Prljava krava

Trend imenovanja događaja i vremenskih pojava proširio se i na svijet informatičke sigurnosti, pa osim tropskih oluja, ciklona i inih pošasti, imena dodjeljujemo i sigurnosnim propustima.

Nije to novost u svijetu računala: razvojne verzije operacijskih sustava imaju svoja kodna imena [1], pa čak i Linux kerneli imaju svoje šašave nazive [2]; zašto onda i sigurnosni propusti ne bi dobili simpatično ime, pa da ih tako možda lakše prebolimo?

Dirty Cow je zapravo poprilično dosadan sigurnosni problem... tehnički nimalo spektakularan, reći ćemo klasičan primjer eskalacije privilegija lokalnog korisnika. Svojim dosadnim tehničkim nazivom CVE-2016-5195 [3] i činjenicom da se radi o relativno jednostavnom "race condition [4]" propustu koji omogućuje korisničkom procesu pisanje po segmentu memorije koji bi u normalnim prilikama trebao biti otvoren samo za čitanje. Propust je otkriven u *copy-on-write* [5] operaciji čija implementacija u Linux kernelu dozvoljava korisničkoj aplikaciji da forsiranjem sistemskog poziva madvise *[6]()* u spretnom slučaju dobiju željeni komad memorije sa *read-write* pristupom, što otvara mogućnost zapisivanja podataka u tuđi mapirani kod: proces može mijenjati datoteku koja pripada rootu ili nekom drugom privilegiranom)korisniku i zatim to iskoristiti da svom korisniku da veće privilegije od onih koje pripadaju napadnutom procesu.

No, postoje neke sitnice koje ovaj sigurnosni propust čine zbilja zanimljivim i opravdavaju davanje posebnog imena, nadimka uz klasičnu CVE notifikaciju.

Ovaj je sigurnosni propust poznat Linux developerima već jedanaest godina. Problem je uočen još 2005. godine kad je i ispravljen, no zbog ozbiljnog problema kojeg je propust izazvao na s390 [7] računalima od zakrpe se odustalo, računajući tada da je potencijalni *race condition* više teorijske naravi nego što je praktičo izvediv.

Jedanaest godina je mnogo vremena, u što smo se imali prilike uvjeriti nedavnim odustajanjem od cijelog niza enkripcijskih algoritama, koji su u tom razdoblju prešli put od teoretski ranjivih do posve praktično provaljivih. Tako su i na ovaj propust vjerojatno i *developeri* zaboravili, dok se nedavno nisu pojavile demonstracije [8] koda koji iskorištava tu ranjivost zloupotrebljavajući /proc/self/mem (reprezentacija memorije pokrenutog procesa kao datoteke) i PROC_POKEDATA ptrace [9] zahtjev, a navodno je iskorištavanje
propusta primjećeno i "u divljini".

Rješenje problema, srećom, posve je jednostavno: instalirajte novi kernel koji ima odgovarajuću zakrpu. Red Hat je izdao i zasebno privremeno rješenje [10] za korisnike njihovih distribucija koji nisu u mogućnosti promjeniti kernel na računalima u pogonu: ova zakrpa isključuje ptrace, pa niti normalni programi poput debuggera ili antivirusa kojima je ptrace nužno potreban za ispravan rad neće funkcionirati, pa tako sprečava izvršavanje napada koji su uočeni "u divljini". No to je samo privremeno rješenje: pravo rješenje je osvježenje kernela na verziju u kojoj je ovaj propust ispravljen.

Najveći utjecaj ovaj bi propust, pak, mogao imati na manje očekivanom mjestu: među korisnicima Android uređaja!!

Trenutno su baš svi Android uređaji osjetljivi na ovaj sigurnosni propust - i to je, rekao bih, daleko, daleko veći problem od PC računala: računala ćemo lako i brzo osigurati, dapače ažurni administratori vjerojatno već jesu preuzeli novi kernel i instalirali ga na svoja računala, no mobilne telefone, tablete, IoT i druge uređaje puno je teže osigurati zbog jednostavne činjenice da su

korisnici osuđeni na update kojeg trebaju izdati proizvođači uređaja, a koliko često oni to (ne) čine već je postala notorna činjenica.

S jedne je strane to zastrašujuća pomisao: ogroman broj Android računala ostat će ranjiv na ovaj neugodan i ne pretjerano kompliciran napad koji malicioznom softveru otvara mogućnost infekcije, pri čemu je jedina stvar koja računalo štiti od provale svijest korisnika. Da bi se propust iskoristio, naime, potrebno je instalirati i sa lokalnog medija pokrenuti aplikaciju. To znači da "običnim surfanjem" vjerojatno nećete pokupiti malware, ali dovoljno je samo na trenutak zavarati korisnika i nagovoriti ga da preuzme i instalira malicioznu aplikaciju (koja čak može biti i sa Google Play-a jer nema jednostavnog načina da antivirusni algoritmi u ovom slučaju otkriju maliciozne namjere unutar koda aplikacije), da bi računalo postalo širom otvoreno potencijalnom napadaču.

Zaštita na Android računalima tako se svodi samo na poruku korisniku: "Pamet u glavu!". Poruka sistemašima u BYOD radnim okruženjima: "Sretno!"

No, s druge je strane ovaj propust pravi blagoslov za napredne korisnike Android uređaja koji žele puni pristup uređaju: tzv. "rootanje [11]". Do sad je to bilo ograničeno specifičnim implementacijama softvera različitih proizvođača, a u osnovi se svodilo na traženje specifičnih propusta u OS-u pojedinih proizvođača i određenih modela telefona. Sada smo dobili niverzalni sigurnosni propust koji bi u teoriji trebao biti izvediv na svakom Android uređaju, pa bi sve takve uređaje sad bilo moguće "otključati". To, naravno, ne umanjuje opasnost od malicioznog preuzimanja uređaja od strane aplikacije za "rootanje, ali će ipak razveseliti zajednicu ljudi koji vole hackirati Android uređaje ili jednostavno imati nad njima potpunu kontrolu.

ned, 2016-10-30 13:36 - Radoslav Dejanović**Vijesti:** Sigurnosni propusti [12]
Kategorije: Sigurnost [13]
Vote: 0

No votes yet

Source URL: https://sysportal.carnet.hr/node/1695

Links
[1] https://namingschemes.com/Windows_codenames
[2] https://en.wikipedia.org/wiki/List_of_Linux_kernel_names
[3] https://access.redhat.com/security/cve/CVE-2016-5195
[4] https://en.wikipedia.org/wiki/Race_condition
[5] http://stackoverflow.com/questions/628938/what-is-copy-on-write
[6] http://man7.org/linux/man-pages/man2/madvise.2.html
[7] https://en.wikipedia.org/wiki/IBM_ESA/390
[8] https://github.com/dirtycow/dirtycow.github.io/wiki/PoCs
[9] https://en.wikipedia.org/wiki/Ptrace
[10] https://bugzilla.redhat.com/show_bug.cgi?id=1384344#c13
[11] https://en.wikipedia.org/wiki/Rooting_(Android_OS)
[12] https://sysportal.carnet.hr/taxonomy/term/14
[13] https://sysportal.carnet.hr/taxonomy/term/30