Zranitelnost PuTTY vuln-p521-bias

chiark.greenend.org.uk/~sgtatham/putty/wishlist/vuln-p521-bias.html

shrnutí : Soukromé klíče NIST P521 jsou vystaveny předpojatému generování podpisu
třídy : *zranitelnost:* Toto je bezpečnostní chyba.
priorita : *vysoká:* Toto by mělo být opraveno v příštím vydání.
nepřítomný v : 0,67
přítomný v : 0,68 0,69 0,70 0,71 0,72 0,73 0,74 0,75 0,76 0,77 0,78 0,79 0,80
pevný v : c193fe9872feac9a40650 0,81)
Každá verze nástrojů PuTTY od 0,68 do 0,80 včetně má kritickou zranitelnost v kódu, který generuje podpisy ze soukromých klíčů ECDSA, které používají křivku NIST P521. (PuTTY nebo Pageant generuje podpis z klíče, když jej používá k ověření na serveru SSH.)

Této chybě zabezpečení bylo přiřazeno CVE-2024-31497 . Objevili jej Fabian Bäumer a Marcus Brinkmann z Ruhr University Bochum; viz jejich zápis na mailing listu oss-security .

Špatná zpráva: výsledkem této chyby zabezpečení je **kompromitace soukromého klíče** . Útočník, který vlastní několik desítek podepsaných zpráv a veřejného klíče, má dostatek informací k tomu, aby obnovil soukromý klíč, a pak falšoval podpisy, jako by byly od vás, což mu umožňuje (například) přihlásit se na všechny servery, které tento klíč používáte. pro. K získání těchto podpisů útočníkovi stačí krátce kompromitovat jakýkoli server, ke kterému se pomocí klíče ověřujete, nebo dočasně získat přístup ke kopii Pageant, která klíč drží. (Tyto podpisy však nejsou vystaveny pasivním odposlechům připojení SSH.)

Pokud tedy máte klíč tohoto typu, doporučujeme jej okamžitě odvolat: odstraňte starý veřejný klíč ze všech `authorized_keys`souborů OpenSSH a jeho ekvivalent na jiných serverech SSH, aby podpis z kompromitovaného klíče již neměl žádnou hodnotu. Poté vygenerujte nový pár klíčů, který jej nahradí.

(Problém není v tom, jak byl klíč původně vygenerován; nezáleží na tom, zda pochází z PuTTYgen nebo někde jinde. Důležité je, zda byl někdy *použit* s PuTTY nebo Pageant.)

Dobrá zpráva: *jediným* ovlivněným typem klíče je 521bitový ECDSA. To znamená, že klíč, který se objeví ve Windows PuTTYgen s `ecdsa-sha2-nistp521`na začátku pole 'Key fingerprint', nebo je popsán jako 'NIST p521' při načtení do Windows Pageant, nebo má id začínající `ecdsa-sha2-nistp521`v protokolu SSH nebo souboru klíče. Ostatní velikosti ECDSA a další klíčové algoritmy nejsou ovlivněny. Konkrétně Ed25519 není ovlivněn.

Podrobnosti o chybě:

Všechna podpisová schémata DSA vyžadují, aby byla během podepisování vynalezena náhodná hodnota, známá jako „nonce“ (kryptografický žargon pro hodnotu použitou pouze jednou) nebo někdy pod písmenem *k* . Je dobře známo, že pokud útočník dokáže uhodnout hodnotu *k* , kterou jste použili, nebo najde jakékoli dva podpisy, které jste vygenerovali se stejným *k* , pak může okamžitě obnovit váš soukromý klíč.

To znamená, že je nebezpečné generovat podpisy DSA na systémech bez vysoce kvalitního zdroje náhodnosti. Podstatně *nebezpečnější* než generování šifrovacích klíčů pro jednu relaci: únik soukromého klíče ohrozí mnohem více než jednu relaci SSH.

Z tohoto důvodu, protože PuTTY byl vyvinut na Windows ještě předtím, než měl vůbec nějaký kryptografický generátor náhodných čísel, PuTTY vždy generoval své *k* pomocí deterministické metody, takže náhodná čísla vůbec nepotřebuje. Chytrý trik je vypočítat bezpečný hash, jehož vstup obsahuje zprávu, která má být podepsána, *a také soukromý klíč* . Zabezpečený výstup hash je nerozeznatelný od náhodných dat (nebo jinak hashovací funkce neplní svou práci) a tuto metodu generování nemůže zopakovat útočník, který se snaží zjistit *soukromý* klíč – pokud by dokázal vygenerovat stejný hash zadejte jako vy, už by soukromý klíč měli.

Tato technika je nyní mainstreamová a RFC 6979 dokumentuje specifický dobře známý způsob, jak toho dosáhnout. PuTTY však tuto specifikaci nedodržel, protože jsme začali dělat totéž v roce 2001 a RFC bylo zveřejněno až v roce 2013.

Technika PuTTY fungovala tak, že se vytvořil hash SHA-512 a pak se redukoval mod *q* , kde *q* je pořadí skupiny použité v systému DSA. Pro celočíselné DSA (pro které byla původně vyvinuta technika PuTTY) je *q* asi 160 bitů; pro DSA s eliptickou křivkou (která přišla později) má přibližně stejný počet bitů jako modul křivky, takže 256 nebo 384 nebo 521 bitů pro křivky NIST.

Ve všech těchto případech kromě P521 je zkreslení způsobené snížením 512bitového čísla mod *q* zanedbatelné. Ale v případě P521, kde *q* má 521 bitů (tj. *více* než 512), snížení 512bitového čísla mod *q* nemá vůbec žádný vliv – získáte hodnotu *k* , jejíž horních 9 bitů je vždy nula.

Toto zkreslení je dostatečné pro umožnění útoku na obnovu klíče. Je to méně okamžité, než když útočník zná celý *k* , ale ukazuje se, že pokud má *k* takto zkreslenou distribuci, je možné agregovat informace z více podpisů a nakonec soukromý klíč obnovit. Počet potřebných podpisů je podle všeho kolem 60.

Abychom tuto zranitelnost opravili, zcela jsme opustili starý systém PuTTY pro generování *k* a přešli na techniku RFC 6979 pro všechny typy klíčů DSA a ECDSA. (Klíče EdDSA, jako je Ed25519, již používaly jiný systém, který se nezměnil.) To však neovlivňuje skutečnost, že informace o existujících soukromých klíčích P521 *již* unikly, kdykoli byl podpis generován pomocí starého generátoru *k* .

Pokud chcete tento web okomentovat, podívejte se na stránku Zpětná vazba .
Audit trail pro tuto chybu zabezpečení.
(poslední revize tohoto záznamu o chybě byla 2024-04-15 20:49:32 +0100)