Poodle ranjivost – može li (i treba li) SSL biti ispravljen?

SSL je samo jedan od standarda za siguran prijenos podataka, ali najpoznatiji i vjerojatno najviše korišten. Kolokvijalno se koristi kao zajednički naziv za dva popularna standarda: stariji SSL i mlađi TLS. Masovno se koristi uz web servise i faktički je neizostavan dio implementacije bilo kojeg SaaS rješenja: sve osjetljive informacije danas putuju enkriptirane tim standardom, uključujući i mnoge financijske transakcije.

U teoriji, takvi sustavi osiguravaju visoku razinu zaštite korisničkih podataka. U praksi, riječ je o softveru koji je jednako osjetljiv na programerske pogreške i neotkrivene sigurnosne propuste kao i bilo koji drugi softver. Činjenica da je riječ o sigurnosnom protokolu znači i da je kod temeljitije i stručnije kontroliran od kakve aplikacijice za mobilne uređaje, no to ga neće učiniti apsolutno neprobojnim.

Problem SSL 3.0 protokola koji omogućuje Poodle napad je algoritamske naravi, tj. nije riječ o loše programiranom komadu softvera (što bi bilo relativno trivijalno ispraviti) već o propustu u samom algoritmu prijenosa enkriptiranih podataka.

SSL paket sastoji se od tri dijela: prvi dio su enkriptirani podaci, drugi dio je MAC checksum, dok je treći dio tzv. "padding space", niz posve irelevantnih brojeva koji se nigdje ne koriste i koji završavaju brojem koji označava količinu bajtova tog dijela podataka.

Padding je uveden zato što neke metode enkripcije zahtjevaju točno određenu veličinu bloka kojeg procesiraju, za što služi padding – za popunjavanje praznog prostora.

Proces dekripcije tako uzima cijeli podatak, dekriptira ga i zatim iz zadnjeg bajta saznaje koliko bajtova treba izrezati i odbaciti kako bi ostali samo ispravni, dekriptirani podaci i njihov checksum.

Problem Poodle napada je u tom naizgled bezazlenom smeću na kraju paketa: točnije, u zadnjem bajtu koji ne smije biti slučajan, već mora sadržavati informaciju o broju suvišnih bajtova.

Poodle napad iskorištava tu činjenicu kako bi saznao više o sadržaju napadnutog paketa: napadač uzima originalni paket, odbacuje *padding* bajtove i umjesto njih jednostavno ponavlja enkriptirani sadržaj paketa.

U tom trenutku mogu se dogoditi dvije stvari:

- dekripcijom paketa dobit će se zadnji bajt neispravne veličine, zbog čega se reže prevelik ili premali komad paketa i dobiveni podatak ne odgovara checksumu, te se automatski odbacuje kao neispravan;

- dogodi li se da zadnji bajt bude dekriptiran kao 15, algoritam će ispravno odrezati padding space, dobiveni podatak odgovarat će checksumu i paket će biti prihvaćen.

Napadač i dalje nema pojma o sadržaju povjerljivih podataka: sve što je učinio je zamjena slučajnih brojeva padding bajtova kopijom enkriptiranog podatka iz tog paketa.

No, ono što u tom trenutku napadač može saznati jest činjenica da je vrijednost zadnjeg bajta enkripitranog podatka 15, odnosno da nije 15 (u slučaju da server odbaci paket).

SSL koristi *Cipher Block Chaining*, metodu kojom se paket nastoji dodatno zaštititi tako što se

plaintext sadržaj paketa XOR-a *cyphertext* sadržajem prethodnog paketa prije pakiranja u paket: na taj se način osigurava da čak i dugi nizovi repetitivnog sadržaja izgledaju "razbacano", čime se povećava otpornost na pokušaje probijanja enkripcije, ali u slučaju Poodle napada to napadaču omogućuje da utvrdi sadržaj zadnjeg bajta podatka čak i ako on nema vrijednost 15.

Znati vrijednost samo jednog bajta u jednom od mnogih paketa koji prolaze mrežom nije velika stvar. No, kombinacija *padding* algoritma i CBC metode pokazala se fatalno osjetljivom: ako napadač uspije natjerati korisnikov browser da uporno i iznova šalje isti HTTPS zahtjev (koristeći maliciozni JS na web stranici ili slično), skupljanjem velike količine podataka bit će u stanju sa sigurnošću potvrditi sadržaj određenog bajta podatka.

Nakon toga, izmjenom HTTPS zahtjeva moguće je "ranjivo mjesto" pomaknuti za jedan byte i nakon nekog vremena i taj će podatak biti čitljiv, te napadač može pažljivom manipulacijom HTTPS zahtjeva saznati kompletne podatke, bajt po bajt.

Srećom po korisnike, u ovom slučaju nije riječ o nevidljivom napadu: obzirom da za svaki bajt podatka napadač treba u prosjeku dati 256 zahtjeva (šansa za dobivanje podatka iz paketa je 1:256), napad ostavlja vrlo vidljive tragove, a isto tako moguće ga je i spriječiti.

Problem SSL-a je, kako smo već spomenuli, u tome što je ranjivost na razini algoritma, a ne implementacije: kombinacija potrebe za *padding* prostorom i CBC metode dokazano je ranjiva sama po sebi i nije ju moguće ispraviti bez ozbiljnog narušavanja kompatibilnosti i funkcionalnosti protokola. Drugi način enkripcije (RC4) koji ne koristi fiksnu veličinu paketa također je poznat po "propuštanju" podataka (*RC4 biases*), što faktički obje vrste enkripcije korištene u SSL protokolu čini nepopravljivo nesigurnim.

Srećom po nas, postoje i drugi protokoli neosjetljivi na taj napad, ponajprije već dugo vremena korišteni TLS, nasljednik SSL-a: neosjetljiv je na Poodle napad, a i vrlo raširen. Nažalost, neke je korisničke aplikacije (kao i neke embedded uređaje) moguće natjerati da se spuste sa TLS protokola na ranjivi SSL protokol i tako efektivno otvore bokove za Poodle napad. Zbog toga je potrebno u svim aplikacijama koje dozvoljavaju promjenu protokola onemogućiti korištenje SSL-a.

Je li ovo posljednji čavao u lijesu SSL-a? Sudimo li po ozbiljnosti i nepopravljivosti propusta, jest. SSL će uskoro postati prošlost u modernim browserima i serverima; nažalost, sudbina embedded sustava ostat će ista: prepušteni volji svojih proizvođača, mnogi od njih neće biti pokrpani, u kojem slučaju jedina zaštita ostaje zabrana SSL-specifične konekcije na izlazu lokalne mreže.

A to je razlog više za kupnju ***OpenWRT*** kompatibilnih uređaja.

pon, 2014-10-27 06:48 - Radoslav Dejanović**Vijesti:** Sigurnosni propusti [1]
Kategorije: Servisi [2]
Vote: 0

No votes yet

Source URL: https://sysportal.carnet.hr/node/1446

Links
[1] https://sysportal.carnet.hr/taxonomy/term/14
[2] https://sysportal.carnet.hr/taxonomy/term/28