Laboratorul numărul 10
Dezvoltarea de programe pentru injecție și exploatare utilizând Python

Practic acestea sunt programe personalizate pentru un anumit context bine definit. Din nefericire dezvoltarea lor necesită cunoștințe din toate ramurile științei calculatoarelor ceea ce face ca mulți hackeri de nivel scăzut sau mediui să utilizeze când este posibil programe gata dezvoltate de alții. Această soluție are un grad scăzut de succes dar și un grad crescut de risc atât de detectare cât și de infectare a celui care îl folosește. Unii hackeri de nivel mediu merg pe analiza și modificarea programelor existente de acest tip. Evident că și această comoditate implică aceleași dezavantaje dar într-o măsură mai mică. După cum am discuta și la curs un profesionist real în domeniul securității folosește după situație o combinație de instrumente existente dar și multe aplicații dezvoltate de el. Acest laborator vă va da câteva exemple de dezvoltare rapidă pentru începători a unor aplicații de tip payload sau exploit dar inclusiv framework-ul prezentat are tot rol educațional adică fără nici o valoare în câmpul tactic.
Totuși acestea sunt un prim pas în înțelegerea primară a manierei de dezvoltare a unor astfel de instrumente pentru cazurile reale. Deși exemplele sunt pe Python datorită ușurinței în dezvoltare a acestui limbaj profesioniștii nu prea îl folosesc în acest scop.
Având în vedere că v-am cerut să fie referințe clare la vulnerabilități dacă dezvoltați voi ceva veți avea nevoie și de https://pypi.org/project/cvesearch/ deci pip3 install cvesearch sau în pycharm adăugați direct în mediul specific proiectului curent
În primul rând a nu se uita că dezvoltarea de payload/exploits malware/ransomware se realizează numai în interiorul unei mașini virtuale bine izolată și în nici într-un caz pe sistemul de operare de bază. Sau se mai poate merge și pe abordarea cu HDD extern pe care v-am explicat-o (dar să dezactivați automount-ul hdd-urilor suplimentare detectate.
Pentru fii windows-ului puteți testa pentru faza de post exploatare și instrumentul aflat la
https://securityonline.info/crackmapexec-v4-0-pentesting-networks/
Evident că pinguinii pot să se joace testând/modificând modulele lui ale căror coduri sunt în link-urile de la aceeași pagină
De unde găsim aplicații sau servere vulnerabile pentru a realiza testarea. În afara soluțiilor discutate la laborator gen metasploitable2/3 sau abordarea Red/Blue se mai pot folosi și următoarele surse:
https://www.exploit-db.com/google-hacking-database - *pentru site-uri*
https://www.exploit-db.com/ - *pentru aplicații*
Dacă vreți să vedeți cum arată preocupările un fost student interesat de pentesting poate ar trebui să vă uitați la pagina de blog al lui Orange - https://blog.orange.tw/.

Să testăm în Pycharm un simplu scanner de porturi

```
import socket
import sys
import argparse



class PortScan:
    @staticmethod
    def get_args():
```

```
            parser = argparse.ArgumentParser()
            parser.add_argument("address", help="addresă IP")
            parser.add_argument("-p", "--ports", nargs='*', help="Porturi pentru analiză " +
"Acceptă valor 1,2 sau din gama 1-5")

            return parser.parse_args()
```

```
      def _scan(self, ip, ports):
            try:
                  for port in ports:
                        sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
                        result = sock.connect_ex((ip, port))
                        if result == 0:
                              print("Port {}: Deschis".format(port))

            except KeyboardInterrupt:
                  sys.exit("Analiză teminată")

            except socket.error:
                  sys.exit("Nu pot accesa serverul")
```

```

            finally:
                  sock.close()

      def main(self):
            args = self.get_args()

            ports = list()
            if not args.ports:
                  ports = range(1, 1025)
            else:
                  for entry in args.ports:
```

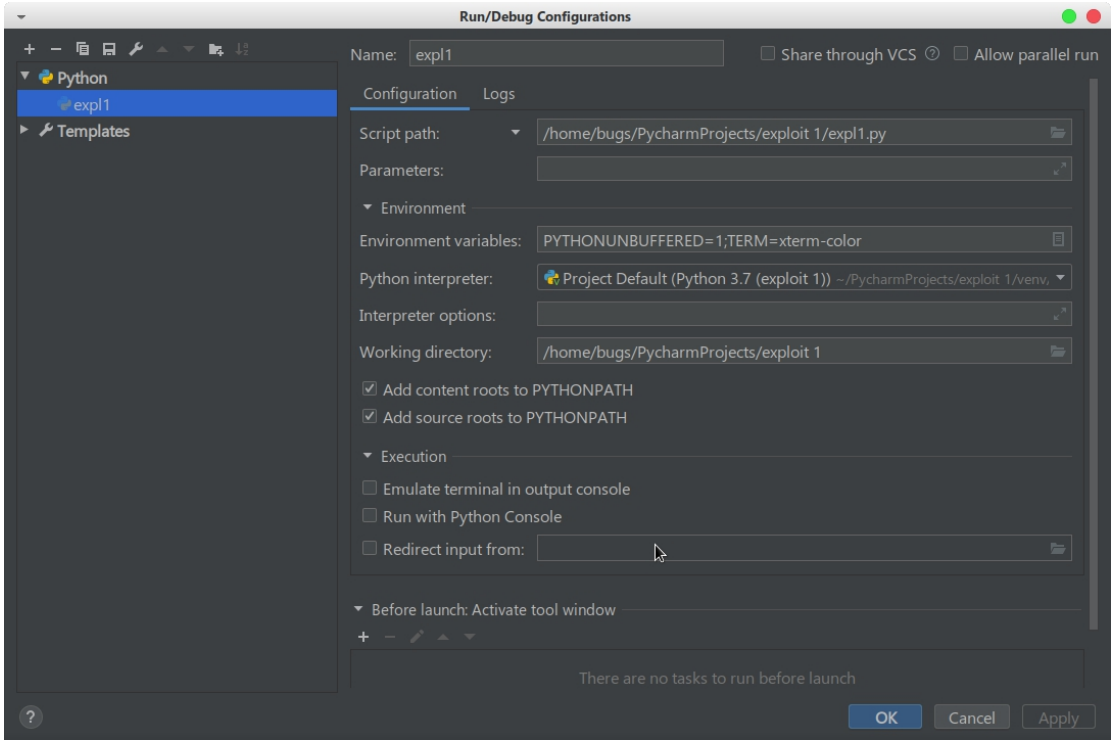
```
                        if '-' in entry:
                              values = [int(x) for x in entry.split('-')]
                              [ports.append(x) for x in range(values[0], values[1] + 1)]
                        else:
                              ports.append(int(entry))

            self._scan(args.address, ports)


if __name__ == '__main__':
      port_scan = PortScan()
      port_scan.main()
```

Pentru a testa programul de analiză porturi vă duceți la configurations și stabiliți lista de parametri cu ip-ul stației 127.0.0.1 -p1-100 (pentru a analiza primele 100 de porturi)

Exploit-uri mai găsim și la
https://www.exploitalert.com/search-results.html?search=Python

https://www.securitynewspaper.com/2016/11/12/exploiting-python-code-injection-web-applications/

Exemplu didactic de vierme
Pentru un mic viermișor duceți-vă la https://github.com/keniel123/Worm. Codul este self-explanatory. Totuși în cadrul testării programului o să vă loviți de unele mici probleme a căror rezolvare este detailată mai jos:
Pentru generarea altei chei secrete
openssl rand -base64 16
Cred ca este evident ca după înțelegerea codului trebuie să schimbați adresele de rețea corespunzător cu stațiile și să modificați activitățile worm-ului până la un nivel absolut inofensiv
Pentru eroarea că nu am setat variabila TERM la testarea în Pycharm a serverului worm adaugi variabila ca mai jos



Nu uitați pwgen pentru parole ceva mai sigure

Pwntools este un framework pentru script kiddies care permite dezvoltarea unor exploit-uri. Deci sunt bune numai cu rol educativ și în nici într-un caz pentru treburi mai serioase. În cazul exercițiului RedBlue puteți folosi aceste resurse deoarece la nivelul vostru de expertiză esențială este deprinderea tuturor abilităților necesare kill chain-ului și nu nivelul de complexitate al tehnicilor și aplicațiilor utilizate.

Lahttps://github.com/Gallopsled/pwntools/tree/dev/examples aveți o serie de exemple pentru care vă rog să creați proiecte separate utilizând pycharm (cum se adaugă la mediul curent o bibliotecă găsiți în laboratorul 14 de la PP)

Și tutorialul pentru pwntools

https://github.com/Gallopsled/pwntools-tutorial

După aceste exemple puteți trece și la cele din cartea Gray Hat Hacking The Ethical Hacker's Handbook.

Tot aici și

https://www.fccomposites.com.mx/roypnirue/wtuds/pwntools-remote

Mai puteți și să vă uitați la https://ocw.cs.pub.ro/courses/cns/labs/start

Temă pe acasă

Dacă sunteți din echipă red atunci ați descoperit o serie de vulnerabilități utilizând tool-urile educaționale prezentate până acum. Deci puteți fiecare să vă alegeți cea mai promițătoare și să încercați să creați un payload/exploit conform cu informațiile primare din baza de date CVE plus studiu suplimentar pe subiect de pe Internet.

Dacă sunteți din echipa blue și ați descoperit vulnerabilități pe stația unui atacator urmați aceeași procedură. Dacă nu încercați să dezvoltați utilizând framework-ul din laborator un payload.