Heartbleed

Vahid Ramazani

Shahid Beheshti University

imo.ramazani@gmail.com

چکیده

حفره امنیتی Heartbleed که در ماه آوریل 2014 (فروردین 1393) در اینترنت پدیدار گشت. این حفره که از یکی از مهمترین‌ها از زمان پیدایش اینترنت است، باعث شد مهاجمان بتوانند – از طریق HTTPS – حافظه اصلی حدود یک چهارم سایت‌های اینترنتی را بخوانند[1]. در این مقاله به معرفی اجمالی این حفره میپردازیم. شامل : (1) معرفی سطح بالا و جدای از پیچیدگی مساله ، (2) زمینه، دلیل و رویدادهای حول Heartbleed ، (3) تاثیر آن ، (4) پیشگیری‌های آینده.

1. معرفی

این حفره یک اشکال پیاده سازی در کتابخانه رمزنگاری محبوب و فراگیر OpenSSL است که به مهاجم اجازه میدهد حافظه رایانه ای که در حال اجرای این نرم افزار است را بخواند. بعلاوه مهاجم میتواند از این طریق کلیدهای خصوصی SSL را بازیابی کند [2]. این مشکل به صورت مستقل توسط سه مهندس امنیت در Codenomicon و Neel Mehta از امنیت گوگل در 7 آوریل 2014 (18 فروردین 1393) کشف شد [3]. خطر این حفره – که به دلیل وجود اشکال در کتابخانه OpenSSL به وجود آمد – به سبب استفاده وب سرورهای محبوبی مثل Apache و Nginx اینترنت و کاربرانش را تحت شعاع قرار داد [4].

2. زمینه

روز 7 آوریل 2014، پروژه OpenSSL علنا این حفره را اعلام کرد؛ این حفره اشکالی در پیاده سازی افزوه Heartbeat (extension) برای TLS است. حفره مذکور به مهاجم اجازه گرفتن اطلاعات حساسی شامل کلیدهای خصوصی رمزنگاری، در هر دو سمت سرور و کلاینت، را میدهد[5]. در این بخش نگاهی به تاریخچه OpenSSL، افزونه ایراد دار آن، حفره و جدول زمانیش میکنیم.

2.1 تاریخچه کوتاهی از OpenSSL

کتابخانه محبوب و متن باز OpenSSL پروتکل‌های امنیتی SSL و TLS را پیاده سازی میکند. این کتابخانه به صورت گسترده ای توسط سرور ها و همینطور دستگاه های سمت کاربر استفاده شده است. موارد استفاده از آن بسیار فراگیر است که شامل Web, Email, VPN و سرویس های پیام رسان میشود. پروژه OpenSSL از سال 1998 شروع شد که خود پروژه بعد از دوسال شروع به مستندسازی حفره ها و ایرادهایش کرد. برای لیست حفره‌های این پروژه رجوع کنید به [6].

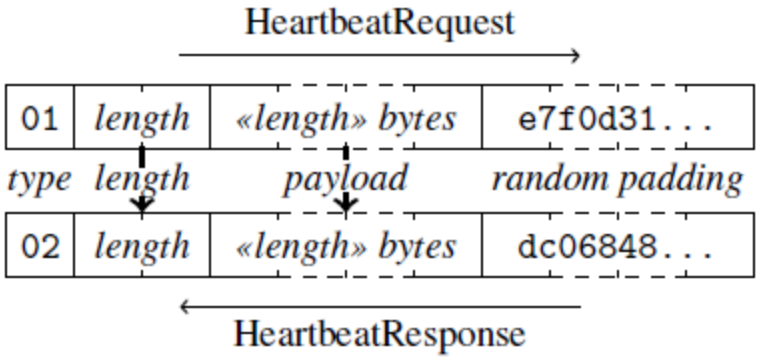
حفره Heartbleed به چند دلیل یکی از مهمترین و تاثیرگذارترین حفره های شناخته شده است: اول اینکه به مهاجم دسترسی به کلید خصوصی و متعاقبا اطلاعات و دیتای شخصی کاربران را میدهد، دوم

اینکه بکاربردن و بهره برداری(exploit) از این حفره آسان است و سوم اینکه سرویس های رمزنگاری بر پایه TLS مثل HTTPS به طور فزاینده ای فراگیر شده اند که گستردگی آلودگی را افزایش داده است.

2.2 افزونه Heatbeat

این افزونه به دو سمت TLS امکان تشخیص حضور دیگری را میدهد؛ انگیزه این عمل، مدیریت session در DatagramTLS بوده است و طبیعتا نسخه های استاندارد TLS – به دلیل استفاده از TCP – نیازی به این افزونه ندارند.

داشتن یا عدم وجود این افزونه در طرفین، هنگام عمل دست دادن(handshaking) در TLS اعلام میشود که در صورت وجود، هر کدام از طرفین میتواند یک پیغام **HeartbeatRequest** بفرستد تا اتصال را با دریافت **HeartbeatResponse** تایید کند (شکل 1). این افزونه در فوریه 2012 در RFC 6520 [7] معرفی و در نسخه 1.0.1 از OpenSSL در مارس 2012 منتشر شد.



شکل 1

2.3 حفره Heartbleed

حفره موجبه از پیاده سازی های ایراد دار OpenSSL – که شامل نسخه های 1.0.1 و 1.0.2-beta میشود [5] – به طرفین اجازه خواندن حافظه‌ی بعد از payload را میدهد. حافظه قابل دسترسی با دادن اندازه ای بیشتر از آنچه در HeartbeatRequest میسر میشود. چون قسمت payload فقط دو بایت است، پاسخ دریافتی حداکثر حاوی 64KB از حافظه است. نکته سادگی استفاده اشاره شده اینجا آشکار میشود: در این پیاده سازی رایانه قربانی به اندازه مشخص شده در درخواست مهاجم اعتماد میکند.

در نسخه های بعدی OpenSSL که این ایراد رفع شد، این اندازه توسط رایانه دریافت کننده بررسی و اندازه مشخص شده توسط فرستنده نادیده گرفته میشود. به این ترتیب این مشکل از بین میرود. [8]

2.4 جدول زمانی Heartbleed

حفره ذکر شده، ابتداعا توسط مهندس امنیت گوگل Neel Mehta در مارس 2014 کشف شد[9]. این اتفاق منجر به اصلاح و وصله کردن رایانه های سرور گوگل و بعد از آن اطلاع رسانی به تیم پروژه OpenSSL در روز اول آوریل – کمتر از یازده روز – شد. به صورت مستقل تیم مشاور امنیتی Codenomicon روز دوم آوریل این مشکل را پیدا کرده و به مرکز ملی امنیت سایبری فنلاند(NCSC-

FI) اعلام کرد. تیم OpenSSL که از دو منبع مستقل این کشف را دریافت کرده بود، درصدد ساختن وصله اصلاحی برآمد.

پس از آماده شدن و بررسی وصله اصلاح شده – نسخه 1.0.1g - تیم OpenSSL به صورت علنی این مشکل را اعلام و راه حل آن را در 7 آوریل ذکر کرد. قبل از آگاه کردن عموم از طریق mailing list، شرکت‌هایی از جمله Facebook, CloudFlare, Debian, RedHat از این مساله آگاهی داشتند. همینطور Linux Foundation کمتر از 24 ساعت قبل از اعلان عمومی در مورد این مشکل تذکر لازم را دریافت کرد. سایرین مثل Ubuntu, Chromium, Cisco قبل از اعلان عمومی آگاه نشدند[10].

جدول2: رویدادها به ترتیب روز

03/21 Neel Mehta of Google discovers Heartbleed
03/21 Google patches OpenSSL on their servers
03/31 CloudFlare is privately notified and patches
04/01 Google notifies the OpenSSL core team
04/02 Codenomicon independently discovers Heartbleed
04/03 Codenomicon informs NCSC-FI
04/04 Akamai is privately notified and patches
04/05 Codenomicon purchases the heartbleed.com domain
04/06 OpenSSL notifies several Linux distributions
04/07 NCSC-FI notifies OpenSSL core team
04/07 OpenSSL releases version 1.0.1g and a security advisory
04/07 CloudFlare and Codenomicon disclose on Twitter
04/08 Al-Bassam scans the Alexa Top 10,000
04/09 University of Michigan begins scanning

3. تاثیر Heartbleed

قسمت 2.3 سادگی و در عین حال اهمیت این حفره را نمایان میکند. چنان ساده که بهره برداری از آن نیاز به امکانات خاصی ندارد و از طرفی، اصلاح کردن مشکل هم ساده بود – چنانکه گوگل در همان روز تشخیص، سرورهایش را اصلاح کرد – ولی تاثیر زیاد و خطرناک این مشکل وقتی آشکار میشود که نکته مهمی را بدانید: هر سرویسی که از OpenSSL ایراد دار برای TLS استفاده میکند، به این حفره آلوده است. از دسته این سرویس ها میتوان به: وب (از طریق HTTPS)، ایمیل، پایگاه داده‌ها و پیام رسان ها اشاره کرد. برای نمونه ای از گستره آلودگی و اهمیت، جدول 3 را ببینید، حاصله از [1].

Web Servers	**Mail Servers**	**Database Servers**	**XMPP Servers**	**Other Servers**
Apache (mod_ssl) Yes	Sendmail Yes	MySQL Yes	OpenFire No	OpenVPN Yes
Microsoft IIS No	Postfix Yes	PostgreSQL Yes	Ejabberd Yes	OpenLDAP Yes
Nginx Yes	Qmail Yes	SQL Server No	Jabberd14 Yes	Stunnel Yes
Lighttpd Yes	Exim Yes	Oracle No	Jabberd2 Yes	Openswan Yes
Tomcat Yes	Courier Yes	IBM DB2 No		Telnetd-ssl Yes
Google GWS Yes	Exchange No	MongoDB Yes		OpenDKIM Yes
LiteSpeed Yes	Dovecot Yes	CouchDB No		Proftpd Yes
IBM Web Server Yes	Cyrus Yes	Cassandra No		Bitcoin Client Yes
Tengine Yes	Zimbra Yes	Redis No		
Jetty No				

جدول3

طبق بررسی‌های Codecomicon [3] حدود 66% از وبسایت‌های استفاده کننده از HTTPS – که HTTP over TLS است – به این حفره آلوده و آسیب پذیر بوده اند. البته این تقریب از روی رواج وب سرورهای معروف Apache و Nginx بوده و شاید اغراق باشد، زیرا امکان دارد مسئولان وب سرورهای مختلف از طرق متنوعی مثل غیرفعال کردن افزونه، استفاده از کتابخانه SSL متفاوت یا نسخه قدیمی‌تر OpenSSL که آسیب پذیر نبوده، به این حفره آلوده نشده بوده باشند.

3.1 تاثیر روی سایت‌های محبوب

همه‌ی 100 سایت اول(رتبه بندی Alexa) در کمتر از 48 ساعت پس از اعلان عمومی این ایراد را برطرف کردند. در این بین منابعی همچون سایت Mashable لیستی از سایت‌هایی که به این آلودگی دچار بوده و اصلاح شده اند ارائه کردند؛ این مساله از آنجایی که احتمال سرقت کلیدهای خصوصی کاربران میرفت اهمیت داشت. زیرا در اینصورت باید کلیدهای جدید ساخته میشد. بنابراین منابع مختلفی به کاربران سایت‌های آسیب پذیر که اصلاح شده اند، توصیه کردند که رمز کاربری خود را عوض کنند تا در صورتی که کلیدها به سرقت رفته شده باشند، خطر آتی ای پیش نیاید. [11]

لیست 30 سایت محبوب – استفاده از بستر HTTPS

Site	Vuln.	Site	Vuln.	Site	Vuln.
Google	Yes	Bing	No	Wordpress	Yes
Facebook	No	Pinterest	Yes	Huff. Post	?
Youtube	Yes	Blogspot	Yes	ESPN	?
Yahoo	Yes	Go.com	?	Reddit	Yes
Amazon	No	Live	No	Netflix	Yes
Wikipedia	Yes	CNN	?	MSN.com	No
LinkedIn	No	Instagram	Yes	Weather.com	?
eBay	No	Paypal	No	IMDB	No
Twitter	No	Tumblr	Yes	Apple	No
Craigslist	?	Imgur	Yes	Yelp	?

3.2 وصله قبل از افشا

قبل از افشا و اعلان عمومی، گوگل، Akamai و سایت‌هایی که قبل از عموم از این آسیب پذیری اطلاع داشتند، افزونه مربوطه را غیرفعال کردند. برای زمان غیرفعال شدن این افزونه در این سرویس‌ها از نتایج [12] – که داده‌هایی از ICSI Certificate Notary (دفتر اسناد گواهی‌ها) را بررسی کرده، که خود این دفتر ارتباطات TLS را از هفت نقطه تحقیق و شبکه دانشگاهی زیرنظر دارد – استفاده میکنیم: اطلاعات دفتر نشان میدهد که گوگل حداقل 12 روز قبل از اعلان عمومی افزونه مربوطه را غیرفعال کرده یا در بعضی موارد با وجود داشتن این افزونه، امکان بهره برداری و تهاجم (به دلیل تنظیمات مرتبط) وجود نداشته است. با این حال نهایتا تا 15 آوریل، گوگل به عنوان یک مقررات سازمانی، افزونه‌های همه سرورهایش را غیرفعال کرد. مشابها Akamai، 4 روز قبل از اعلان این عملیات را شروع و تا 18 آوریل به اتمام رساند.

4. پیشگیری‌های آینده به منظور عدم تکرار حفره‌های مشابه

آسیب پذیری امنیتی Heartbleed، اهمیت وجود Forward Secrecy – یعنی داشتن امنیت رو به جلو، که فاش شدن کلیدهای یک دسته از ارتباطات، باعث آشکار شدن ارتباطات آتی نمیشود – را پررنگ‌تر کرد. این امکان در TLS، توسط استفاده از رمزنگاری‌های تعبیه شده ای – از جمله ephemeral Diffie–Hellman (TLS_DHE) - ، وجود دارد [13].

بدون این روش رمزنگاری، فاش شدن کلید خصوصی سرور، موجب باز شدن رمز همه‌ی ارتباطات رمزنگاری شده قبلی یا بعدی که از آن کلید خصوصی استفاده کرده اند، میشود؛ این فرآیند توسط مهاجمی که این ارتباطات را ضبط و در زمانی به این کلید دستیابی پیدا کند قابل اجراست. بدیهتا اینچنین اتفاق یک فاجعه به بار خواهد آورد.

بنابراین، انتظار میرود، اپراتورهای سرورها، از این روش رمزنگاری استفاده کنند تا در صورت اتفاق مشابهی در آینده، مشکلات به وجود آمده حداقلی باشد.

5. نتیجه گیری

در این مقاله نگاهی اجمالی به Heartbleed و تاثیرات خطرناک آن در عین سادگی داشتیم و مشاهدات گسترده حاصله از مراجع مختلفی را بررسی کردیم؛ اشتباهی ساده در یک پیاده سازی فراگیر، حتی متن باز، میتواند فاجعه بار باشد. چنانکه بین یک چهارم تا نصف سرویس‌های تحت وب را تحت الشعاع قرار دهد. مسائل اینچنینی به قدری سهمگین هستند که تا مدت ها طرفین دخیل را درگیر خود میکنند، همچنان که 3% از یک میلیون سایت برتر Alexa تا یک ماه بعد از اعلان عمومی هنوز آسیب پذیر بودند.

راه حل عنوان شده استفاده از روش‌های بازدارنده در صورتی که رمز افشا شود، مثل Forward Secrecy ، در کنار هشیاری همیشگی و واکنش به موقع و درست به ابزارهای مورد استفاده است.

6. منابع

[1] Durumeric, Zakir, et al. "The matter of heartbleed." *Proceedings of the 2014 Conference on Internet Measurement Conference*. ACM, 2014. (dx.doi.org/10.1145/2663716.2663755)

[2] *"Security Advisory – OpenSSL Heartbleed Vulnerability" Cyberroam*

[3] N. Mehta and Codenomicon. The Heartbleed Bug. heartbleed.com

[4] Apache and Nginx Git Repository

[5] CVE-2014-0160: openssl.org/news/secadv/20140407.txt

[6] https://www.openssl.org/news/vulnerabilities.html

[7] R. Seggelmann, M. Tuexen, and M. Williams. Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS) Heartbeat Extension. IETF Request for Comments (RFC) 6520, February 2012.

[8] A. Ellis. Akamai heartbleed Update (V3), Apr. 2014. https://blogs.akamai.com/2014/04/heartbleed-update-v3.html.

[9] Heartbleed F.A.Q., 2014. https://www.startssl.com/?app=43.

[10] B. Grubb. Heartbleed Disclosure Timeline: Who Knew What and When. Apr. 2014. http://www.smh.com.au/it-pro/security-it/heartbleed-disclosure-timeline-who-knew-whatand-when-20140415-zqurk.html.

[11] The Heartbleed Hit List: The Passwords You Need to Change Right Now, Apr. 2014. http://mashable.com/2014/04/09/heartbleed-bug-websites-affected/.

[12] B. Amann, M. Vallentin, S. Hall, and R. Sommer. Extracting Certificates from Live Traffic: A Near Real-Time SSL Notary Service. Technical Report TR-12-014, ICSI, Nov. 2012.

[13] SSL: Intercepted today, decrypted tomorrow, https://news.netcraft.com/archives/2013/06/25/ssl-intercepted-today-decrypted-tomorrow.html