rychlý, moderní a bezpečný VPN tunel

wireguard.com

WireGuard ® je extrémně jednoduchá, ale rychlá a moderní VPN, která využívá **nejmodernější kryptografii** . Jeho cílem je být rychlejší , jednodušší , štíhlejší a užitečnější než IPsec a zároveň se vyhnout velkým bolestem hlavy. Má v úmyslu být podstatně výkonnější než OpenVPN. WireGuard je navržen jako univerzální VPN pro běh na vestavěných rozhraních a superpočítačích, která je vhodná pro mnoho různých okolností. Původně byl vydán pro linuxové jádro, nyní je multiplatformní (Windows, macOS, BSD, iOS, Android) a široce implementovatelný. V současné době je intenzivně vyvíjen, ale již může být považován za nejbezpečnější, nejsnadněji použitelné a nejjednodušší řešení VPN v oboru.

Jednoduché a snadno použitelné

WireGuard si klade za cíl být stejně snadné na konfiguraci a nasazení jako SSH. Připojení VPN se provádí jednoduše výměnou velmi jednoduchých veřejných klíčů – přesně jako výměna klíčů SSH – a vše ostatní transparentně řeší WireGuard. Je dokonce schopen roamingu mezi IP adresami, stejně jako Mosh . Není třeba spravovat připojení, starat se o stav, spravovat démony nebo se starat o to, co je pod kapotou. WireGuard představuje extrémně základní, ale výkonné rozhraní.

Kryptograficky zvuk

WireGuard využívá nejmodernější kryptografii, jako je rámec protokolu Noise , Curve25519 , ChaCha20 , Poly1305 , BLAKE2 , SipHash24 , HKDF a bezpečné důvěryhodné konstrukce. Dělá konzervativní a rozumná rozhodnutí a byl zkontrolován kryptografy.

Minimální útočná plocha

WireGuard byl navržen s ohledem na snadnou implementaci a jednoduchost. Má být snadno implementován ve velmi malém počtu řádků kódu a snadno auditovatelný pro zranitelnosti zabezpečení. Ve srovnání s monstry jako *Swan/IPsec nebo OpenVPN/OpenSSL, ve kterých je auditování gigantických kódových základen zdrcujícím úkolem i pro velké týmy bezpečnostních expertů, má být WireGuard komplexně kontrolovatelný jednotlivými jednotlivci.

Vysoký výkon

Kombinace extrémně vysokorychlostních kryptografických primitiv a skutečnosti, že WireGuard žije uvnitř linuxového jádra, znamená, že zabezpečené sítě mohou být velmi rychlé. Je vhodný jak pro malá vestavěná zařízení, jako jsou chytré telefony, tak pro plně zatížené páteřní routery.

Dobře definované a důkladně zvážené

WireGuard je výsledkem zdlouhavého a důkladně promyšleného akademického procesu, jehož výsledkem je technický whitepaper , akademický výzkumný dokument, který jasně definuje protokol a intenzivní úvahy, které byly součástí každého rozhodnutí.

Koncepční přehled

Pokud byste chtěli obecný koncepční přehled o tom, o čem WireGuard je, čtěte dále zde. Poté můžete přejít k instalaci a přečíst si pokyny pro rychlý start , jak ji používat.

Pokud vás zajímá vnitřní vnitřní fungování, mohlo by vás zajímat stručné shrnutí protokolu nebo jít více do hloubky přečtením technického dokumentu , který se podrobněji zabývá protokolem, kryptografií a základy. Pokud máte v úmyslu implementovat WireGuard pro novou platformu, přečtěte si prosím poznámky pro různé platformy .

WireGuard bezpečně zapouzdřuje IP pakety přes UDP. Přidáte rozhraní WireGuard, nakonfigurujete jej pomocí svého soukromého klíče a veřejných klíčů vašich kolegů a poté přes něj posíláte pakety. Všechny problémy distribuce klíčů a push konfigurací jsou *mimo rozsah* WireGuard; to jsou problémy, které je lepší nechat pro jiné vrstvy, abychom neskončili s nadýmáním IKE nebo OpenVPN. Oproti tomu více napodobuje model SSH a Mosh; obě strany mají navzájem své veřejné klíče a pak si mohou jednoduše začít vyměňovat pakety přes rozhraní.

Jednoduché síťové rozhraní

WireGuard funguje přidáním síťového rozhraní (nebo více), jako `eth0` nebo `wlan0` , volaného `wg0` (nebo `wg1` , `wg2` , `wg3` atd.). Toto síťové rozhraní lze poté konfigurovat normálně pomocí `ifconfig(8)` nebo `ip-address(8)` , přičemž trasy pro něj lze přidávat a odebírat pomocí `route(8)` nebo `ip-route(8)` , a tak dále se všemi běžnými síťovými nástroji. Konkrétní aspekty rozhraní WireGuard se konfigurují pomocí tohoto `wg(8)` nástroje. Toto rozhraní funguje jako tunelové rozhraní.

WireGuard spojuje IP adresy tunelu s veřejnými klíči a vzdálenými koncovými body. Když rozhraní odešle paket peerovi, provede následující:

1. Tento paket je určen pro 192.168.30.8. Který je to vrstevník? Nech mě se podívat... Dobře, je to pro vrstevníky `ABCDEFGH` . (Nebo pokud to není pro žádného nakonfigurovaného peer, paket zahoďte.)

2. Šifrujte celý IP paket pomocí `ABCDEFGH` veřejného klíče partnera.
3. Co je vzdálený koncový bod peer `ABCDEFGH`? Podívej se... Dobře, koncovým bodem je port UDP 53133 na hostiteli 216.58.211.110.
4. Odešlete šifrované bajty z kroku 2 přes internet na 216.58.211.110:53133 pomocí UDP.

Když rozhraní přijme paket, stane se toto:

1. Právě jsem dostal paket z portu UDP 7361 na hostiteli 98.139.183.24. Pojďme to dešifrovat!
2. Pro peer se správně dešifroval a autentizoval `LMNOPQRS`. Dobře, připomeňme si, že `LMNOPQRS` nejnovější internetový koncový bod peer je 98.139.183.24:7361 používající UDP.
3. Po dešifrování je paket prostého textu z 192.168.43.89. Může nám peer `LMNOPQRS` posílat pakety jako 192.168.43.89?
4. Pokud ano, přijměte paket na rozhraní. Pokud ne, zahoďte to.

V zákulisí se toho děje mnoho, aby bylo zajištěno náležité soukromí, autenticita a dokonalé dopředné utajení pomocí nejmodernější kryptografie.

Směrování kryptoklíčů

Základem WireGuard je koncept nazvaný *Cryptokey Routing* , který funguje tak, že spojuje veřejné klíče se seznamem IP adres tunelu, které jsou povoleny v tunelu. Každé síťové rozhraní má soukromý klíč a seznam partnerů. Každý peer má veřejný klíč. Veřejné klíče jsou krátké a jednoduché a používají je kolegové ke vzájemné autentizaci. Lze je předat pro použití v konfiguračních souborech jakoukoli mimopásmovou metodou, podobně jako je možné poslat svůj veřejný klíč SSH příteli pro přístup k serveru shellu.

Serverový počítač může mít například tuto konfiguraci:

```
[Interface]
PrivateKey = yAnz5TF+lXXJte14tji3zlMNq+hd2rYUIgJBgB3fBmk=
ListenPort = 51820

[Peer]
PublicKey = xTIBA5rboUvnH4htodjb6e697QjLERt1NAB4mZqp8Dg=
AllowedIPs = 10.192.122.3/32, 10.192.124.1/24

[Peer]
PublicKey = TrMvSoP4jYQlY6RIzBgbssQqY3vxI2Pi+y71lOWWXX0=
AllowedIPs = 10.192.122.4/32, 192.168.0.0/16

[Peer]
PublicKey = gN65BkIKy1eCE9pP1wdc8ROUtkHLF2PfAqYdyYBz6EA=
AllowedIPs = 10.10.10.230/32
```

A klientský počítač může mít tuto jednodušší konfiguraci:

```
[Interface]
PrivateKey = gI6EdUSYvn8ugXOt8QQD6Yc+JyiZxIhp3GInSWRfWGE=
ListenPort = 21841

[Peer]
PublicKey = HIgo9xNzJMWLKASShiTqIybxZ0U3wGLiUeJ1PKf8ykw=
Endpoint = 192.95.5.69:51820
AllowedIPs = 0.0.0.0/0
```

V konfiguraci serveru bude každý peer (klient) schopen posílat pakety do síťového rozhraní se zdrojovou IP odpovídající jeho odpovídajícímu seznamu povolených IP adres. Například, když server přijme paket od peer `gN65BkIK...`, poté, co byl dešifrován a ověřen, je-li jeho zdrojová IP adresa 10.10.10.230, pak je povolen vstup na rozhraní; jinak to spadlo.

Když chce síťové rozhraní v konfiguraci serveru odeslat paket peerovi (klientovi), podívá se na cílovou IP tohoto paketu a porovná ji se seznamem povolených IP každého peeru, aby zjistil, kterému peeru jej poslat. Pokud je například síťové rozhraní požádáno o odeslání paketu s cílovou IP adresou 10.10.10.230, zašifruje jej pomocí veřejného klíče peer `gN65BkIK...` a poté jej odešle na nejnovější internetový koncový bod tohoto peeru.

V konfiguraci klienta bude jeho jediný peer (server) moci posílat pakety do síťového rozhraní s *libovolnou* zdrojovou IP (protože 0.0.0.0/0 je zástupný znak). Například, když je paket přijat od peer `HIgo9xNz...`, pokud se dešifruje a autentizuje správně, s libovolnou zdrojovou IP, pak je povolen vstup na rozhraní; jinak to spadlo.

V konfiguraci klienta, když chce síťové rozhraní poslat paket svému jedinému peeru (serveru), zašifruje pakety pro jednoho peer s *libovolnou* cílovou IP adresou (protože 0.0.0.0/0 je zástupný znak). Pokud je například síťové rozhraní požádáno o odeslání paketu s libovolnou cílovou IP adresou, zašifruje jej pomocí veřejného klíče jednoho peer `HIgo9xNz...` a poté jej odešle na nejnovější internetový koncový bod jednoho peer.

Jinými slovy, při odesílání paketů se seznam povolených IP chová jako jakási směrovací tabulka a při příjmu paketů se seznam povolených IP chová jako jakýsi seznam řízení přístupu.

To je to, co nazýváme *Cryptokey Routing Table* : jednoduchá asociace veřejných klíčů a povolených IP adres.

Pro kterékoli pole lze použít libovolnou kombinaci IPv4 a IPv6. WireGuard je v případě potřeby plně schopen zapouzdřit jeden do druhého.

Vzhledem k tomu, že všechny pakety odesílané na rozhraní WireGuard jsou šifrované a ověřené, a protože existuje tak těsné spojení mezi identitou peer a povolenou IP adresou peer, správci systému nepotřebují složitá rozšíření firewallu, jako např. IPsec, ale spíše se mohou jednoduše shodovat na „je to z této IP? na tomto rozhraní?“ a mít jistotu, že jde o bezpečný a autentický paket. To značně zjednodušuje správu sítě a řízení přístupu a poskytuje mnohem větší jistotu, že vaše pravidla iptables skutečně dělají to, co jste jim zamýšleli.

Vestavěný roaming

Konfigurace klienta obsahuje *počáteční* koncový bod jeho jednoho peer (serveru), takže ví, kam má odeslat šifrovaná data, než přijme šifrovaná data. Konfigurace serveru nemá žádné počáteční koncové body svých vrstevníků (klientů). Důvodem je, že server zjišťuje koncový bod svých protějšků zkoumáním, odkud pocházejí správně ověřená data. Pokud server sám změní svůj vlastní koncový bod a odešle data klientům, klienti objeví nový koncový bod serveru a aktualizují konfiguraci stejně. Klient i server odesílají šifrovaná data na nejnovější koncový bod IP, pro který autenticky dešifrovali data. Na obou koncích tedy existuje plný IP roaming.

Připraveno pro kontejnery

WireGuard odesílá a přijímá šifrované pakety pomocí síťového jmenného prostoru, ve kterém bylo rozhraní WireGuard původně vytvořeno . To znamená, že můžete vytvořit rozhraní WireGuard ve svém hlavním síťovém jmenném prostoru, který má přístup k internetu, a poté jej přesunout do síťového jmenného prostoru patřícího kontejneru Docker jako jediné rozhraní tohoto *kontejneru* . Tím je zajištěno, že jediný možný způsob, jakým je kontejner schopen přistupovat k síti, je prostřednictvím zabezpečeného šifrovaného tunelu WireGuard.

Další informace

Zvažte pohled na příkazy a rychlý start, abyste získali dobrou představu o tom, jak se WireGuard používá v praxi. Je zde také popis protokolu, kryptografie a výměny klíčů , kromě technického dokumentu , který poskytuje nejpodrobnější informace.

O projektu

Zdrojový kód

WireGuard je rozdělen do několika úložišť hostovaných v ZX2C4 Git Repository a jinde. Prohlédněte si seznam úložišť projektu .

Diskuse IRC

Pokud máte potíže s nastavením nebo používáním WireGuard, nejlepším místem, kde můžete získat pomoc, je `#wireguard`IRC kanál na Libera.Chat . Probíráme tam i rozvojové úkoly a plánujeme budoucnost projektu.

Poštovní seznam

Zapojte se do diskuze o vývoji WireGuard tím, že se připojíte k mailing listu . Zde se odehrávají všechny rozvojové aktivity. Odešlete opravy pomocí `git-send-email`, podobného stylu LKML.

Emailový kontakt

Pokud nás chcete z konkrétního důvodu kontaktovat soukromě, můžete nás kontaktovat na adrese team@wireguard.com . Mějte však na paměti, že požadavky na „podporu“ jsou mnohem vhodnější pro náš IRC kanál.

Bezpečnostní kontakt

Jakékoli problémy se zabezpečením nahlašte a pouze na adresu security@wireguard.com . Neposílejte na tento e-mailový alias problémy, které se netýkají zabezpečení. Neposílejte problémy související se zabezpečením na různé e-mailové adresy.

Licence

Komponenty jádra jsou vydány pod licencí GPLv2, stejně jako samotné jádro Linuxu. Ostatní projekty jsou licencovány pod MIT, BSD, Apache 2.0 nebo GPL, v závislosti na kontextu.