Grau d'Enginyeria Informàtica: Gestió i Administració de Xarxes

Pràctica 4: serveis de xarxa & seguretat

Treball previ: Contestar a aquestes preguntes (màxim 15 minuts).

- Quina comanda ens permet veure els límits del usuari de processos, file descriptors, memòria, etc?
- Si els arxius de log del sistema comencen a créixer indefinidament, com evitaríem que ens consumeixi tot l'espai del disc?
- Com podem evitar que un usuari concret pugui deixar d'executar un binari que es troba a /usr/bin?
- Què és un atac de DoS? És Apache susceptible d'un atac de DoS? Com el podríem evitar?
- Quines són les diferents taules de Iptables? Com s'accepta o denega un servei concret (per exemple FTP)? Quins són els paràmetres més comuns a Iptables?

Exercici 1

En aquesta pràctica realitzarem configuracions de seguretat al nostre node A des del qual accedirà un usuari que vol saturar el nostre sistema. Haurem de limitar les seves accions i evitar que afecti a la resta d'usuaris o al nostre sistema.

Sobre A es crearà un usuari anomenat “troll” amb password “troll”. Aquest usuari intentarà bloquejar el sistema o molestar i s’haurà de configurar el sistema per evitar-ho. La idea és que tot usuari pot arribar a fer caure el nostre sistema i s'han de limitar les seves accions. El que intentarà fer és el següent:

- Utilitzar tota la memòria del sistema.

- Crear un fork bomb i paralitzar la CPU.

- Obrirà tants file descriptors que no tindrem espai per més.

- En el seu directori /home/troll crearà molts arxius grans per a bloquejar l’espai del sistema de fitxers.

- Començarà a omplir els logs del sistema per a fer-los créixer indefinidament i bloquejar l’espai del sistema de fitxers.

a) Què hauríem de fer per a que l’usuari “troll” no pugui executar binaris ni scripts.

b) Què faríem si també vol omplir el directori /tmp?

c) En relació a la xarxa:

- Volem saber quines connexions hi han establertes al nostre sistema i quins ports s'estan utilitzant. Quina comanda s'hauria d'utilitzar?

- Es creu que hi ha un ordinador dintre de la xarxa local que té algunes connexions

sospitoses. Com s'ha d'explorar amb nmap aquesta IP?

- Un usuari mirarà de crear errors a Apache per a omplir els seus logs. Com es pot evitar que aquesta acció també saturi el sistema?

- Des de fora ha preparat un atac de denegació de servei (DoS) a Apache. Apache té eines per evitar ser susceptible a aquests atacs. Descriu quina lògica es segueix per evitar que deixin fora de servei el servidor.

- Com es pot deshabilitar tot tipus de connexions per a una IP concreta, i per al servei de SSH des de altre IP.

- Després de veure que s'estan realitzant atacs des de fora hem de restringir l'accés a la nostra màquina amb regles de iptables que realitzin les següents restriccions:

  - No contestar missatges de ICMP.

  - Bloquejar als usuaris interns la sortida a twitter.com facebook.com i youtube.com.

  - No es vol que hi hagin més de tres connexions simultànies per SSH

  - Evitar que algunes IPs concretes es connectin. Mirar rangs de ips que es donen a un país o conjunt de països i no permetre l'accés.

d) Reconfigurar les MV (o esborrar i crear de noves) de forma que C estigui connectada a fent servir la xarxa 20.20.20/21.0 i crear sobre B una DMZ on A serà el Firewall que controlarà l'accés. En totes les MV haurà de tenir Apache2 i l'Openssh-server. Les accions que haurà de controlar A són:

i. Des d'una 4ª MV (D) es podrà accedir a l'Apache instal·lat a B però res més (i també a cap servei d'A ni de C).
ii. Des de C i A es podrà connectar a B tant a l'Apache com a ssh.
iii. Des de B NO es podrà connectar ni a A ni a C a cap servei.

Exercici 2

Rebutjar un atac de DoS sobre Apache. Sobre màster (A) instal·lar (o baixar i compilar) el programa slowhttptest del repositorio de Debian.

https://packages.debian.org/stable/slowhttptest

Slowhttptest implementa un dels atacs més freqüents de baix ample de banda de la capa d'aplicació de denegació de servei, com ara slowloris, Slow HTTP POST, Slow Read (basat en TCP persist timer exploit) pel drenatge de les connexions concurrents, així com atacs d'Apache Range Header.

Això genera que el servidor quedi fora de servei i faci un ús molt important de la memòria i de la CPU del sistema. Aquest tipus d'atacs (Slowloris i slow HTTP POST DoS) es basen en el fet que el protocol HTTP, per disseny, requereix que les sol·licituds es rebin totalment abans de ser processades.

Si una sol·licitud HTTP no està completa, o si la velocitat de transferència és molt baixa, el

servidor manté els seus recursos ocupat esperant la resta de les dades. Si el servidor manté massa recursos ocupats, això crea una denegació de servei. En aquesta eina es realitza l'enviament de peticions HTTP parcials generant desconnexió (denegació de servei) del servidor HTTP destí.

El codi font de l'última versió es a https://github.com/shekyan/slowhttptest

Per compilar:

```
tar -xzvf slowhttptest-1.7.tar.gz
cd slowhttptest-1.7
./configure –prefix=/root/slow
make
sudo make install
```

Per fer un test sobre el localhost:

```
/root/slow/slowhttptest -g
```

Observar l'estat i la pàgina html que deixa com estadística.

Per fer un test sobre B:

```
/root/slow/slowhttptest -g -c 10000 -u http://20.20.20.x -l 1000
```

Analitzar els paràmetres, executar els tests, verificar la seva caiguda (amb la connexió al servidor, tcpdump i ss) i mitigar-los. En l'informe de les pràctiques s'haurà d'incloure la gràfica de l'abans i després de l'atac generada pel mateix slowhttptest amb l'opció -g.

Exercici 3

Per protegir el servidor Apache consideri l'opció d'instal·lar un Apache Proxy Balancer. Per això es pot fer servir la configuració de l'exercici 1.d eliminat el firewall i balancejar les peticions que arribin des de D (client) a A (proxy balancer) entre dos nodes (C i B). Verificar el funcionament posant diferents index.html en C i B.

Instal·lar slowhttptest sobre D, fer peticions a A i veure el resultat, analitzar les possibilitats i extraure conclusions si permet millorar los resultats abans obtingut quan aquest no es tenia el servidor proxy (exercici 2).

Exercici 4

Considerar que els nostres servidors de SSH estan en un entorn hostil i necessitem controlar automàticament les connexions als servidors.

a) Instal·lar el programari **medusa** del repositori de Debian sobre A, crear un usuari en B, baixar una llista de passwd per exemple "500 worst passwd" de:

https://wiki.skullsecurity.org/Passwords

i fer l'atac esbrinar si podem superar per força bruta el passwd a través de connexions ssh a B.

b) Instal·lar **fail2ban** i configurar-ho per a que afegeixi una nova regla amb iptables que faci un 'drop' de la connexió rebutjant l'acat amb medusa.

Grau d'Enginyeria Informàtica: Gestió i Administració de Xarxes

Generar un informe que inclogui les captura de pantalles més importants indicant com s'han resolt cadascun dels exercicis, indicar quins passos s'han seguit en la configuració, una descripció de cadascun i les verificacions realitzades de funcionament. Finalment afegir unes conclusions personals sobre la tasca desenvolupada explicant quines son les aportacions realitzades I inconvenients trobats.

L'informe s'haurà de lliurar al CV a les dates indicades (just abans de la següent sessió) i està sotmès a control de plagis.