressen, `engineer2ezpeji52ipphljcflzyhetvy5r6tnqubtq45nzlxxv6rsqd.onion`, ved siden af Gemini- og Gopher-adresserne. Onion-tjenesten henter sine filer fra den mappe, som web, Gemini og Gopher også læser, så den ikke kan komme ud af trit med dem: hver udrulning når alle fire på én gang. Webudgaven sender `Onion-Location`, så Tor Browser selv tilbyder onion-adressen. Siden gør også rede for, hvad læserens eget netværk ser gennem hver dør: over Gopher hvilken side der læses, over Gemini hvilken adresse der blev bedt om, over onion-tjenesten ingen af delene. Præfikset kalder den en bekvemmelighed, ikke et bevis, og den beder læseren sammenligne hele adressen.

Adressens opbygning kan efterprøves uden netforbindelse. Disse linjer afkoder den og regner kontrolsummen efter, og begge sammenligninger udskriver `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 kontrolsum viser, at adressen er velformet, men ikke hvem den tilhører. Det får man at vide ved at læse den på virksomhedens egne HTTPS-sider.

Hvor det går galt

Nøglen er det eneste skøde. Der står ingen registrator bag navnet, så en mistet privat nøgle er en mistet adresse, og en kopieret nøgle lader en anden svare på den. Tor-projektets opsætningsvejledning siger, at nøglefilerne skal holdes private, og at man bør tage backup af nøglen, hvis tjenesten skal leve længe. Behandl den som en signeringsnøgle og ikke som konfiguration.

Kun de første bogstaver bliver læst. RFC 7686 advarer om, at en angriber kan finde nøgler, hvis navne ligner det tilsigtede, og at risikoen vokser, fordi navnene intet betyder for et menneske. RFC'en regner læsbare navne, købt med betydelig regnekraft, som én modforanstaltning, men et præfiks på otte tegn efterlader de øvrige 48 lige så uigennemskuelige som før. Sammenlign hele adressen, eller kom ind via et bogmærke eller et link på sitets egne HTTPS-sider.

Serveren røber sig selv. Tor skjuler, hvor tjenesten kører, men softwaren bag den kan stadig fortælle det. RFC 7686 nævner en server, der lækker sin identitet i en besked på applikationsniveau, blandt de måder beskyttelsen brydes på, og en fejlside med maskinens rigtige værtsnavn er netop sådan en. Opsætningsvejledningen anbefaler desuden at give trafikken videre til webserveren over en Unix-socket i stedet for TCP, så tjenesten ikke siver ud på det lokale netværk:

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

Det vejer tungest dér, hvor selve placeringen er hemmeligheden.

Navnet havner i DNS. Software, der ikke taler Tor, spørger den almindelige resolver om et onion-navn, og så ved det lokale netværk og DNS-udbyderen, hvilken onion der var tale om. RFC 7686 siger, at sådan software skal melde fejl uden at slå navnet op, og at DNS-servere skal svare, at navnet ikke eksisterer. Giv i stedet navnet videre til Tor-klienten.

En `Onion-Location` -knap, der aldrig dukker op. En header, der kun sættes på omdirigeringen fra almindelig HTTP, bryder kravet om HTTPS, og en med en tastefejl i adressen sender læserne ud i ingenting. Kontrollér headeren på det endelige HTTPS-svar.

Læs mere

- Tor Rendezvous Specification, version 3 — hele protokollen: identitetsnøgler og blindede nøgler, beskrivelser, introduktion og rendezvous.
- Encoding onion addresses — formlen, der gør en offentlig nøgle til et navn på 56 tegn, med eksempeladresser.
- How do Onion Services work? — introduktionspunkter, beskrivelser og rendezvous gennemgået trin for trin.
- Set up Your Onion Service — torrc-linjerne, de filer tor opretter, og råd om nøgler og om at røbe placeringen.
- Onion-Location — opsætning af header og meta-tag til Apache, Nginx og Caddy og de betingelser, headeren skal opfylde.
- RFC 7686 — hvorfor `.onion` er reserveret, og hvad resolvere, DNS-operatører og registratorer skal gøre med det.

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 en onion-tjeneste?« 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/onion/