პორტების სკანირების პროგრამის შემუშავება Python დაპროგრამების ენაზე

ლელა გაჩეჩილაძე, ნანა კურკუმული
საქართველოს ტექნიკური უნივერსიტეტი

რეზიუმე

წარმოდგენილია ქსელური პორტების სკანირების პროგრამა ობიექტზე ორიენტირებულ Python დაპროგრამების ენაზე. პროგრამულ კოდში გამოყენებულია socket და sys მოდულები, რომლებიც სოკეტების საშუალებით კომპიუტერულ ქსელთან ურთიერთობას უზრუნველყოფს. შემუშავებული კოდის გამოყენებით შეგვიძლია ვნახოთ, თუ რომელი პორტებია გახსნილი კონკრეტულ IP-მისამართზე ან დომენზე. ეს მნიშვნელოვანია უსაფრთხოების ანალიზის თვალსაზრისით, რათა გამოვავლინოთ გახსნილი პორტები და საჭიროების შემთხვევაში დავხუროთ ისინი. მსგავსი სკანერების გამოყენებით ჰაკერები იმ კომპიუტერების პორტების სკანირებას ახდენენ, რომელთა გატეხვაც სურთ.

საკვანძო სიტყვები: სოკეტი. პორტი. სკანირება. პროტოკოლი. IP-მისამართი.

1. შესავალი

დაპროგრამების ენა Python-ის ქსელურ მხარდაჭერას საფუძვლად **სოკეტის** (socket) კონცეფცია უდევს. ქსელის საბოლოო წერტილის იდენტიფიცირებას სოკეტი ახდენს, რომლის პარადიგმა გასული საუკუნის 80-ან წლებში 4.2 BSD Berkley UNIX ვერსიაში გამოჩნდა. სწორედ ამ მიზეზით გამოიყენება ტერმინი **ბერკლის სოკეტი** [1].

სოკეტები თანამედროვე ქსელების საფუძველია, რადგან სოკეტი ცალკეულ კომპიუტერს უფლებას აძლევს ერთდროულად გაუწიოს მომსახურება როგოც მრავალ კლიენტს, ასევე სხვადასხვა ტიპის არაერთ ინფორმაციას. ეს **პორტის** (port) გამოყენების ხარჯზე მიიღწევა, რომელიც განსაზღვრულ კომპიუტერზე ნუმირებულ სოკეტს წარმოადგენს. ამბობენ, რომ სერვერული პროცესი პორტს მანამდე „უსმენს“, სანამ კლიენტი არ დაუკავშირდება მას. სერვერს მრავალი კლიენტის მიღება შეუძლია, რომლებიც პორტის ერთსა და იმავე ნომერთანაა ჩართულები, თუმცა ყოველი სეანსი უნიკალურია.

ქსელში არსებულ ყოველ კომპიუტერს TCP/IP პროტოკოლის მიხედვით უნიკალური IP-მისამართი (Internet Protocol Address - ინტერნეტ ოქმის მისამართი) - ლოკალურ ქსელში ან ინტერნეტში ჩართული მოწყობილობის (როგორც წესი, კომპიუტერის) უნიკალური იდენტიფიკატორი) გააჩნია, რომელიც იდენტიფიცირებისა და კავშირის დასამყარებლად გამოიყენება. ის 32-ბიტიანი რიცხვია, რომელიც, როგორც წესი, ოთხი რიცხვის სახითაა ჩაწერილი და გამოყოფილია წერტილებით. თითოეული მათგანი 0-დან 255-მდე რიცხვით დიაპაზონში იცვლება. IP-მისამართი შეიძლება იყოს დროებითი და ყოველი დაკავშირების დროს დინამიკურად გამოიყოს ან იყოს მუდმივი. IP-მისამართები შიგა ქსელურ სისტემებში გამოიყენება [2].

კავშირის დამყარებისთანავე მაღალი დონის პროტოკოლს მივმართავთ, რომელიც გამოსაყენებელ პორტზეა დამოკიდებული. TCP/IP პროტოკოლი სპეციფიური პროტოკოლებისთვის პირველი 1024 პორტის რეზერვირებას ახდენს. მაგალითად, პორტი №21 FTP

პროტოკოლისთვისაა განკუთვნილი, №23 - Telnet პროტოკოლისთვის, №25 - ელექტრონული ფოსტისთვის, №80 – HTTP პროტოკოლისთვის, №119 – netnews-თვის და ა.შ. ყოველი პროტოკოლი კლიენტის პორტთან ურთიერთობის სახეს განსაზღვრავს.

2. ძირითადი ნაწილი

ნებისმიერი სერვერი თავის მომსახურებას ინტერნეტში დანომრილი პორტებით ხდის შესაძლებელს. მაგალითად, თუ სერვერ-მანქანაზე გაშვებულია ვებსერვერი და FTP (File Transfer Protocol — ქსელური ოქმი, რომლის დანიშნულებაა ერთი კომპიუტერიდან მეორეში ფაილების გადაგზავნა კომპიუტერული ქსელის საშუალებით) სერვერი, ვებ-სერვერი ჩვეულებრივ ხელმისაწვდომი უნდა იყოს მე-80 პორტზე, ხოლო FTP სერვერი ხელმისაწვდომი იქნება 21-ე პორტზე. კლიენტი მომსახურებას უკავშირდება სპეციფიკურ IP-მისამართზე, სპეციფიკურ პორტზე. თუ, ჩვენი კომპიუტერი ცნობილია, როგორც xxx.yyy.com, ნებისმიერ ინტერნეტში ჩართულ ადამიანს შეუძლია დაუკავშირდეს ჩვენს სერვერს მისამართით: http://xxx.yyy.com:918. “:918” პორტის ნომერს აღნიშნავს. როდესაც პორტი მითითებული არ არის, ბრაუზერი ასკვნის, რომ სერვერი კარგად ცნობილ მე-80 პორტს იყენებს [2].

ამრიგად, პორტების სკანირების პროგრამა, ვფიქრობთ საინტერესოა იმ თვალსაზრისით, რომ მისი საშუალებით შეგვიძლია ვნახოთ, თუ რომელი პორტებია გახსნილი კონკრეტულ IP მისამართზე ან დომენზე, რაც უსაფრთხოების ანალიზის თვალსაზრისით ძალზედ მნიშვნელოვანია.

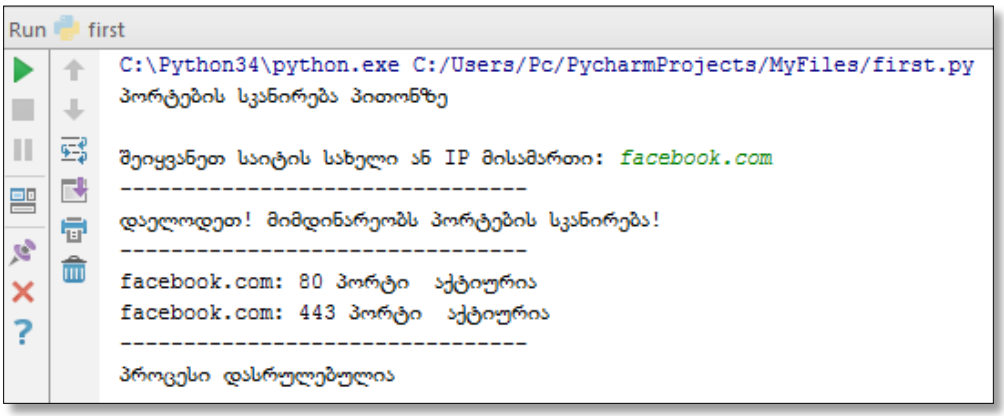
ქვემოთ წარმოდენილია Python ენაზე შემუშავებული პროგრამა (ლისტინგი_1), რომელიც ქსელური პორტების სკანირებას ახდენს.

```
#--- ლისტინგი_1 ---
import socket
import sys
ports = [20, 21, 22, 23, 25, 42, 43, 53, 67, 69, 80, 110, 115, 123,
137, 138, 139, 143, 161, 179, 443, 445, 514,
515, 993, 995, 1080, 1194, 1433, 1702, 1723, 3128, 3268, 3306, 3389,
5432, 5060, 5900, 5938,
8080, 10000, 20000]
print ("პორტების სკანირება პითონზე")
print (" ")
host = input('შეიყვანეთ საიტის სახელი ან IP მისამართი: ')
print ("--------------------------------")
print ("დაელოდეთ! მიმდინარეობს პორტების სკანირება!")
print ("--------------------------------")
for port in ports:
    s = socket.socket()
    s.settimeout(1)
    try:
        s.connect((host, port))
    except socket.error:
```

```
        pass
    else:
        s.close
print ( host + ': ' + str(port) +  ' პორტი ' + ' აქტიურია ')
print ("--------------------------------")
print ("პროცესი დასრულებულია")
```

პროგრამის დასაწყისში ადგილი აქვს socket და sys მოდულების ჩართვას. socket მოდული სოკეტების საშუალებით ქსელთან ურთიერთობას უზრუნველყოფს, ხოლო sys მოდული საერთო დანიშნულების სისტემური მოდულია [3]. შემდგომ პროგრამაში ვქმნით იმ პორტებისგან შემგარ სიას, რომლებიც სკანერების მიერ მოწმდება. კოდის ძირითადი ნაწილი ციკლური პროცესია, სადაც ვცდილობთ კავშირი დავამყაროთ ports სიაში არსებულ პორტებთან. თუ კავშირის დამყარება შეუძლებელია, pass ბრძნება აქტირუდება, რომელიც ცარიელია და, შესაბამისად, არაფერი სრულდება. იმ შემთხვევაში, თუ რომელიმე პორტი გახსნილი აღმოჩნდა, პროგრამა შესაბამის შეტყობინებას გვაძლევს. პროგრამის დასასრულს, მოცემული დომენის ან IP მისამართის ყველა პორტის შემოწმების შემდეგ, სკანერის მუშაობის დამთავრების შესახებ მიიღება შეტყობინება [4].

პროგრამის შესრულების შედეგს შემდეგი სახე აქვს:



3. დასკვნა

ამრიგად, მიღებული შედეგების მიხედვით შეგვიძლია დავასკვნათ, რომ ჩვენ მიერ დაპორგრამების Python ენაზე შემუშავებული პროგრამა წარმატებით ახორციელებს პორტების სკანირებას.

ლიტერატურა-References-Литература:

1. Marsic I. (2013). Computer Networks. Rutgers University, New Brunswick, New Jersey.
2. Tanenbaum Andrew S. (2003). Computer Networks. Prentice Hall PTR, Pages 891.
3. Lutz M. (2008). Learning Python. Third Edition. Published by O'Reilly Media, Inc. Printed in the United States of America.
4. გაჩეჩილაძე ლ. (2018). დაპროგრამების ენა Python. სტუ. გამომც. „ტექნიკური უნივერსიტეტი". თბ.

DEVELOP THE PORTS SCANNING PROGRAM ON THE PYTHON PROGRAMMING LANGUAGE

Gachechiladze Lela, Nana Kurkumuli
Georgian Technical University

Summary

The paper represented the network ports scanning program on Python object-oriented programming language. In the program code are used socket and sys modules which to connect with the computer network by sockets. Through this we can see which ports are opened on a particular IP address or domain. This is important in terms of safety analysis, in order to show open ports and to close them in case of tar. Using similar scanners, hackers are scanning the computers ports they want to break.

РАЗРАБОТКА ПРОГРАМЫ СКАНИРОВАНИЯ ПОРТОВ НА ЯЗЫКЕ ПРОГРАММИРОВАНЯ PYTHON

Гачечиладзе Л., Куркумули Н.
Грузинский Технический Университет

Резюме

Представлена программа сканирования сетевых портов на объектно-ориентированном языке Python. В програмном коде использованы модули socket и sys, которые обеспечивают связь к компьютерным сетям с помощью сокетов. Используя разработанный код можно посмотреть, какие порты открыты на определённом IP адресе или доменном имени. Это важно для анализа безопасности, чтобы выявить открытые порты, и при необходимости, закрыть их. С помощью похожих сканеров, хакеры сканируют порты компьютеров, которые хотят взломать.