Gabriela Czarska
Sieci Komputerowe
30 stycznia 2017

PROXY SSL

Oto dokumentacja zawierająca opis projektu Proxy. Folder zawiera dwa pliki napisane w pythonie3 *socks.py* - kilent oraz *sockd.py* - server. Klient łączy się z wybranym hostem nie bezpośednio, a za pośrednictwem mojego serwera proxy. Dodatkowo w folderze znajdują się pliki dotyczące certyfikatu wygenerowane poprzez self signed certificate, potrzebne do tego żeby połączenie między klientem a serwerem proxy było szyfrowane.

Uruchamianie:

./sockd.py

./socks.py

Klient - socks.py

Plik został skopiowany z Pańskiej strony, a funkcja create_connection została zmodyfikowana tak aby mój certyfikat został zatwierdzony. Tworze socket, następnie opakowuje go poprzez funkcje socket wrap (podając nazwę certyfikatu) , robię connect i teraz tak ja było wcześniej wysyłam zapytanie CONNECT.

Serwer Proxy - sockd.py

W tym pliku zmieniłam klasę MainServer zmieniając konstruktor oraz nadpisując get_request oraz zmieniłam funkcje handle.

W funkcji handle:

1. Czytamy pierwsza linię tzn czytam 2048, do parsera przekazuje to co przed \n a resztę będe wsadzać potem do pliku wf. Parser odczytuje hosta, port i username (tego z kim klient prosi nas o pośrednictwo).

2. Nawiązuje połączenie z tym kim poprosił mnie klient.

3. Tworze pliki rf i wf. Z rf będę czytać i przekazywać do klienta1, a do wf bede pisac to co napisze klient1.

4. Przepisuję do pliku wf resztę która pozostała mi z pierwszego odczytu

5. Odpalam funkcje reading która czyta i przepisuje aż dostanę 0 bytów. Używając select na rfile(plik do którego pisze nam klient1) i rf (utworzony plik do którego pisze ten z którym nawiązaliśmy polaczenie). To co klient1 napisze do rfile przepisuje do wf (pliku do pisania tego z którym nawiązaliśmy żądane połączenie, analogicznie z rf piszę do wfile. Gdy dostanę 0 bytów zamykam oba połączenia.