Report del progetto di Reti di Calcolatori: Bot-Net Binotto Team

Informazioni sul progetto

Il progetto in questione si pone l'obiettivo di approfondire una delle tematiche più importanti del corso di Reti di Calcolatori (la comunicazione tra applicativi tramite l'utilizzo delle Socket) e allo stesso tempo di analizzare con un approccio di tipo "hands-on" una particolare tipologia di malware (i Bot) al fine di capirne le potenzialità, il funzionamento, i rischi e le modalità di trasmissione.

Realizzazione del Bot

Il nostro Bot è stato sviluppato con una certa tipologia di funzionalità, tenendo in considerazione i tipici utilizzi ed il comportamento di questo genere di malware. Inoltre, al fine di rendere il Bot quanto più possibile "realistico", e dunque simulare il fatto che le vittime sono tipicamente ignare del fatto di essere state infettate, il suo comportamento è stato "sdoppiato", simulando l'azione di un Trojan. In particolare, a download completato, la vittima penserà di star utilizzando un tool di visualizzazione in percentuale dello stato di attività dei vari componenti hardware, oltre alla visualizzazione degli attuali processi in esecuzione. Il software è diviso in più file, tra cui Client.py e Server.py che rappresentano rispettivamente il malware che verrà eseguito sulla macchina vittima e il remote controller eseguito sul server centrale. La suddivisione è la seguente:

- **generalFiles**: funzioni generali (stampa, controlli, gestione dei segnali)
- **connectionFiles**: gestione della connessione
- **remoteCommandsFiles**: gestione dei comandi di controllo remoto

Requisiti e test effettuati

Il software utilizza le seguenti librerie:

- ***colorama==0.4.6***
- ***psutil==5.9.3***
- ***PyAutoGUI==0.9.53***
- ***tqdm==4.64.1***

Molti test sono stati effettuati sulla versione di Python 3.8 e su vari sistemi operativi tra cui: Windows, Mac OS e Linux.

Client e Server sono stati progettati in modo tale da essere in grado di ristabilire una connessione in caso di interruzioni, non avendo la possibilità, in uno scenario realistico, di poter riavviare il malware sulla macchina target.

Ogni comando è dotato di una sua specifica regular expression distinta per Windows/Unix che ci permette di definire precisamente come deve essere espresso il comando e permette la gestione degli errori di scrittura.

Alcuni esempi di controlli:

```
def regexcheck_download(comando):

    windows_regex_tip1 = r'^download "[\s\S]+\.[a-z]{1,4}" (\.(\\[^\/]+)+|\.{1,2}|(\\[^\/]+)+)'
    unix_regex_tip1 = '^download \"[\s\S]+\.[a-z]{1,4}\" (\.(\/[^\/]+)+|\.{1,2}|(\/[^\/]+)+)'
    windows_regex_tip2 = r'^download "[\s\S]+\.[a-z]{1,4}" nel percorso: C:(\\[^\/]+)+'
    unix_regex_tip2 = '^download \"[\s\S]+\.[a-z]{1,4}\" nel percorso: (\/[^\/]+)+'
    result1 = 'null'
    result2 = 'null'
    systemName = platform.system()

    if systemName == 'Windows':
        if re.match(windows_regex_tip1,comando):
            return "windowstip1"
        elif re.match(windows_regex_tip2,comando):
            return "windowstip2"
        else:
            return "not matched"
    else:
        if re.match(unix_regex_tip1,comando):
            return "unixtip1"
        elif re.match(unix_regex_tip2,comando):
            return "unixtip2"
        else:
            return "not matched"

# OK regExpr find
def regexcheck_find(comando):
    windows_regex=r'^find \.[a-z]{1,4} (\.(\\[a-zA-Z0-9, ,\_,\-]+)+|\.{1,2}|(\\[a-zA-Z0-9, ,\_,\-]+)+)'
    unix_regex='^find \.[a-z]{1,4} (\.(\/[a-zA-Z0-9, ,\_,\-]+)+|\.{1,2}|(\/[a-zA-Z0-9, ,\_,\-]+)+)'
    result = 'null'

    if platform.system()=='Windows':
        result = re.match(windows_regex,comando)
    else:
        result = re.match(unix_regex,comando)

    if result:
        return True
    else:
        return False
```

Funzionalità

Forniremo ora la lista dei comandi disponibili e la relativa spiegazione.

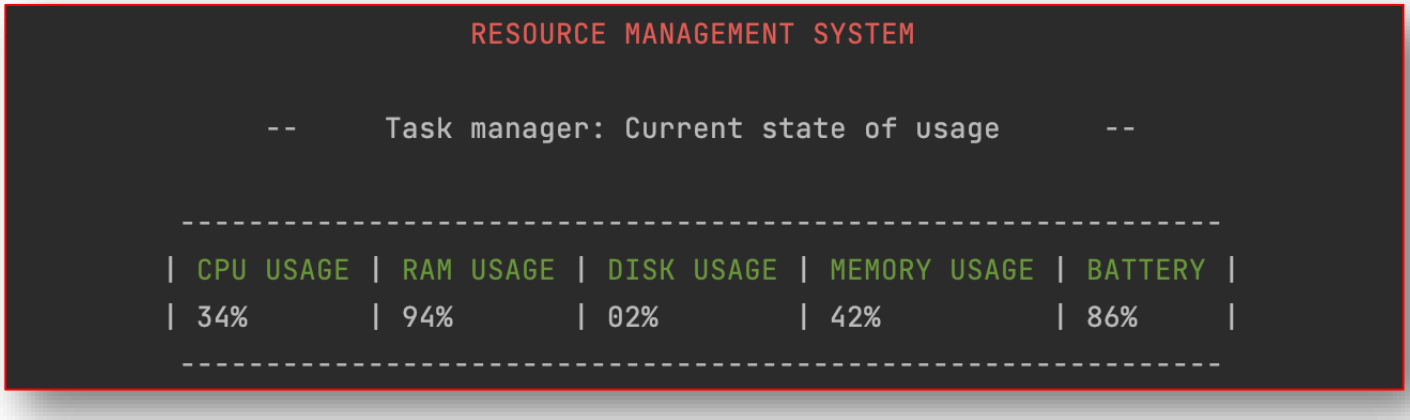
- ***Info**: stampa le informazioni sul sistema operativo della vittima*
- ***Clear**: pulisce lo schermo*
- ***Pwd**: stampa la working directory*
- ***Help**: stampa la lista dei comandi con relativa spiegazione*
- ***ls path**: lista tutti i file in un path*
- ***cd path**: cambia la posizione in base al path indicato*
- ***find .est path**: trova i file con una certa estensione in un certo percorso*
- ***filespath .est {1,n}**: trova in tutto il filesystem i file che corrispondono ad una o n estensioni e li scrive in un .txt con relativi percorsi*
- ***download "nomefile.estensione" path**: scarica il file con una certa estensione nel path desiderato*
- ***network**: corrisponde ad un ipconfig/ifconfig*
- ***screenshot**: effettua uno screen*
- ***open "nomezip"**: apre un file zip*
- ***exit**: chiusura controllo remoto*

Tipologie di Path disponibili:

- ***<.>*** *path corrente*
- ***<..>*** *path fino al percorso precedente alla wd*
- ***<./<Path> >*** *path dalla cartella corrente + path*
- ***<Path>*** *path relativo o assoluto*

Funzionamento

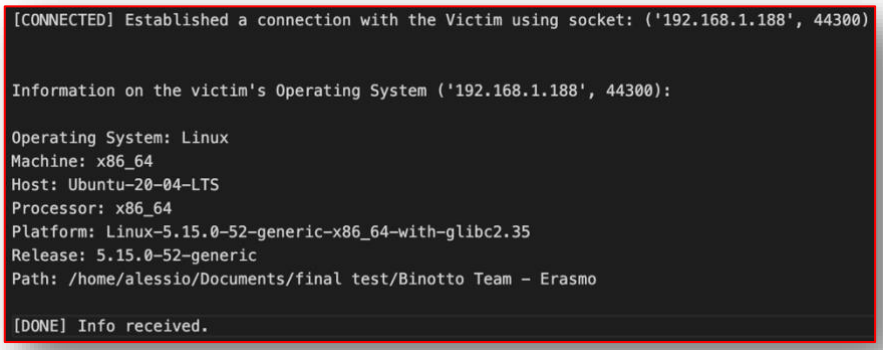
Ad avvio del Client, la vittima potrà visualizzare solo ed esclusivamente il comportamento da Trojan, rimanendo ignara di tutto quello che in maniera velata sta avvenendo. Il Server creerà una cartella apposita con la rispettiva Socket assegnata, dove verranno inseriti tutti gli eventuali file scaricati dalla macchina target. Subito dopo aver effettuato la connessione, qualsiasi comando e/o informazione di risposta dal client sarà registrata in un file Log accessibile al termine della procedura di remote control, sempre nella cartella inizialmente creata. Il malware inoltra al Server informazioni sulla macchina Target, prima di avviare la procedura di controllo remoto. A procedura attivata, il malintenzionato può eseguire le funzionalità descritte nel file "README.txt", controllabili anche digitando il comando "help".

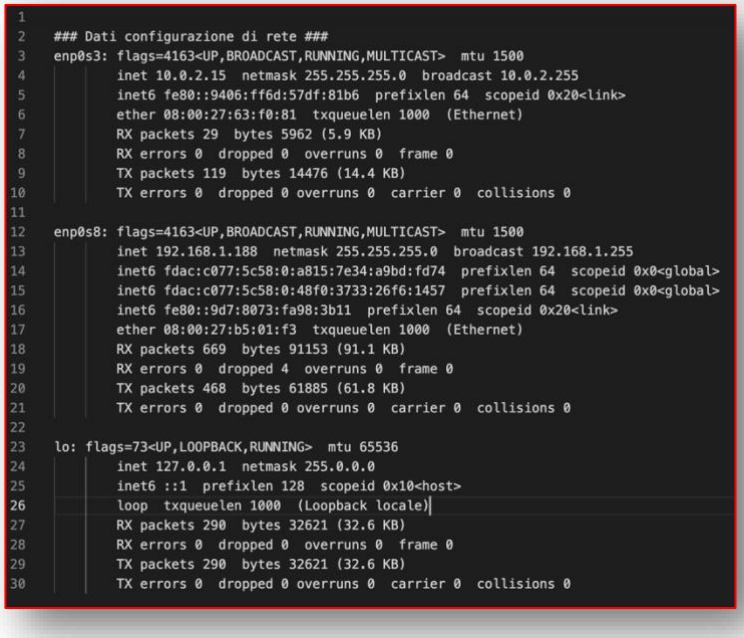


Dati raccolti

Descriviamo ora sinteticamente la tipologia di dati che sono stati raccolti nella prova finale del 13/12/2022.

- *Sistema Operativo e relativa versione, Architettura e Micro-Architettura del sistema, Path di esecuzione del malware*
- *Dati di configurazione di rete*
- *File .pdf, .txt, .png e relativi percorsi*
- *Contenuto di file .zip*

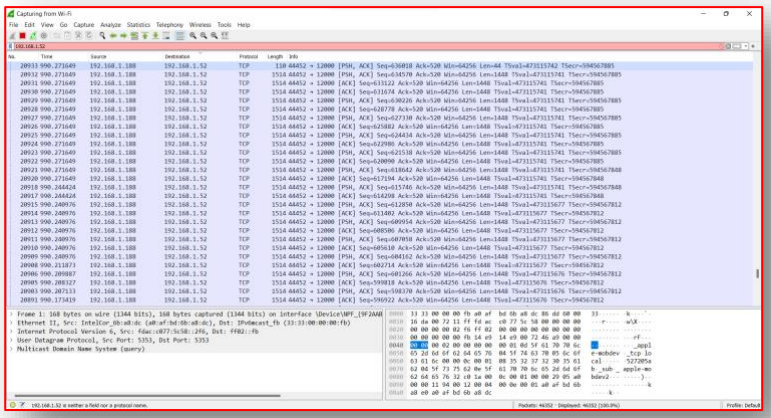




Nella cartella “WireShark” è disponibile il file .pcapng apribile tramite l’applicativo WireShark dove è possibile visualizzare lo scambio di pacchetti effettuato tra macchina Target e Server. Nelle due “cartelleClient” sono invece disponibili tutti i file scaricati e i relativi file Log. È inoltre disponibile un file creato tramite uno dei comandi sviluppati ad-hoc, il comando “filespath”, che permette la ricerca di qualsiasi tipologia di estensione all’interno di tutto il File System con indicazione del relativo percorso.

Di seguito alcuni screen di esempio:

confidential.pdf default.pdf form_russian.pdf topsecret.pdf secret.pdf fileLog.txt



```
1
2   /home/alessio/.myPwd
3   "SBAGLIATO! VEDI ALTROVE"
4
5   /home/alessio/.profile
6   "QUASI..."
7
8   /home/alessio/.profiles
9   "...
10  # WHAT? I PUT MY PASSWORDS HERE!
11  # AlessioPassword123
12  # ThisIsMyBestOne!
13  # isThisAJoke?
14  # sad@123@ertre@ERTY
15  ..."
```

```
1
2   /home/alessio/Documents/final test/Binotto Team - Erasmo$ ls /home/alessio
3   -: .config
4   -: Videos
5   -: .lesshst
6   -: .cache
7   -: .vboxclient-draganddrop.pid
8   -: Documents
9   -: .ssh
10  -: .vboxclient-vmsvga-session-tty2.pid
11  -: .local
12  -: .local_
13  -: Templates
14  -: .python_history
15  -: Desktop
16  -: Public
17  -: Music
18  -: .gnupg
19  -: .vboxclient-clipboard.pid
20  -: .vboxclient-seamless.pid
21  -: .bash_history
22  -: .profile
23  -: .myPwd
24  -: Downloads
25  -: snap
26  -: .bashrc
27  -: Pictures
28  -: .bash_logout
29  -: .pam_environment
```

```
Lista dei risultati per estensione: .pdf

"README.pdf" nel percorso: /home/alessio/Documents/final test/StealBot
"shared-mime-info-spec.pdf" nel percorso: /snap/snap-store/638/usr/share/doc/shared-mime-info
"shared-mime-info-spec.pdf" nel percorso: /snap/snap-store/599/usr/share/doc/shared-mime-info
"standard.pdf" nel percorso: /usr/share/cups/data
"secret.pdf" nel percorso: /usr/share/cups/data
"form_russian.pdf" nel percorso: /usr/share/cups/data
"classified.pdf" nel percorso: /usr/share/cups/data
"form_english.pdf" nel percorso: /usr/share/cups/data
"topsecret.pdf" nel percorso: /usr/share/cups/data
"default.pdf" nel percorso: /usr/share/cups/data
"unclassified.pdf" nel percorso: /usr/share/cups/data
"confidential.pdf" nel percorso: /usr/share/cups/data
"default-testpage.pdf" nel percorso: /usr/share/cups/data
"shared-mime-info-spec.pdf" nel percorso: /usr/share/doc/shared-mime-info
"manual.pdf" nel percorso: /usr/share/doc/printer-driver-foo2zjs
"xpdfimport_err.pdf" nel percorso: /usr/lib/libreoffice/share/xpdfimport
"README.pdf" nel percorso: /osx-shared/final test/StealBot

----Trovati 17 elementi.
### TROVATI TUTTI I FILE CON ESTENSIONE  .pdf###


Lista dei risultati per estensione: .docx


----Trovati 0 elementi.
### TROVATI TUTTI I FILE CON ESTENSIONE  .docx###


Lista dei risultati per estensione: .txt

"qfxblacklist.txt" nel percorso: /boot/grub
"last-crawl.txt" nel percorso: /home/alessio/.cache/tracker3/files
"first-index.txt" nel percorso: /home/alessio/.cache/tracker3/files
"readme.txt" nel percorso: /home/alessio/Documents/final test/Carnevale-Bosco-Granillo
"vendor.txt" nel percorso: /home/alessio/Documents/final test/StockFish-Bait/venv/Lib/site-packages/pip/_vendor
"top_level.txt" nel percorso: /home/alessio/Documents/final test/StockFish-Bait/venv/Lib/site-packages/psutil-5.9.4.dist-info
"entry_points.txt" nel percorso: /home/alessio/Documents/final test/StockFish-Bait/venv/Lib/site-packages/wheel-0.37.1.dist-info
"LICENSE.txt" nel percorso: /home/alessio/Documents/final test/StockFish-Bait/venv/Lib/site-packages/wheel-0.37.1.dist-info
"top_level.txt" nel percorso: /home/alessio/Documents/final test/StockFish-Bait/venv/Lib/site-packages/wheel-0.37.1.dist-info
"entry_points.txt" nel percorso: /home/alessio/Documents/final test/StockFish-Bait/venv/Lib/site-packages/pip-21.3.1.dist-info
"LICENSE.txt" nel percorso: /home/alessio/Documents/final test/StockFish-Bait/venv/Lib/site-packages/pip-21.3.1.dist-info
"top_level.txt" nel percorso: /home/alessio/Documents/final test/StockFish-Bait/venv/Lib/site-packages/pip-21.3.1.dist-info
"entry_points.txt" nel percorso: /home/alessio/Documents/final test/StockFish-Bait/venv/Lib/site-packages/setuptools-60.2.0.dist-info
"top_level.txt" nel percorso: /home/alessio/Documents/final test/StockFish-Bait/venv/Lib/site-packages/setuptools-60.2.0.dist-info
"Indicazioni.txt" nel percorso: /home/alessio/Documents/final test/ProgettoReti - IPythonieri - Francesco Bruno (TheMasterN1)
"README.txt" nel percorso: /home/alessio/Documents/final test/ThePhantomThieves
"requirements.txt" nel percorso: /home/alessio/Documents/final test/ThePhantomThieves
"README.txt" nel percorso: /home/alessio/Documents/final test/LonaCkers
"README.txt" nel percorso: /home/alessio/Documents/final test/TCPIPino
"requirements.txt" nel percorso: /home/alessio/Documents/final test/TCPIPino/bot
"README.txt" nel percorso: /home/alessio/Documents/final test/Bot TattiCi Nucleari - Giuseppe Francione
"requirements.txt" nel percorso: /home/alessio/Documents/final test/Cookie Thieves - macro21KGB
"README.txt" nel percorso: /home/alessio/Documents/final test/Gruppo Reti DGBS - Salvatore
"requirements.txt" nel percorso: /home/alessio/Documents/final test/Gruppo Reti DGBS - Salvatore
"README.txt" nel percorso: /home/alessio/Documents/final test/Binotto Team - Erasmo
"requirements.txt" nel percorso: /home/alessio/Documents/final test/Binotto Team - Erasmo
"README.txt" nel percorso: /home/alessio/Documents/final test/StealBot
"requirements.txt" nel percorso: /home/alessio/Documents/final test/StealBot
"Dipendenze & Istruzioni.txt" nel percorso: /home/alessio/Documents/final test/BeeBeeGee - Tenam
```