تمرین شماره شش

درس شبکه های کامپیوتری

مدرس: دکتر هاشمی

***

۱- RFC 5321 را مطالعه کرده و بگویید که MTA کوتاه شده ی چیست؟ سپس ایمیل اسپم دریافت شده ی زیر را در نظر گرفته و کاربری را که این اسپم را ارسال کرده شناسایی کنید. (با این فرض که همه ی کاربران به جز ارسال کننده ی اسپم راستگو هستند.)

```
From - Fri Nov 07 13:41:30 2008
Return-Path: <tennis5@pp33head.com>
Received: from barmail.cs.umass.edu
(barmail.cs.umass.edu [128.119.240.3]) by cs.umass.edu
(8.13.1/8.12.6) for <hg@cs.umass.edu>; Fri, 7 Nov 2008
13:27:10 -0500


Received: from asusus-4b96 (localhost [127.0.0.1]) by
barmail.cs.umass.edu (Spam Firewall) for
<hg@cs.umass.edu>; Fri,  7 Nov 2008 13:27:07 -0500
(EST)
Received: from asusus-4b96 ([58.88.21.177]) by
barmail.cs.umass.edu for <hg@cs.umass.edu>; Fri,
07 Nov 2008 13:27:07 -0500 (EST)
Received: from [58.88.21.177] by
inbnd55.exchangeddd.com; Sat, 8 Nov 2008 01:27:07 +0700
From: "Jonny" <tennis5@pp33head.com>
To: <hg@cs.umass.edu>
Subject: How to secure your savings
```

۲- فرض کنید که از طریق پروتکل POP3 به ایمیل خود وصل شده اید.

الف. اگر POP mail client شما در مد download-and-delete پیکربندی شده باشد، مذاکره ی زیر را کامل کنید.

```
C: list
S: 1 498
S: 2 912
S: .
C: retr 1
S: blah blah ...
S: .........blah
S: .
?
?
```

ب. حال فرض کنید که در مد download-and-keep کار می کنید. مذاکره بالا را با فرض جدید کامل کنید.

پ. با این فرض که مد کاری ما download-and-keep باشد (از آن چه که در بند ب نوشته اید استفاده کنید.)، اگر پیغام های ۱ و ۲ را بازیابی کنیم، سپس از POP خارج شویم، و چند دقیقه بعد دوباره وارد POP شویم تا ایمیل های جدید را بازیابی نماییم، متن نشست (session transcript) دوم را بنویسید. (در این فاصله پیغام جدیدی به ما ارسال نشده است.)

***

۳- الف. پایگاه داده ی whois چیست؟

ب. از پایگاه های داده ی whois بر روی اینترنت استفاده کنید و نام دو سرور DNS را به دست آورید. مشخص کنید که از کدام پایگاه های whois استفاده کرده اید.

پ. از nslookup استفاده کرده و در خواست های DNS را به سرور محلی تان و دو سروری که در بند ب به دست آورده اید، بفرستید. در خواست های خود را برای نوع A، MX و NS فرستاده و نتایج را خلاصه وار بنویسید.

ت. با استفاده از nslookup بررسی کنید که آیا وب سایت دانشگاه صنعتی اصفهان چندین IP address دارد؟

ث. توضیح دهید که یک مهاجم چگونه می تواند از پایگاه های whois و نیز ابزار nslookup بهره ببرد تا اطلاعاتی درباره ی سازمان مورد حمله اش به دست آورد؟

ج. توضیح دهید که چرا پایگاه های داده ی whois باید در دسترس همگان باشد؟

***

۴- فرض کنید که دانشگاه شما یک DNS server محلی داشته باشد. شما یک کاربر عادی هستید (مسوول شبکه یا نظیر آن نیستید.) آیا برای شما ممکن است که بفهمید یک وب سایت مربوط به خارج از دانشگاه در چند ثانیه ی گذشته توسط یکی از کاربران داخل دانشگاه بازدید شده است؟

***

۵- آیا می توانید مرورگر کامپیوتر خود را به گونه ای تنظیم کنید تا چندین ارتباط موازی با یک وب سایت ایجاد نماید؟ مزایا و معایب داشتن تعداد زیادی ارتباط TCP همزمان چیست؟

۶- برنامه های TCPClient، UDPClient، TCPServer و UDPServer را که به زبان پایتون نوشته شده اند در نظر بگیرید. (پیوست یک) دو برنامه ی اول (Client) بر روی یک کامپیوتر و دو برنامه ی بعدی (Server) بر روی یک کامپیوتر دیگر اجرا می شود.

الف. اگر برنامه ی TCPClient قبل از برنامه ی TCPServer اجرا شود، چه اتفاقی رخ می دهد و چرا؟

ب. اگر برنامه ی UDPClient قبل از برنامه ی UDPServer اجرا شود، چه اتفاقی رخ می دهد و چرا؟

پ. اگر شماره پورت های متفاوتی را بر روی طرف کلاینت و سرور اجرا کنیم، چه رخ می دهد؟

ت. فرض کنید که در برنامه ی UDPClient بعد از ایجاد سوکت، خط زیر را اضافه کنیم. آیا نیاز هست که برنامه ی سمت سرور را تغییر دهیم؟ شماره پورت برای سوکت های UDPClient و UDPServer چند است؟

```
clientSocket.bind(('  ', 5432))
```

شاد و سربلند باشید

مازندرانی

پیوست یک

TCPClient

```
  5 import socket
  6 TCP_IP = '127.0.0.1'
  7 TCP_PORT = 5005
  8 BUFFER_SIZE = 1024
  9 MESSAGE = "Hello, World!"
 10
 11 s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
 12 s.connect((TCP_IP, TCP_PORT))
 13 s.send(MESSAGE)
 14 data = s.recv(BUFFER_SIZE)
 15 s.close()
 16
 17 print "received data:", data
```

TCPServer

```
  5 import socket
  6 TCP_IP = '127.0.0.1'
  7 TCP_PORT = 5005
  8 BUFFER_SIZE = 20  # Normally 1024, but we want fast response
  9
 10 s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
 11 s.bind((TCP_IP, TCP_PORT))
 12 s.listen(1)
 13
 14 conn, addr = s.accept()
 15 print 'Connection address:', addr
 16 while 1:
 17     data = conn.recv(BUFFER_SIZE)
 18     if not data: break
 19     print "received data:", data
 20     conn.send(data)  # echo
 21 conn.close()
```

برنامه های مربوط به UDP نیز به طرز مشابه نوشته می شوند. البته این برنامه ها را به فرم های مختلفی می توان نوشت که پیشنهاد می شود یک جستجوی اینترنتی در این باره انجام شود.