

Tűzfal készítés mesterfokon: Zorp-GPL 1.

1. Telepítés

a. Bejelentkezés a tűzfalra

- i. ssh 172.16.12.202 -l adminla
- ii. sudo su -

b. APT repository beállítása

- i. vim /etc/apt/sources.list
- ii. deb
http://download.opensuse.org/repositories/home:/VPetya:/zorp/xUbuntu_14.04/ /
- iii. apt-get update

c. Zorp telepítése

- i. apt-get install zorp zorp-modules iptables-zorp-addons iptables-kzorp-addon python-openssl=0.12-1ubuntu2
- ii. cd /var/cache/apt/archives
- iii. wget http://people.balabit.hu/szilard/zorp-gpl/packages/linux-image-3.2.0-10-generic-zorp50_3.2.0-10.24_amd64.deb
- iv. dpkg -i linux-image-3.2.0-10-generic-zorp50_3.2.0-10.24_amd64.deb
- v. apt-get purge linux-image-3.13*
- vi. reboot
- vii. ssh 172.16.12.202 -l adminla
- viii. sudo su -

2. Minimális beállítások

- i. cd /etc/zorp

b. kZorp

- i. touch zones.py
- ii. chown root:zorp zones.py
- iii. vim zones.py
 1. from Zorp.Zone import Zone
- iv. service kzorpd restart

c. Zorp

- i. touch instances.conf
- ii. chown root:zorp instances.conf
- iii. vim instances.conf
- iv. touch policy.py
- v. default_instance --verbose 3 --policy /etc/zorp/policy.py -- --num-of-processes 1
- vi. vim policy.py
 1. from Zorp.Core import *
 2. def default_instance():
 3. pass
- vii. service zorp start
- viii. ps axu | grep zorp

3. Alapvető fogalmak bemutatása

a. Zone

i. Internet zóna hozzáadása

1. vim zones.py
 - a. Zone('internet', addrs=['0.0.0.0/0', '::0/0'])

ii. Internet zóna tesztelése

1. service kzorpd reload
2. kzorp-client --zones
3. kzorp-client --lookup 1.1.1.1

- a. Zone name='internet', admin_parent='None'

iii.intranet zónák hozzáadása

1. vim zones.py
 - a. Zone(name='client', addrs=['192.168.200.0/24',])
 - b. Zone(name='server', addrs=['192.168.201.0/24',])

iv.zóna hierarchia tesztelése

1. service kzorpd reload
2. kzorpd-client --zones
3. kzorpd-client --lookup 10.0.0.1
 - a. Zone name='internet', admin_parent='None'
4. kzorpd-client --lookup 192.168.202.1
 - a. Zone name='internet', admin_parent='None'
5. kzorpd-client --lookup 192.168.201.1
 - a. Zone name='server', admin_parent='None'
6. kzorpd-client --lookup 192.168.200.1
 - a. Zone name='client', admin_parent='None'

v.hostnév alapú zónák

1. vim zones.py
 - a. Zone('google', hostnames=['google-public-dns-a.google.com', 'imap.gmail.com'])

vi.hostnév alapú zónák tesztelése

1. service kzorpd reload
2. kzorpd-client --zones
3. kzorpd-client --lookup 8.8.8.8
 - a. Zone name='google', admin_parent='None'

b. PFService

i. iptables szabályok hozzáadása

1. vim /etc/network/iptables.up.rules
2. *mangle
- 3.
4. :DIVERT -
- 5.
6. :PREROUTING ACCEPT
7. # mark and accept connection already handled by Zorp <1>
8. -A PREROUTING -m socket --transparent -j MARK --set-mark 0x80000000/0x80000000
9. -A PREROUTING -m mark --mark 0x80000000/0x80000000 --jump ACCEPT
10. -A PREROUTING -j DIVERT
- 11.
12. :INPUT ACCEPT
13. -A INPUT -j DIVERT
- 14.
15. :FORWARD ACCEPT
16. -A FORWARD -j DIVERT
- 17.
18. :OUTPUT ACCEPT
- 19.
20. :POSTROUTING ACCEPT
21. -A POSTROUTING -j DIVERT
- 22.

```

23. # DIVERT
24. # chain to collect KZorp related rules <2>
25. # insert rules here to bypass KZorp <3>
26. -A DIVERT -p tcp --dport ssh -j ACCEPT
27.
28. # jump to KZorp and mark the packet <4>
29. -A DIVERT -j KZORP --tproxy-mark 0x80000000/0x80000000
30.
31. COMMIT
32.
33. *filter
34.
35. :INPUT ACCEPT
36. -A INPUT -p tcp --dport ssh -j ACCEPT
37. # accept earlier marked packet <5>
38. -A INPUT -m mark --mark 0x80000000/0x80000000 -j ACCEPT
39.
40. :FORWARD DROP
41. # accept connection relates to a packet filter service <6>
42. -A FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT
43. -A FORWARD -m conntrack ! --ctstate INVALID -m service --service-type
    forward -j ACCEPT
44.
45. :OUTPUT ACCEPT
46.
47. COMMIT
48. vim /etc/network/ip6tables.up.rules
49. iptables -t nat -F (a nat nincs a konfiguráción, mert IPv6 esetén nem értelmezett!)
50. iptables -t nat -X
51. iptables-restore /etc/network/iptables.up.rules
52. ip6tables-restore /etc/network/ip6tables.up.rules
53. iptables -L -t mangle
54. iptables -L -t filter
55. iptables -L -t nat

```

ii. default accept rule hozzáadása

```

1. vim policy.py
2.     PFService(name='PFService',
    router=TransparentRouter(forge_addr=TRUE))
3.     Rule(service='PFService')
4. service zorp reload

```

iii. default accept rule tesztelése

```

1. kzorp-client --services --dispatchers
2. w3m http://192.168.201.254
3. ftp 192.168.201.254
    a. ls

```

iv. default drop rule hozzáadása

```

1. vim policy.py
2.     DenyService(name='DenyService')
3.     Rule(service='DenyService')
4. service zorp reload

```

v. default drop rule tesztelése

1. kzorp-client --services --dispatchers
2. w3m <http://192.168.201.254>
 - a. Nem működik (Ctrl-c)
3. ftp 192.168.201.254
 - a. Nem működik (Ctrl-c)

vi. specifikus szabályok hozzáadása

1. vim policy.py
2. from zones import *
3. [Rule\(service='PFService', src_zone=\['client', \], dst_zone=\['server', \], dst_port=\[80, \]\)](#)
4. [Rule\(service='PFService', src_zone=\['client', \], dst_zone=\['server', \], dst_port=\[21, \]\)](#)

vii. specifikus szabályok tesztelése

1. service zorp reload
2. kzorp-client --services --dispatchers
3. kzorp-client --evaluate tcp 192.168.200.1 192.168.201.254 eth1 --src-port 9999 --dst-port 80
 - a. Client zone: client
 - b. Server zone: server
 - c. Service: PFService
 - d. Dispatcher: default_instance/dsp/dispatch:0
4. kzorp-client --evaluate tcp 192.168.200.1 192.168.201.254 eth1 --src-port 9999 --dst-port 21
 - a. Client zone: client
 - b. Server zone: server
 - c. Service: PFService
 - d. Dispatcher: default_instance/dsp/dispatch:0
5. kzorp-client --evaluate tcp 192.168.200.1 192.168.201.254 eth1 --src-port 9999 --dst-port 443
 - a. Client zone: client
 - b. Server zone: server
 - c. Service: not found
 - d. Dispatcher: not found

c. Service

i. ip rule/route szabályok hozzáadása

1. vim /etc/iproute2/rt_tables
 - a. 100 tproxy
2. vim /etc/rc.local
 - a. ip -4 rule add fwmark 0x80000000/0x80000000 lookup tproxy
 - b. ip -4 route add local default dev lo table tproxy
 - c. ip -6 rule add fwmark 0x80000000/0x80000000 lookup tproxy
 - d. ip -6 route add local default dev lo table tproxy
3. reboot

ii. http szabály hozzáadása (PlugProxy)

1. sudo su -
2. cd /etc/zorp
3. vim policy.py
 4. from Zorp.Plug import *
 5. Service(name='PlugService',
router=TransparentRouter(forge_addr=TRUE), proxy_class=PlugProxy)

6. Rule(service='PlugService', src_zone=['client'],
dst_zone=['server'], dst_port=[80,])

iii.http szabály tesztelése (PlugProxy)

1. service zorp reload
2. kzorp-client --services --dispatchers
 - a. Service name='PlugService', flags='logging', type='Service', session_count='0'
 - b. Dispatcher name='default_instance/dsp/dispatch:0'
 - c. num_rules='1'
 - d.
 - e. rule_id='1', service='PlugService'
 - f. dst_port=[(80, 80)]
 - g. src_zone=['client']
 - h. dst_zone=['server']
3. tail -f /var/log/syslog
4. w3m <http://192.168.201.254>

iv.http szabály hozzáadása (HttpProxy)

1. vim policy.py
 2. from Zorp.Http import *
 3. Service(name='HttpService',
router=TransparentRouter(forge_addr=TRUE), proxy_class=HttpProxy)
 4. Rule(service='HttpService', dst_port=[80,])

v.http szabály tesztelése (HttpProxy)

1. service zorp reload
2. tail -f /var/log/syslog
3. w3m <http://192.168.201.254>

vi.ftp szabály hozzáadása (FtpProxyRW)

1. vim policy.py
 2. from Zorp.Ftp import *
 3. Service(name='FtpService', proxy_class=FtpProxyRW)
 4. Rule(service='FtpService', dst_port=[21,])

vii.ftp szabály tesztelése (FtpProxyRW)

1. service zorp reload
2. tail -f /var/log/syslog
3. ftp 192.168.201.254
 - a. ls
 - b. put kliens.txt

viii.ftp szabály módosítása (FtpProxyRO)

1. vim policy.py
 2. Service(name='FtpService',
router=TransparentRouter(forge_addr=TRUE),
proxy_class=FtpProxyRO)

ix.ftp szabály tesztelése (FtpProxyRO)

1. service zorp reload
2. tail -f /var/log/syslog
3. ftp 192.168.201.254
 - a. ls
 - b. put kliens.txt

4. SSL kapcsolat terminálása

- a. https forgalom átengedése

i. https szabály felvétele

```
1. vim policy.py
2. from Zorp.Keybridge import X509KeyBridge
3. from Zorp.Pssl import *
4. class HttpsProxyKeybridge(HttpProxy):
5.     key_generator=X509KeyBridge(
6.         key_file="/etc/zorp/keybridge/key.pem",
7.         key_passphrase="passphrase",
8.         cache_directory="/var/lib/zorp/keybridge-cache",
9.         trusted_ca_files=(
10.            "/etc/zorp/keybridge/ZorpGPL_TrustedCA.cert.pem",
11.            "/etc/zorp/keybridge/ZorpGPL_TrustedCA.key.pem",
12.            "passphrase"
13.        ),
14.         untrusted_ca_files=(
15.            "/etc/zorp/keybridge/ZorpGPL_UnTrustedCA.cert.pem",
16.            "/etc/zorp/keybridge/ZorpGPL_UnTrustedCA.key.pem",
17.            "passphrase"
18.        )
19.     )
20.
21.     def config(self):
22.         HttpProxy.config(self)
23.         self.require_host_header=FALSE
24.         self.ssl.handshake_seq=SSL_HSO_SERVER_CLIENT
25.         self.ssl.key_generator = self.key_generator
26.         self.ssl.client_keypair_generate=TRUE
27.         self.ssl.client_connection_security=SSL_FORCE_SSL
28.         self.ssl.client_verify_type=SSL_VERIFY_OPTIONAL_UNTRUSTED
29.         self.ssl.server_connection_security=SSL_FORCE_SSL
30.         self.ssl.server_verify_type=SSL_VERIFY_REQUIRED_UNTRUSTED
31.         self.ssl.server_ca_directory="/etc/ssl/certs"
32.         self.ssl.server_trusted_certs_directory="/etc/zorp/certs"
33.
34.         Service(name="HttpsService",
35.             router=TransparentRouter(forge_addr=TRUE),
36.             proxy_class=HttpsProxyKeybridge)
37.         Rule(service='HttpsService', dst_port=[443, ])
38.
39. mkdir keybridge
40. chown zorp:root keybridge
41. cd keybridge
42. wget https://raw.githubusercontent.com/balabit/zorp-
43.     examples/master/keybridge/ZorpGPL_TrustedCA.cert.pem
44. wget https://raw.githubusercontent.com/balabit/zorp-
45.     examples/master/keybridge/ZorpGPL\_TrustedCA.key.pem
46. wget https://raw.githubusercontent.com/balabit/zorp-
47.     examples/master/keybridge/ZorpGPL\_UnTrustedCA.cert.pem
48. wget https://raw.githubusercontent.com/balabit/zorp-
49.     examples/master/keybridge/ZorpGPL\_UnTrustedCA.key.pem
50. wget https://raw.githubusercontent.com/balabit/zorp-
51.     examples/master/keybridge/key.pem
```

44. cd ..

ii. https szabály tesztelése

1. service zorp reload
2. tail -f /var/log/syslog
3. w3m <https://192.168.201.254>

b. ssh forgalom blokkolása

i. https szabály változtatlan

ii. http szabály tesztelése

1. openssl s_client -connect 192.168.201.254:443
2. ssh 192.168.201.254 -p 443