Ανάπτυξη Λογισμικού για Δίκτυα και Τηλεπικοινωνίες

Πρώτο Παραδοτέο

Χειμερινό εξάμηνο 2015-2016

Στα πλαίσια του μαθήματος καλείστε να υλοποιήσετε ένα κατανεμημένο σύστημα σάρωσης δικτύων υπολογιστών.

Το Nmap (Network Mapper)[1] είναι ένα λογισμικό ανοικτού κώδικα που έχει ως βασικό στόχο την ανίχνευση δικτυακών συσκευών/συστημάτων και τον έλεγχό τους με διάφορους τρόπους ως προς το λογισμικό που διαθέτουν, τις παρεχόμενος υπηρεσίες και τις ανοιχτές πόρτες στις οποίες μπορούν να συνδεθούν απομακρυσμένα νόμιμοι αλλά και κακόβουλοι χρήστες.

Όπως τα περισσότερα εργαλεία, το nmap χρησιμοποιείται τόσο από hackers που προσπαθούν να εισβάλουν στα υπολογιστικά συστήματα όσο και από τους διαχειριστές συστήματος (system administrators) προκειμένου να ανακαλύψουν αδυναμίες ασφάλειας που υπάρχουν στα συστήματα που διαχειρίζονται.

Για να προστατεύσουν τα δίκτυα υπολογιστών από τις εξωτερικές κακόβουλες σαρώσεις, οι διαχειριστές συστήματος τοποθετούν σε αυτά firewalls. Έτσι προκύπτει επιπλέον φόρτος για αυτούς, διότι θα πρέπει να συνδέονται ξεχωριστά σε κάθε δίκτυο που διαχειρίζονται για να εκτελέσουν σαρώσεις. Επομένως δημιουργείται η ανάγκη για την δημιουργία ενός λογισμικού που θα εκτελείται σε εσωτερικούς κόμβους δικτύων, το οποίο θα λαμβάνει αιτήματα για σαρώσεις και θα επιστρέφει τα αποτελέσματα αυτών στον διαχειριστή.

Πιο αναλυτικά:

Καλείστε να υλοποιήσετε ένα κατανεμημένο σύστημα, το οποίο θα αποτελείται από τρία επιμέρους υποσυστήματα:

1. Software Agent, SA (Πρώτο Παραδοτέο)
2. Aggregator Manager, AM (Δεύτερο Παραδοτέο)
3. Mobile Client, MC (Τρίτο Παραδοτέο)

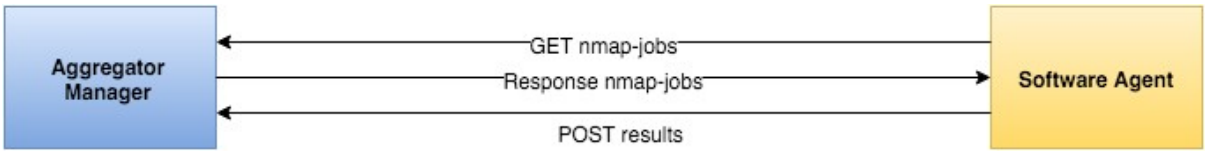
1 https://nmap.org/

Πρώτο Παραδοτέο

Ο Software Agent που καλείστε να υλοποιήσετε, θα είναι μια εφαρμογή σε γλώσσα προγραμματισμού Java.

Η εφαρμογή του SA θα στέλνει αιτήματα, για να ενημερωθεί για nmap-jobs που της έχουν ανατεθεί, στον AM. Αφού πάρει την πληροφορία για τα ποια jobs θα πρέπει να εκτελέσει, θα τα εκτελεί και θα στέλνει τα αποτελέσματα στον AM.

Στο παρακάτω σχήμα φαίνεται η επικοινωνία των AM – SA.



Στο πρώτο παραδοτέο καλείστε να διαβάζετε nmap-jobs από αρχείο[2]. Το αρχείο ενδεικτικά θα έχει την παρακάτω δομή.

```
1  Nmap-job id, Nmap-job parameters, flag periodic, time periodic nmap-job
2
3  1,-O -oX - 192.168.1.3,false,0
4
5  2,-A -oX - 192.168.1.0/24,true,10
6
7  3,-sO  -oX - localhost,true,5
8
```

Πιο συγκεκριμένα ένα nmap-job περιγράφεται από τα εξής πεδία:

1. Nmap-job id,
2. Nmap-job parameters,
3. flag periodic και
4. time periodic nmap-job (τιμή σε seconds)

Τα nmap-jobs χαρακτηρίζονται από ένα nmap-job id, τις παραμέτρους του nmap για το συγκεκριμένο nmap-job, ένα flag που υποδηλώνει αν το nmap-job είναι περιοδικό και ένα πεδίο για το χρόνο επανάληψης του συγκεκριμένου nmap-job. Στην περίπτωση που το flag periodic είναι false τότε το time periodic nmap-job δε λαμβάνεται υπόψιν.

Περιγραφή της βασικής λειτουργείας του SA.

Το κυρίως νήμα(thread) της εφαρμογής:

---

2 Η πλήρης υλοποίηση της επικοινωνίας μεταξύ του AM – SA, θα υλοποιηθεί στο δεύτερο παραδοτέο.

1. Εκκινεί έναν αριθμό από one_time_job-threads. (Ο αριθμός θα δίνετε από property file)
2. Εκκινεί ένα νήμα sender_thread που θα στέλνει περιοδικά τα αποτελέσματα στον AM.
3. Στέλνει περιοδικά αίτημα(request) στον AM[3].
4. Για κάθε nmap-job που θα λαμβάνει
    a. Εάν το αίτημα είναι περιοδικό, θα εκκινεί ένα νέο νήμα periodic_job-threads, που θα αναλάβει να το εξυπηρετεί.
    b. Αλλιώς θα εισάγει το συγκεκριμένο nmap-job σε μία διαμοιραζόμενη ουρά. Τα nmap-jobs που εισάγονται στη διαμοιραζόμενη ουρά θα εξυπηρετούνται από τα one_time_job-threads.
5. Εκτελεί τον ομαλό τερματισμό του προγράμματος SA. Με τη χρήση του control C (ctrl C) από το χρήστη, το πρόγραμμα πρέπει να αποδεσμεύει όλες τις δομές και να τερματίζει ομαλά όλα τα νήματα.

Περιγραφή της λειτουργίας thread pool του SA.

Όπως προαναφέρθηκε, όταν το nmap-job απευθύνεται σε μια μη περιοδική λειτουργία τότε θα εισάγεται σε μία διαμοιραζόμενη ουρά. Τα one_time_job-threads θα εξυπηρετούν τα nmap-job requests σύμφωνα με το γνωστό μοντέλο 1 παραγωγός N καταναλωτές (1 Producer – N Consumer).

Περιγραφή της λειτουργίας των threads του SA.

Τα threads της εφαρμογής (SA) είτε είναι one_time_job-threads είτε είναι periodic_job-threads, αφού αναλάβουν να εκτελέσουν κάποιο nmap-job, θα πρέπει να χρησιμοποιήσουν σωστά τις παραμέτρους και να εκτελέσουν τη εντολή nmap καθώς και να επιστρέψουν το αποτέλεσμα της εκτέλεσης σε XML. Αυτή η διαδικασία θα πρέπει να γίνει με τη χρήση της Κλάσης Runtime που διαθέτει η Java.[4]

Στην κλήση του nmap από τα threads θα πρέπει να φροντίζεται πως υπάρχει πάντα το flag ‘‘-oX -’’ έτσι ώστε το nmap να επιστρέψει το αποτέλεσμα του σε μορφή XML[5].

Στο παρακάτω σχήμα φαίνεται μια χαρακτηριστική κλήση του nmap.

3 Για τις ανάγκες του πρώτου παραδοτέου θα διαβάζετε περιοδικά έναν τυχαίο αριθμό για nmap-job requests από το αρχείο.

4 http://docs.oracle.com/javase/7/docs/api/java/lang/Runtime.html

5 https://nmap.org/book/output-formats-xml-output.html

```
# nmap -T4 -A -p 1-1000 -oX - scanme.nmap.org
<?xml version="1.0"?>
<?xml-stylesheet href="file:///usr/local/bin/../share/nmap/nmap.xsl" type="text/xsl"?>
<!-- Nmap 5.59BETA3 scan initiated Fri Sep  9 18:33:41 2011 as:
     nmap -T4 -A -p 1-1000 -oX - scanme.nmap.org -->
<nmaprun scanner="nmap" args="nmap -T4 -A -p 1-1000 -oX - scanme.nmap.org" start="1315618421"
         startstr="Fri Sep  9 18:33:41 2011" version="5.59BETA3" xmloutputversion="1.03">
 <scaninfo type="syn" protocol="tcp" numservices="1000" services="1-1000"/>
 <verbose level="0"/>
 <debugging level="0"/>
 <host starttime="1315618421" endtime="1315618434">
  <status state="up" reason="echo-reply"/>
  <address addr="74.207.244.221" addrtype="ipv4"/>
  <hostnames>
   <hostname name="scanme.nmap.org" type="user"/>
   <hostname name="li86-221.members.linode.com" type="PTR"/>
  </hostnames>
  <ports>
   <extraports state="closed" count="997">
    <extrareasons reason="resets" count="997"/>
   </extraports>
   <port protocol="tcp" portid="22">
    <state state="open" reason="syn-ack" reason_ttl="53"/>
    <service name="ssh" product="OpenSSH" version="5.3p1 Debian 3ubuntu7"
             extrainfo="protocol 2.0" ostype="Linux" method="probed" conf="10">
     <cpe>cpe:/a:openbsd:openssh:5.3p1</cpe>
     <cpe>cpe:/o:linux:kernel</cpe>
    </service>
    <script id="ssh-hostkey"
            output="1024 8d:60:f1:7c:ca:b7:3d:0a:d6:67:54:9d:69:d9:b9:dd (DSA)&#xa;
                    2048 79:f8:09:ac:d4:e2:32:42:10:49:d3:bd:20:82:85:ec (RSA)"/>
   </port>
   <port protocol="tcp" portid="80">
    <state state="open" reason="syn-ack" reason_ttl="53"/>
    <service name="http" product="Apache httpd" version="2.2.14"
             extrainfo="(Ubuntu)" method="probed" conf="10">
     <cpe>cpe:/a:apache:http_server:2.2.14</cpe>
    </service>
    <script id="http-title" output="Go ahead and ScanMe!"/>
   </port>
  </ports>
  <os>
   <portused state="open" proto="tcp" portid="22"/>
   <portused state="closed" proto="tcp" portid="1"/>
   <portused state="closed" proto="udp" portid="31289"/>
   <osclass type="general purpose" vendor="Linux" osfamily="Linux"
            osgen="2.6.X" accuracy="100">
    <cpe>cpe:/o:linux:linux_kernel:2.6.39</cpe>
   </osclass>
   <osmatch name="Linux 2.6.39" accuracy="100" line="39278"/>
  </os>
  <uptime seconds="23450" lastboot="Fri Sep  9 12:03:04 2011"/>
  <distance value="11"/>
  <tcpsequence index="199" difficulty="Good luck!"
               values="49018209,48C3EBED,495A2E7F,493EF30C,48ED43B3,495A9B0C"/>
  <ipidsequence class="All zeros" values="0,0,0,0,0,0"/>
  <tcptssequence class="1000HZ"
                 values="165CC09,165CC6E,165CCD2,165CD36,165CD9A,165CE48"/>
  <trace port="256" proto="tcp">
   <!-- Several hop elements removed for brevity -->
   <hop ttl="9" ipaddr="72.52.92.109" rtt="15.69" host="10gigabitethernet1-1.core1.fmt1.he.net"/>
   <hop ttl="10" ipaddr="64.62.250.6" rtt="12.06" host="linode-llc.10gigabitethernet2-3.core1.fmt1.he.net"/>
   <hop ttl="11" ipaddr="74.207.244.221" rtt="16.55" host="li86-221.members.linode.com"/>
  </trace>
  <times srtt="26517" rttvar="19989" to="106473"/>
 </host>
 <runstats>
  <finished time="1315618434" timestr="Fri Sep  9 18:33:54 2011" elapsed="13.66"
            summary="Nmap done at Fri Sep  9 18:33:54 2011; 1 IP address (1 host up)
                     scanned in 13.66 seconds" exit="success"/>
  <hosts up="1" down="0" total="1"/>
 </runstats>
</nmaprun>
```

Αφού το νήμα πάρει τα αποτελέσματα σε μορφή XML θα πρέπει να τα τοποθετήσει σε μια διαμοιραζόμενη δομή δεδομένων.

Το νήμα sender_thread, το οποίο θα εκτελείτε περιοδικά, θα στέλνει την πληροφορία της διαμοιραζόμενης δομής δεδομένων στον AM. Στην περίπτωση που η δομή έχει διαθέσιμα αποτελέσματα, το νήμα θα αναλαμβάνει να τα εκτυπώσει στο stdout. Αλλιώς θα αδρανοποιείται μέχρι την επόμενη εκτέλεση του. Στο δεύτερο παραδοτέο η αποστολή των αποτελεσμάτων θα γίνεται μέσω των Restful web services.

Απαιτήσεις συστήματος ανάπτυξης

1. GNU/Linux distribution (Λειτουργικό Σύστημα)
2. Java Oracle SE 7/8
3. MySQL 5.x έκδοση (Βάση Δεδομένων - Δεύτερο παραδοτέο)
4. Android API 4.1 ή νεότερο (Τρίτο παραδοτέο)
5. Eclipse Luna 4.4 (Εργαλείο ανάπτυξης της εφαρμογής.)

Κατά τη διαδικασία της ανάπτυξης της εργασίας είναι η υποχρεωτική η χρήση του εργαλείου Git (Version Control) και της πλατφόρμας gitlab που θα παραχωρηθεί για τις ανάγκες του μαθήματος.

Η υλοποίηση της εφαρμογής θα πρέπει:

i. Να υπακούει στις αρχές του αντικειμενοστρεφούς προγραμματισμού.
ii. Να είναι όσο το δυνατό παραμετροποιήσιμη και δυναμική γίνεται.
iii. Να γίνεται σωστή και αποδοτική διαχείριση της μνήμης.