Trostruki bypass za OpenSSL

Život piše romane, pa započnimo ovaj tekst prigodnom teorijom zavjere: "Nakon što je Snowden otkrio svijetu razmjere špijuniranja NSA, programeri širom svijeta krenuli su u žurno pregledavanje izvornog koda u potrazi za namještenim sigurnosnim propustima. Pao je Apple, zatim GnuTLS, a nedugo zatim i OpenSSL."

Teorija zavjere ili ne, činjenica je da smo u relativno kratko vrijeme doživjeli otkriće tri ozbiljna sigurnosna propusta, od kojih je ovaj posljednji, tzv. "heartbleed [1]" uistinu prodrmao cijeli Internet.

Za razliku od "goto fail" prolema tvrke Apple i sličnog primjera lošeg programiranja u slučaju GnuTLS biblioteke, pogreška u OpenSSL-u daleko je manje trivijalna i teže ju je (bilo) otkriti.

Problem se krije u tzv. "heartbeat" mehanizmu, točnije ekstenziji TLS i DTLS protokola; njegova je uloga rasteretiti TLS i DTLS od dužnosti očuvanja veze (keep-alive [2]) i izračunavanja veličine MTU [3] – ako je uključena, Heartbeat opcija će se pobrinuti da izračuna MTU i zatim održava vezu "na životu" kroz keep-alive pakete čak i ako u komunikacijskom kanalu duže vremena nema razmjene podataka.

Da bi Heartbeat mogao izračunati optimalan MTU, paket treba biti promjenjive veličine i njegov tzv. "payload" je sadržaj koji je u načelu nebitan, ali se od suprotne strane očekuje da vrati paket sa identičnim sadržajem, obično nekim serijskim brojem koji služi kao brojač.

Problem se krije upravo u rukovanju tim sadržajem: dovoljan je bio jedan jedini "boundary check [4]" propust da server urbi et orbi otkrije sadržinu naočigled zaštićene komunikacije: maliciozni napadač može lukavo poslati sasvim normalan Heartbeat paket sa kratkim payload sadržajem, ali promjenjen tako da u polje koje definira dužinu payload informacije upiše ne stvarnu dužinu informacije koju šalje, već mnogo veću – moguće je definirati dužinu informacije do 64KB, dok veličina payload informacije ne mora biti veća od šačice bajtova.

Kad tako promjenjen paket dođe do servera, zadaća druge strane je da odgovori identičnim paketom; međutim, kako OpenSSL ne provjerava stvarnu dužinu payload informacije, dogodit će se neobična stvar: server će nazad vratiti paket u kojem će biti izvorna payload informacija, ali će vratiti i na nju nalijepljen sadržaj radne memorije koji se nalazi odmah iza memorijske lokacije na kojoj je pospremljen payload sadržaj, i to u količini koja je potrebna da se zadovolji veličina deklarirana u malicioznom paketu.

Na taj način prilikom napadač će za svaki poslani maliciozni paket nazad dobiti paket u kojem će biti do 64K sadržine RAM memorije servera.

Podaci koje server šalje su različiti i ovise o tome gdje se u memoriji računala nalazio originalni payload, pa tako svaki maliciozni paket uzrokuje vraćanje, recimo to tako, slučajnog isječka RAM memorije.

Problem je što u tim slučajnim komadićima informacija može biti svašta, od dijelova posve neupotrebljivih informacija pa sve do lako dohvatljivih lozinki korisnika, dijelova njegove komunikacije ili, što je najopasnije, ključeva pa i samih certifikata.

Drugi, možda još gori problem je što ovo neispravno ponašanje ne izaziva nikakvo okidanje IDS sustava, logiranje, upozorenje, ništa! Problem je neprimjećen postojao na mreži već dvije godine, što je dovoljno vremena da strpljivi napadač tiho i posve skriveno skupi gomilu podataka sa servera, a s

rastućom hrpom podataka rasla je i vjerojatnost da će server u nekom trenutku napadaču "na izvol'te" predati i sigurnosno vrlo osjetljive podatke.

Da bi ružna priča postala još ružnija, podsjetimo se kako slanje korisničkih lozinki nije najgora stvar: ako se napadač dočepao certifikata servera, bio je u prilici složiti savršeni MiTM [5] napad jer bi mogao klijentu podmetnuti originalni certifikat kao svoj, pa ni korisnik ni antivirusni/antimalware softver ne bi shvatio kako je riječ o prevari. Napadač bi to iskoristio da, tunelirajući promet prema pravom serveru preko sebe, "oljušti" komunikaciju od enkripcije i tako dobije mogućnost uvida u sav promet između korisnika i servera, ili čak "preotme [6]" pristup (za što čak niti nije potrebno incirati MiTM napad, niti biti na istoj mreži kao i žrtva - što je bio nužan preduvjet za famozni FireSheep [7], potres koji je svojedobno također prodrmao Internet, istina puno manje magnitude od ovog.

Očito je, dakle, da ovaj sigurnosni incident nije tako jednostavne naravi kao ona dva sa početka teksta: u ovom slučaju riječ je o curenju podataka sa otprilike 2/3 servera na Internetu, a strku ste vjerojatno već primjetili jer su ozbiljne stranke već požurile zakrpati OpenSSL, ali i izdati posve nove certifikate – jer je nemoguće sa sigurnošću reći je li u međuvremenu neki certifikat "provaljen" ili nije.

Što nam je za činiti?

Administratori bi trebali obavezno učiniti ove tri stvari:

1. patchirati OpenSSL (to ste vjerojatno već sredili kroz sigurnosni update);

2. povući sve stare certifikate i izdati nove; ovo se ne odnosi samo na certifikate za strojeve koji su okrenuti prema korisniku (HTTPS), već i one koji se koriste za, primjerice, komunikaciju servera sa bazom podataka sa serverom na kojem je Web servis; drugim riječima, trebate povući i ponovo izdati zaista sve certifikate;

3.restartati web i mail servise.

Ovaj je propust relativno mlađe naravi (postoji od OpenSSL 1.0.1), pa serveri koji imaju stariju verziju ili ne koriste OpenSSL nisu osjetljivi na napad.

Ako nekim slučajem niste u mogućnosti napraviti upgrade OpenSSL-a (primjerice, postoje "pametni" uređaji na kojima to nije moguće), nužno je koristiti odgovarajuća IDS pravila poput ovih za Snort [8]:

```
alert tcp any [!80,!445] -> any [!80,!445] (msg:"FOX-SRT - Suspicious - SSLv3 Large H
eartbeat Response"; flow:established,to_client; content:"|18 03 00|"; depth: 3; byte_
test:2, >, 200, 3, big; byte_test:2, <, 16385, 3, big; threshold:type limit, track by
_src, count 1, seconds 600; reference:cve,2014-0160; classtype:bad-
unknown; sid: 1000000; rev:4;)
alert tcp any [!80,!445] -> any [!80,!445] (msg:"FOX-SRT - Suspicious - TLSv1 Large H
eartbeat Response"; flow:established,to_client; content:"|18 03 01|"; depth: 3; byte_
test:2, >, 200, 3, big; byte_test:2, <, 16385, 3, big; threshold:type limit, track by
_src, count 1, seconds 600; reference:cve,2014-0160; classtype:bad-
unknown; sid: 1000001; rev:4;)
alert tcp any [!80,!445] -> any [!80,!445] (msg:"FOX-SRT - Suspicious - TLSv1.1 Large
 Heartbeat Response"; flow:established,to_client; content:"|18 03 02|"; depth: 3; byt
e_test:2, >, 200, 3, big; byte_test:2, <, 16385, 3, big; threshold:type limit, track
by_src, count 1, seconds 600; reference:cve,2014-0160; classtype:bad-
unknown; sid: 1000002; rev:4;)
alert tcp any [!80,!445] -> any [!80,!445] (msg:"FOX-SRT - Suspicious - TLSv1.2 Large
 Heartbeat Response"; flow:established,to_client; content:"|18 03 03|"; depth: 3; byt
e_test:2, >, 200, 3, big; byte_test:2, <, 16385, 3, big; threshold:type limit, track
by_src, count 1, seconds 600; reference:cve,2014-0160; classtype:bad-
unknown; sid: 1000003; rev:4;)
```

Na ovoj adresi:

http://submeet.net/tools/heartbleed.php [9]

nalazi se zgodan servis kojim možete provjeriti je li vaš server osjetljiv na napad, odnosno jeste li ga dobro pokrpali.

pet, 2014-04-11 07:17 - Radoslav Dejanović**Vijesti:** Sigurnost [10]
Kategorije: Servisi [11]
Vote: 0

No votes yet

Source URL: https://sysportal.carnet.hr/node/1382

Links
[1] http://heartbleed.com
[2] http://en.wikipedia.org/wiki/Keepalive
[3] http://en.wikipedia.org/wiki/Path_MTU_Discovery
[4] http://en.wikipedia.org/wiki/Bounds_checking
[5] http://en.wikipedia.org/wiki/Man-in-the-middle_attack
[6] https://www.mattslifebytes.com/?p=533
[7] http://en.wikipedia.org/wiki/Firesheep
[8] http://blog.fox-it.com/2014/04/08/openssl-heartbleed-bug-live-blog/
[9] http://submeet.net/tools/heartbleed.php
[10] https://sysportal.carnet.hr/taxonomy/term/13
[11] https://sysportal.carnet.hr/taxonomy/term/28