Mittuniversitetet
MID SWEDEN UNIVERSITY

Laboration: Tillgänglighetsattacker

Daniel Bosk och Lennart Franked*

sda.tex 933 2013-04-15 13:18:47Z danbos

Innehåll

1 Introduktion **1**

2 Syfte **2**

3 Läsanvisningar **2**

4 Genomförande **2**
4.1 Inledande konfiguration 2
4.2 Inställningar för att simulera en tillgänglighetsattack 3

5 Examination **4**

1 Introduktion

Denna laboration syftar till att ge en grundläggande förståelse för begreppet tillgänglighetsattacker, eller Service Denial Attack (SDA) och Denial of Service (DoS). Det praktiska momentet i laborationen omfattar installation och konfigurering av en webbserver och slutligen ett praktiskt försök att göra en attack mot denna webbserver.

Tillgänglighetsattacker är ett gemensamt begrepp för en samling av attacker som alla syftar till att försätta en viss server eller tjänst ur funktion. Detta görs vanligen genom att man skickar en stor mängd förfrågningar till servern som därmed blir överbelastad och inte längre kan svara på vanliga anrop. Värt att tillägga här är dock att det inte alltid är en tillgänglighetsattack som ligger till grund för att en webbserver går ner, utan det kan hända att en webbplats får mer legitim trafik än den kan hantera, detta är känt som slashdot-effekten[1].

*Inspirerat av en tidigare laboration av Curt-Olof Klasson och Rahim Rahmani.

[1] URL: `http://en.wikipedia.org/slashdot_effect`.

2 Syfte

Syftet med denna laboration är:

- Att du ska få insikt i hur en Service Denial Attack fungerar.
- Att du ska kunna implementera grundläggande åtgärder för att skydda sig mot detta.

3 Läsanvisningar

Ni ska ha läst kapitlet *Network Attack and Defence* i Anderson [1, 2] Ni ska även känna till följande begrepp:

- (D)SDA, (D)DoS
- process
- tråd
- underprocess
- Telnet
- loopback

4 Genomförande

Ni kommer i denna laboration att använda er utav följande program:

- Apache Hypertext Transfer Protocol Server, och
- telnet(1).

Första delen i denna laboration kommer att handla om att konfigurera en webbserver. Vi kommer att använda oss av Apache, men vill ni använda något annat program är ni välkomna att göra detta. Ni får då själv ta reda på hur man gör motsvarande ändringar.

Om ni gör laborationen i en datorsal på campus är servern redan installerad och då behöver ni bara konfigurera och starta den. Annars installerar ni serverprogrammet genom att installera paketet *apache2* i Ubuntu.

4.1 Inledande konfiguration

Nästa steg blir att ändra Apaches konfigurationsfiler. Eftersom vi nu ska simulera en tillgänglighetsattack mot er webbserver vill vi köra denna experimentserver som er egen användare utan superuserrättigheter. Kopiera katalogen */etc/apache2* till din hemkatalog.

I din nykopierade katalog finns en fil *envvars* som du nu ska öppna för redigering. Ersätt raden `unset HOME` med

```
export APACHE_CONFDIR=/home/<user>/apache2
```

Där *<user>* är användarnamnet för den användare som är inloggad. Sätt även variablerna `APACHE_RUN_USER` och `APACHE_RUN_GROUP` till *<user>*. Ändra därefter alla variabler med katalognamn till värdet `${APACHE_CONFDIR}`, exempelvis

```
export APACHE_RUN_DIR=${APACHE_CONFDIR}
```

Alla filer ska också ligga i `${APACHE_CONFDIR}`, exempelvis

```
export APACHE_PID_FILE=${APACHE_CONFDIR}/apache2.pid
```

När du är klart med ovanstående öppnar du filen *ports.conf* för redigering. I ett UNIX-likt system är alla portar lägre än 1024 reserverade för användning av superuser, vår server kan därför inte längre använda den vanlig HTTP-porten med nummer 80. Ändra därför 80 till 8080 och 443 till 4433 i denna fil. Detsamma ska du göra med filen *sites-enabled/000-default*. I denna fil ska du också ändra */var/www* till */home/<user>/apache2*. Det ska noteras att detta är en ytterst osäker konfiguration där serverns konfigurationsfiler finns tillgängliga helt öppet för alla. Anledningen att vi gör detta är av ren lättja för att vårt syfte är något helt annat.

För att starta servern kör du följande kommandorad i terminalen:

```
1 $ . /home/<user>/apache2/envvars && \
2   apache2 -d /home/<user>/apache2 -f apache2.conf -k start
3 $
```

Notera att det inledande dollartecknet är prompten i terminalen. Om du nu öppnar en webbläsare och ansluter till

`http://localhost:8080`

ska du se en listning av alla konfigurationsfilerna. För att senare starta om servern byter du ut *start* i kommandoraden ovan mot *stop* för att stoppa den och sedan kör du igång den igen med *start*.

4.2 Inställningar för att simulera en tillgänglighetsattack

För att kunna simulera en tillgänglighetsattack behöver vi ändra följande parametrar i filen *apache2.conf*:

- `Timeout`,
- `KeepAlive`,
- `MaxKeepAliveRequests`, och
- `KeepAliveTimeout`.

Vi behöver också parametrarna

- `StartServers`,
- `MinSpareServers`,
- `MaxSpareServers`,

- `MaxClients`, och
- `MaxRequestsPerChild`.

som finns i avsnittet `<IfModule mpm_prefork_module>` i samma fil. Läs om dessa parametrar i Apaches dokumentation [3] så att ni förstår deras funktion.

För att ni ska slippa skapa ett hundratal anslutningar till er webbserver för att simulera en tillgänglighetsattack ska vi därför ändra dessa parametrar lite. Vi börjar med att ändra `StartServers`, `MinSpareServers`, `MaxSpareServers` och `MaxClients` till 1 samt sätter `MaxRequestsPerChild` till 5. Efter att detta är utfört sparar ni konfigurationsfilen och startar därefter igång Apache. Därefter ansluter ni till webbservern via telnet(1) och skickar *GET /* till servern, som svar borde du få HTML-kod:

```
(0):danbos@ID20809793:apache2$ telnet localhost 8080
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
GET /
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
<html>
 <head>
   <title>Index of /</title>
[...]
</body></html>
Connection closed by foreign host.
(1):danbos@ID20809793:apache2$
```

Testa nu att öppna flera telnetanslutningar till servern och öppna webbsidan i webbläsaren samtidigt.

Ni har nu utfört en väldigt enkel tillgänglighetsattack. Testa nu att ändra värdena på de olika parametrarna och gör om attacken, se hur de olika värdena påverkar serverns beteende.

5 Examination

1. Redogör kort för de nya begrepp och tekniker ni har fått förståelse för under laborationen. Redogör också kort för eventuella tekniska problem ni har haft under genomförandet, och hur ni har löst dem.
2. Utifrån inställningsparametrarna nämnda ovan, det vill säga
   - `Timeout`,
   - `KeepAlive`,
   - `MaxKeepAliveRequests`,
   - `KeepAliveTimeout`,
   - `StartServers`,
   - `MinSpareServers`,
   - `MaxSpareServers`,

- `MaxClients`, och
- `MaxRequestsPerChild`,

förklara hur man endast med hjälp av dessa värden kan försvåra en tillgänglighetsattack likt den vi utförde och varför detta kan fungera. Finns det några nackdelar med att ändra parametrarna?

3. Förklara skillnaden mellan en SDA- och en DSDA-attack.

4. Ge ett konkret exempel på en verklig tillgänglighetsattack och beskriv kortfattat hur den gått till. Ange korrekta referenser.

5. Hur kan en organisation skydda sig mot en tillgänglighetsattack, exempelvis med hjälp av en brandvägg?

Referenser

[1] Anderson, Ross J. *Security engineering : a guide to building dependable distributed systems*. Wiley, New York, 2001. ISBN 0-471-38922-6. URL `http://www.cl.cam.ac.uk/~rja14/book.html`.

[2] Anderson, Ross J. *Security engineering : a guide to building dependable distributed systems*. Wiley, Indianapolis, IN, 2 utgåvan, 2008. ISBN 978-0-470-06852-6 (hbk.).

[3] *Documentation: Apache HTTP Server*. The Apache Software Foundation, 2012. URL `https://httpd.apache.org/docs/`.