

# Σύγχρονες Εφαρμογές Ασφάλειας Δικτύων

## Άσκηση 2

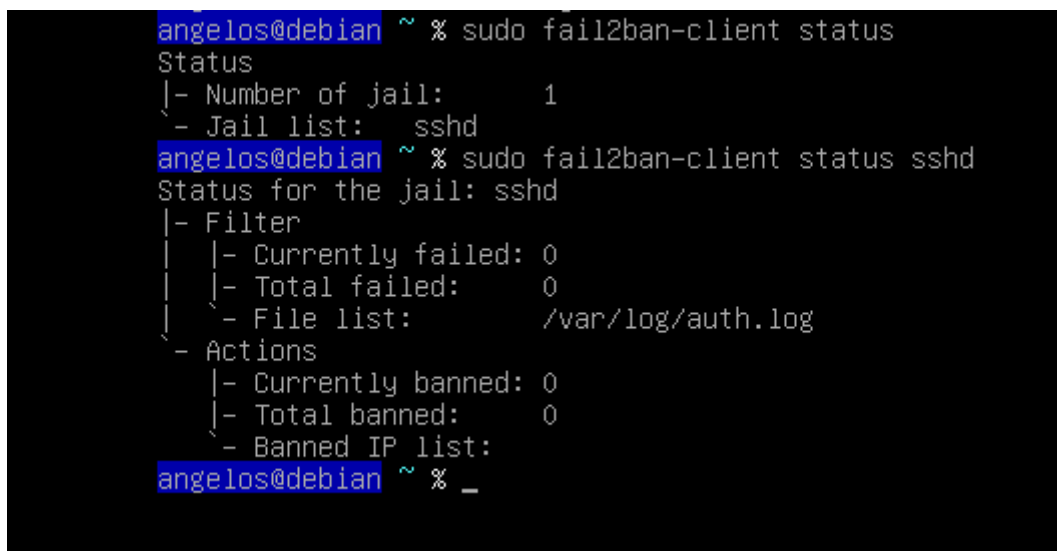
Αναγνωστόπουλος Άγγελος Νικόλαος  
5<sup>ο</sup> Έτος ΗΜΤΥ  
up1066593

## 2.1 Χρήση του πακέτου fail2ban

Σημείωση:

Τα αποτελέσματα των εντολών θα δίνονται ως επι το πλείστον με screenshots από τα virtual machines. Να σημειωθεί εδώ ότι υπάρχουν 2 VM, ένα debian headless (server όπου τρέχουμε τα πάντα) και ένα fedora (client) για testing. Αυτά δημιουργούνται με KVM και έχουν το δικό τους μικρό δίκτυο με NAT ορισμένο σε ένα .xml file. Αυτά για να μπορούν να “βλέπουν” το ένα το άλλο.

Αρχικά δημιουργούμε ένα αρχείο jail.local, το οποίο είναι αντίγραφο του jail.conf και επάνω στο οποίο θα εφαρμόσουμε το configuration για το fail2ban.



```
angelos@debian ~ % sudo fail2ban-client status
Status
|- Number of jail:      1
|- Jail list:   sshd
angelos@debian ~ % sudo fail2ban-client status sshd
Status for the jail: sshd
|- Filter
|   |- Currently failed: 0
|   |- Total failed:    0
|   - File list:        /var/log/auth.log
|- Actions
|   |- Currently banned: 0
|   |- Total banned:    0
|   - Banned IP list:
angelos@debian ~ % _
```

Figure 1: Χρήση εντολής status (μετά την δημιουργία αρχείου jail.local)

Έπειτα τροποποιούμε το αρχείο με τις κάτωθι εντολές:

```
# # "ignoreip" can be a list of IP addresses, CIDR masks or DNS hosts. Fail2ban
# # will not ban a host which matches an address in this list. Several addresses
# # can be defined using space (and/or comma) separator.
# #ignoreip = 127.0.0.1/8 ::1
#
# # External command that will take an tagged arguments to ignore, e.g. <ip>,
# # and return true if the IP is to be ignored. False otherwise.
# #
# # ignorecommand = /path/to/command <ip>
# ignorecommand =
#
# # "bantime" is the number of seconds that a host is banned.
bantime = 10m
#
# # A host is banned if it has generated "maxretry" during the last "findtime"
# # seconds.
findtime = 10m
#
# # "maxretry" is the number of failures before a host get banned.
maxretry = 5
#
# # "maxmatches" is the number of matches stored in ticket (resolvable via tag <matches> in act
# maxmatches = %(maxretry)s
#
# # "backend" specifies the backend used to get files modification.
# # Available options are "pyinotify", "gamin", "polling", "systemd" and "auto".
# # This option can be overridden in each jail's config file.
```

Figure 2: Εντολές jail file για sshd

Μετά τις 5 προσπάθειες σύνδεσης από τον fedora client, βλέπουμε ότι το terminal παύει να μας “πετάει” και απλώς παγώνει χωρίς να συνδεθεί πουθενά.

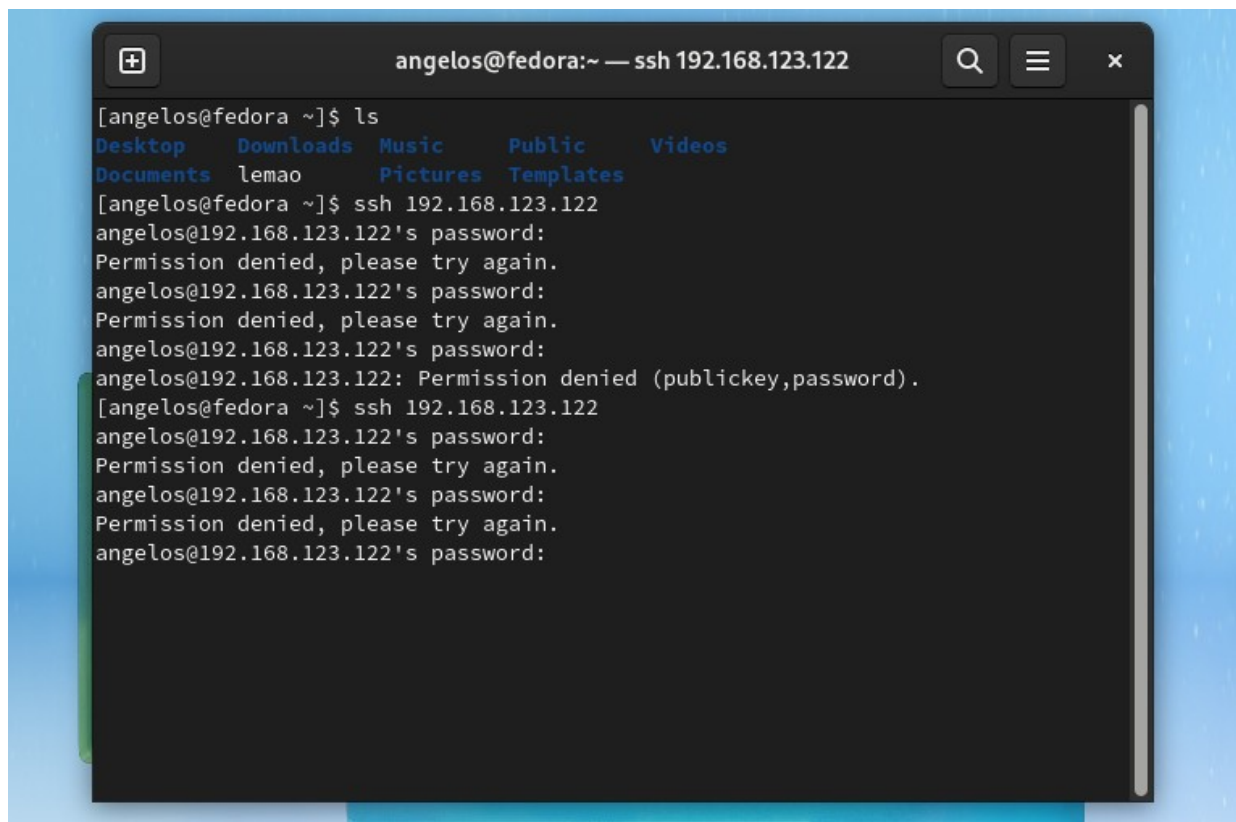


Figure 3: Πολλαπλές λανθασμένες απόπειρες σύνδεσης με ssh

Το αρχείο το οποίο μας ενδιαφέρει είναι το `/var/log/fail2ban.log`. Με την εντολή `cat` μπορούμε να δούμε τα περιεχόμενά του/.

```
2022-10-31 19:37:11,555 fail2ban.filter [640]: INFO Added logfile: /var/log/auth.log (pos = 11010, hash = 0007080947adca85243b98ad7337701ead47c61c)
2022-10-31 19:37:12,000 fail2ban.jail [640]: INFO Jail 'sshd' started
2022-10-31 19:37:39,137 fail2ban.filter [640]: INFO [sshd] Found 192.168.123.244 - 2022-10-31 19:37:33
2022-10-31 19:37:39,137 fail2ban.filter [640]: INFO [sshd] Found 192.168.123.244 - 2022-10-31 19:37:37
2022-10-31 19:37:41,191 fail2ban.filter [640]: INFO [sshd] Found 192.168.123.244 - 2022-10-31 19:37:41
2022-10-31 19:37:47,320 fail2ban.filter [640]: INFO [sshd] Found 192.168.123.244 - 2022-10-31 19:37:46
2022-10-31 19:37:50,440 fail2ban.filter [640]: INFO [sshd] Found 192.168.123.244 - 2022-10-31 19:37:50
2022-10-31 19:37:50,652 fail2ban.actions [640]: NOTICE [sshd] Ban 192.168.123.244
angelos@debian ~ %
```

Figure 4: Αποτέλεσμα `cat /var/log/fail2ban.log`

Όπως είναι λογικό, η IP του ενοχλητικού μας client έχει γίνει πλέον blacklist. Πράγμα που μπορούμε να δούμε και από το firewall μας.

```
angelos@debian ~ % sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source                destination            multiport dports ssh

Chain FORWARD (policy ACCEPT)
target     prot opt source                destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source                destination

Chain f2b-sshd (1 references)
target     prot opt source                destination            reject-with icmp-port-unreachable
REJECT    all  --  192.168.123.244        anywhere
RETURN    all  --  anywhere              anywhere
angelos@debian ~ %
```

Figure 5: Αποτέλεσμα `sudo iptables -L` μετά από πολλαπλές αποτυχίες σύνδεσης

Και εδώ βλέπουμε πως το IP μας έχει γίνει ban.

Ελέγχοντας ξανά τις εντολές για το status του fail2ban έχουμε πλέον το εξής output:

```
2022-10-31 19:37:50,652 fail2ban.actions [640]: M
angelos@debian ~ % sudo fail2ban-client status
Status
|- Number of jail:      1
- Jail list:  sshd
angelos@debian ~ % sudo fail2ban-client status sshd
Status for the jail: sshd
|- Filter
| |- Currently failed: 0
| |- Total failed:    5
| - File list:        /var/log/auth.log
- Actions
| |- Currently banned: 1
| |- Total banned:    1
- Banned IP list:     192.168.123.244
angelos@debian ~ %
```

Figure 6: Fail2ban status output μετά από πολλαπλές αποτυχίες σύνδεσης

Το fail2ban λειτουργεί όπως περιμέναμε, όμως τώρα κλειδωθήκαμε έξω απ' τον server μας! Για να αφαιρέσουμε μία διεύθυνση IP από την μαύρη λίστα, μπορούμε να χρησιμοποιήσουμε την εντολή unbanip του πακέτου fail2ban-client:

```
angelos@debian ~ % sudo fail2ban-client set sshd unbanip 192.168.123.244
1
angelos@debian ~ % sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source                destination            multiport dports ssh
f2b-sshd    tcp  --  anywhere              anywhere                multiport dports ssh

Chain FORWARD (policy ACCEPT)
target     prot opt source                destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source                destination

Chain f2b-sshd (1 references)
target     prot opt source                destination
RETURN     all  --  anywhere              anywhere
angelos@debian ~ %
```

Figure 7: Αφαίρεση της IP του fedora client από τις ανεπιθύμητες

Και για να μην ξανασυμβεί αυτό στο μέλλον, μπορούμε να δώσουμε στο fail2ban μία λίστα με διευθύνσεις οι οποίες να μην επηρεάζονται από τους γενικούς κανόνες που ορίζουμε. Να τις κάνουμε με άλλα λόγια whitelist. Αυτό γίνεται πανεύκολα από το αρχείο jail.local.

```
# # -----
# #
# # "ignoreself" specifies whether the local resp. own IP addresses should be ignored
# # (default is true). Fail2ban will not ban a host which matches such addresses.
# #ignoreself = true
# #
# # "ignoreip" can be a list of IP addresses, CIDR masks or DNS hosts. Fail2ban
# # will not ban a host which matches an address in this list. Several addresses
# # can be defined using space (and/or comma) separator.
ignoreip = 127.0.0.1/8 ::1 192.168.123.244/24
#
```

Figure 8: Whitelisted IPs για το fail2ban

Πλέον μπορούμε να δούμε ότι επιστρέψαμε στις 0 αποκλεισμένες διευθύνσεις:

```
angelos@debian ~ % sudo fail2ban-client status
Status
|- Number of jail:      1
- Jail list:  sshd
angelos@debian ~ % sudo fail2ban-client status sshd
Status for the jail: sshd
|- Filter
| |- Currently failed: 0
| |- Total failed:    0
| \- File list:       /var/log/auth.log
- Actions
  |- Currently banned: 0
  |- Total banned:    0
  \- Banned IP list:
angelos@debian ~ % _
```

Figure 9: Fail2ban Banned IPs μετά το whitelisting

## 2.2 Χρήση Public Key Authentication

Η αυθεντικοποίηση δημοσίου κλειδιού βασίζεται στο γεγονός ότι ένας server μπορεί να ελέγχει την ταυτότητα ενός client που προσπαθεί να συνδεθεί σε αυτόν, με βάση του private key του, δεδομένου ότι έχει πρόσβαση στο public key. Οι servers τυπικά έχουν “keychains” με πολλαπλά public keys στην κατοχή τους, από όλους αυτούς που έχουν δικαίωμα πρόσβασης. Το private key του client πρέπει να είναι καλά προστατευμένο, διότι σε αντίθετη περίπτωση κάποιος επιτηθέμενος θα είχε πρόσβαση σε όλους τους servers στους οποίους είχαμε και εμείς.

Γενικά θεωρείται ασφαλέστερη μέθοδος από αυτή των κωδικών μιας και δεν είναι τόσο εύκολο να μαντευτεί (πρακτικά αδύνατον), να βρεθεί με brute force, με dictionary ή κάποιο άλλο attack, στα οποία οι κωδικοί είναι ευάλωτοι. Παρόλα αυτά καλό είναι σε έναν ssh server να απενεργοποιούμε το root login και να συνδεόμαστε με χρήστη ειδικά διαμορφωμένο για τις administrative ανάγκες μας, καθώς και να εγκαθιστούμε λογισμικά όπως το fail2ban (βλ. παραπάνω) τα οποία προσθέτουν εξτρά τείχη ασφαλείας στο δίκτυό μας.

Για την αυθεντικοποίηση με κλειδιά, θα δημιουργήσουμε ένα ζεύγος κλειδιών (δημόσιο-ιδιωτικό) και θα στείλουμε το δημόσιο κλειδί μας στον ssh server. Είναι τόσο απλό. Έπειτα θα κλείσουμε το password authentication στον server μας και θα εμποδίσουμε το root login. Ουσιαστικά είναι 2 εντολές στον client και ένα file edit στον server.

Ξεκινάμε με την εντολή `ssh-keygen -b 4096` (προαιρετικά μέγεθος >2048 για ασφάλεια) και δημιουργούμε τα κλειδιά μας. Στη συνέχεια είτε με `scp` του κλειδιού στο directory `.ssh`, είτε με την εντολή `ssh-copy-id username@remote\_host` (προαιρετικά με το flag `-i` και όρισμα το path του δημοσίου κλειδιού. Αν δεν αλλάξαμε το default path τότε το ssh θα το βρεί μόνο του).

Μπορούμε πλέον να δοκιμάσουμε να συνδεθούμε στον server και θα δούμε ότι πλέον δεν μας ζητάει κωδικό! Το μόνο που μένει είναι να αλλάξουμε το αρχείο `sshd_config` και να κάνουμε restart την υπηρεσία.

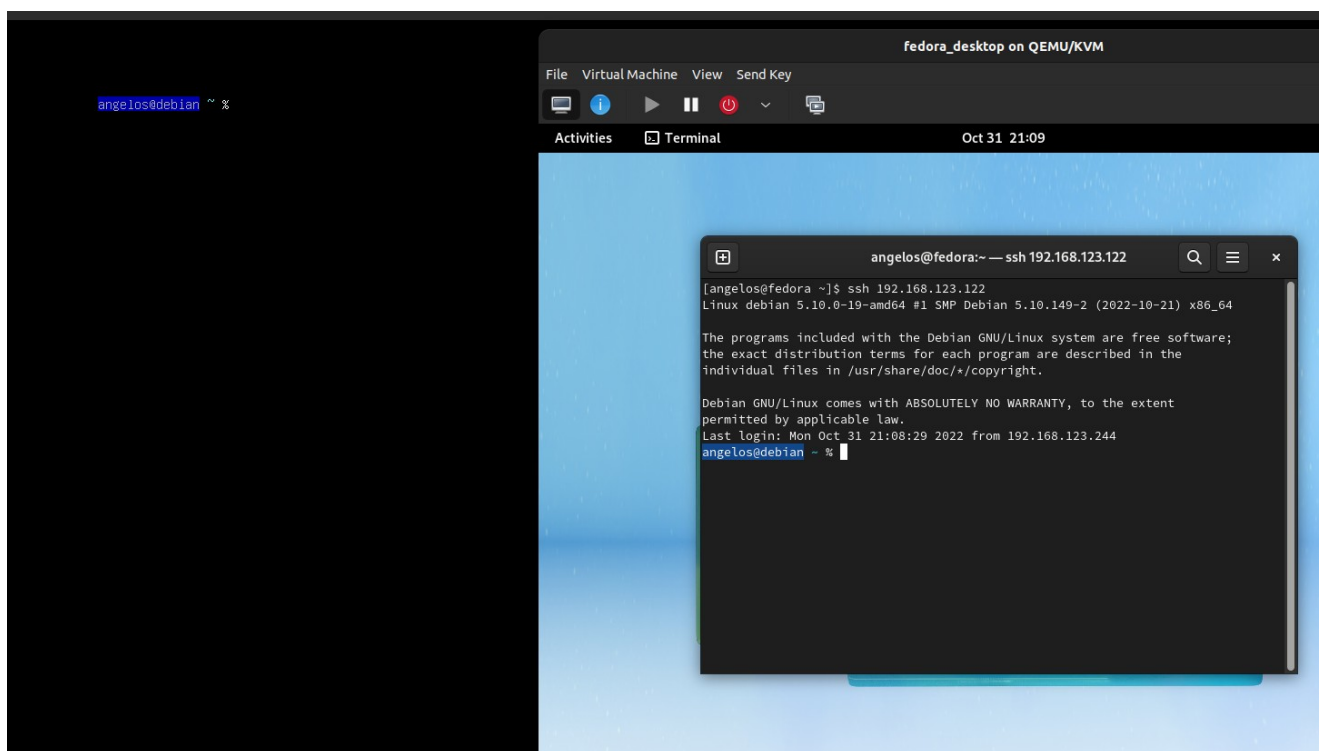


Figure 10: Σύνδεση με ssh αφού έχουμε μεταφέρει το δημ. κλειδί (δεν μας ζητήθηκε κωδικός για τη σύνδεση)

```
#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key

# Ciphers and keying
#RekeyLimit default none

# Logging
#SyslogFacility AUTH
LogLevel INFO

# Authentication:

#LoginGraceTime 2m
PermitRootLogin prohibit-password
#StrictModes yes
MaxAuthTries 5
MaxSessions 10

PubkeyAuthentication yes

# Expect .ssh/authorized_keys2 to be disregarded by default in future.
#AuthorizedKeysFile .ssh/authorized_keys .ssh/authorized_keys2

#AuthorizedPrincipalsFile none

#AuthorizedKeysCommand none
#AuthorizedKeysCommandUser nobody

# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#HostbasedAuthentication no
# Change to yes if you don't trust ~/.ssh/known_hosts for
# HostbasedAuthentication
#IgnoreUserKnownHosts no
# Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes

# To disable tunneled clear text passwords, change to no here!
PasswordAuthentication no
PermitEmptyPasswords no

# Change to yes to enable challenge-response passwords (beware issues with
# some PAM modules and threads)
ChallengeResponseAuthentication no
```

Figure 11: Τελικό configuration του sshd\_conf

## 2.3 Δημιουργία νέων φίλτρων για εφαρμογές με το fail2ban

Τα regular expressions που μας δίνουν την πληροφορία του ποιος συνδέθηκε (IP address) στην εφαρμογή είναι τα εξής:

```
([A-Z]+ )(\d\d\d\.\d\d\d\.\d\d\d\.\d\d\d)(\w+) (Joomla)
```

```
(\"[a-zA-Z]+\":)(\"\\d\\d\\d\\.\\d\\d\\d\\.\\d\\d\\d\\.\\d\\d\\d\")
```

Θα μπορούσαμε να γράξουμε κάτι πολύ πιο λεπτομερές που συμπεριλαμβάνει ώρα, agent κλπ. Το βασικότερο όμως είναι τα Ips οπότε και μέχρι εκεί πάω τα regex.

Για να τεστάρουμε τα παραπάνω, μπορούμε να δώσουμε σαν ορίσματα στην εντολή fail2ban-regex το αρχείο log που θέλουμε να εξετάσουμε, καθώς και τα match/ignore regex που θα εφαρμοστούν επάνω του. Πχ.

```
$fail2ban-regex /var/log/auth.log "([A-Z]+ )(\d\d\d\.\d\d\d\.\d\d\d\.\d\d\d)(\w+)"
```

Να σημειωθεί εδώ ότι τα regular expressions τα έγραφα σε python οπότε δεν είμαι 100% σίγουρος αν θα γίνουν match. Αντίστοιχο συντακτικό έχει και το UNIX αλλά ίσως υπάρχουν μικρές λεπτομέριες.

Στο κομμάτι miscelanious του jail.local μας προσθέτουμε τα δύο νέα φίλτρα με τα κατάλληλα parameters.

```
[joomla]
port      = 12345,12346
action_   = %(default/action_)s[name=%(__name__)s-tcp, protocol="tcp"]
           %(default/action_)s[name=%(__name__)s-udp, protocol="udp"]
findtime  = 10m
retries   = 5
logpath   = /var/log/joomla.log

[nextcloud]
port      = 22345,22346
action_   = %(default/action_)s[name=%(__name__)s-tcp, protocol="tcp"]
           %(default/action_)s[name=%(__name__)s-udp, protocol="udp"]
findtime  = 10m
retries   = 5
logpath   = /var/log/nextcloud.log
```

Figure 12: Τα φίλτρα μας στο jail.local