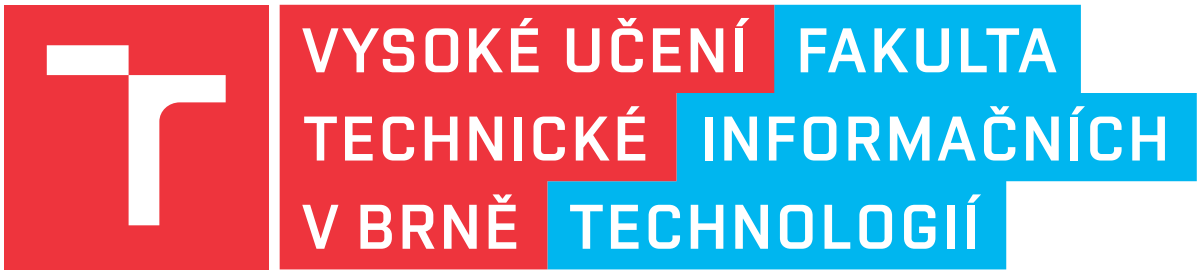

Dokumentácia projektu v predmete
Síťové aplikace a správa sítí

FTP/SSH Honeypot

5. novembra 2015

Dávid Mikuš (xmikus15@stud.fit.vutbr.cz)

Obsah

1 Úvod

Zadaním projektu je vytvoriť honeypot simulujúci FTP a SSH server cez protokoly IPv4 a IPv6.

1.1 Honeypot

Honeypot je mechanizmuz zabezbečenia počítača na detekovanie neoprávneneho použitia informačného systému. Informačný systém sa javí ako legitímny ale je izolovaný a monitorovaný. Slúži na nalákanie ľudí ktorí chcú preniknúť do systému iných ľudí kde nemajú oprávnenie.

V našom prípade bude honeypot len zachytávať autorizačne údaje a ukladať si ich. Dotyčného uživateľa len informuje o zlých údajoch a prípadne ukončí komunikáciu.

1.2 FTP

FTP (file transfer protocol) je TCP/IP protokol určený na prenos súborov medzi zariadeniami. Ako aj na internete tak aj na lokálnej sieti. Používa 2 porty, jeden slúžiaci na prenos dát a druhý na kontrolú dat a prenos FTP príkazov.

V tomto projekte si vystačíme len s jedným portom na prenos FTP príkazov pre prenos autorizačných údajov.

1.3 SSH

SSH (secure shell) je kryptografický sieťový TCP protokol ktorý umožnuje bezbečne vzdialené prihlásenie a využitie ďalších sieťových operácii cez nezabezbečenu sieť.

2 Návrh aplikácie

Celková aplikácia bude rozdelená na 3 častí:

1. Spracovanie argumentov
2. FTP Server
3. SSH Server

2.1 Spracovanie argumentov

Aplikácia musí pred spustenim FTP/SSH servera overiť, či uživateľ zadal všetky potrebné údaje, v spŕavnom formáte a dovolených kombinácii jednotlivých argumentov.

Povinné argumenty:

- mód -m - Uživateľ musí zadať prepínač `-m` a za ním parameter `FTP` alebo `SSH`, ktorý špecifikuje aký server bude spustený. Pri móde `SSH` je vyžadovaný ešte prepínač `-r`

- adresa -a - Za prepínačom `-a` musí nasledovať adresa na ktorej bude server počúvať
- port -p - Za prepínačom `-p` musí nasledovať port na ktorom bude server počúvať
- logovací súbor -l - Prepínač `-l` vyžaduje parameter - meno logovacieho suboru, do ktorého sa budú ukladať informácie zachytené v aplikácii
- RSA kĺúč -r - Prepínač ktorý je povinný len pre SSH mód, parameter za ním špecifikuje lokáciu súkromneho RSA kľúča.

Nepovinné argumenty:

- -c - Slúži na špecifikáciu maximálneho počtu paralelne pripojených klientov. Východza hodnota bez zadania je 10
- -t - v SSH móde slúži na špecifikáciu maximálneho počtu zadania hesla v jednom pripojení. Východzia hodnota je 3

2.2 FTP Server

FTP server sa spustí v nekonečnej slučke podla uživatelovych parametrov. Podporovaný je aj IPv4 aj IPv6 protokol. Po spusteni server začne počúvať na zadanej adrese a otvorí logovací súbor.

Pri príchode nového klienta skontroluje či aktuálny počet klientov není väčší ako je dovolený. Ak áno, ukončí ihned spojenie. Inak server pošlie klientovi odpoveď že je pripravený a začne príjimat príkazy od klienta. Server si zaznamená autorizačné údaje ktoré mu klient poslal. Keď príde údaj s heslom tak v tom momente si zaznamená čas a uloži zozbierané údaje do logovacieho súboru. Server odošle klientovi správu o zlom hesle a ukonči komunikáciu.

2.3 SSH Server

SSH server tak isto ako aj FTP beží v nekonečnej slučke, počúva na adrese a porte zadané uživateľom a skontroluje či aktuálny počet klientov není väčší ako je dovolený. Ak áno, ukončí ihned spojenie. Potom sa uskotoční `key exchange` počas ktorého sa nastaví symetrické šifrovanie slúžiace na zašifrovanie sedenia("session"). Po výmene kľúčov požiada server o službu. Server to príjme.

Server bude vždy vyžadovať autorizáciu heslom dokým mu nepríde. Server zaznamená do logovacieho súboru autorizačne údaje spolu s časom a odošle klientovi správu o zlých údajoch. Klient potom môže zasielať ďalšie autorizačne údaje, server bude stále odpovedať klientovi že zadal zle údaje. Toto sa bude opakovať dovtedy pokiaľ sa klient neodpojí alebo neprokočí maximalný počet pokusov, vtedy server ukončí spojenie.

3 Popis implementácie

Celá implementácia aplikácie je napísana v jazyku C++ s použitím štandardu C11. Aplikácia je rozdelená do viacero modulov:

1. main.cpp - Štart aplikácie, používa len ostatné moduly a odchytáva výnimky.

2. ArgParser.cpp - Trieda ktorá sa stará o spracovanie, skontrolovanie a uloženie argumentov.
3. FTPServer.cpp - Trieda ktorá imlementuje FTP server pomocou BSD socketov.
4. SSHServer.cpp - Trieda ktorá imeplementuje SSH server pomocou `libssh` knižnice.

3.1 Parsovanie argumentov

Parsovanie argumentov je implementované triedou `ArgParser` v module `ArgParser.cpp`. Jadro parsovanie je `getopt` ktorý zachytí prepínače a ich parametre. Samotná trieda sa už postará o ich uloženie a skontrolovanie správnosti parametrov a kombinácii prepínačov. Ak niečo nesedí vyvolá sa výnimka `ISAExc` ktorej sa predá správa o chybe a informuje uživateľa na `stderr`.

3.2 FTP Server

FTP server je implementovaný pomocou BSD socketov. Po pustení sa server pokusí vytvoriť socket a nabidnovat sa naň. Je vybraté prvé dostupné zariadenie vrátene funkciou `getaddrinfo` ktorému je predaná adresa a port od uživateľa. Táto funkcia sa sama vysporiada s IPv4 a IPv6. Následne začne na danom sockete počúvať.

Po príjmuti klienta funkciou `incConnections()` zvýši počet aktuálnych klientov a vráti `true` ak klienta môže obslúžiť. Každy klient bude pracovať vo svojom vlákne a pristupovať ku zdielanym premenným. Zabezbečenie synchronizácie zaisťuje `mutex` a `lock_guard` z `std`.

Každý klient obsahuje svoje vlastné `Data` do ktorého budu v priebehu uložene autorizačné údaje, čas obslúženia a adresa kleinta vyextrahovaná zo socketu pomocou `getnameinfo`.

Po príjmuti klienta server zašle klientovi správu `220` ktoré klientovi oznamuje že server je pripravený na nového klienta. Následne server čaká na odpoveď. Pomocou `recv()` ukladá data od klienta do pomocného bufferu ktorý sa v prípade nutnosti zväčší. Keď sa nájde sekvencia \r\n prestane server príjimať data.

Následne sa príkaz spracuje, ak sa tám nachádza `USER` tak sa uloží zadaný login a zmení sa stav klienta, kde server pošle odpoveď `331`, ktorě znamená že server chce ešte heslo. Ak sa v príkaze nájde `PASS`, uloží sa heslo a aj aktuálny čas. Klientovi sa zašle správa `530` ktorá znači ze údaje boli zlé.

Nakoniec sa všetky obdržané údaje uložia do logovacieho súboru ktorý je zabezbečeny mutexom a klienta odpojí.

3.3 SSH Server

SSH server je implementovaný pomocou `libssh` knižnice. Pri konštrukcii servera sa vytvorý nový ssh bind funkciou `ssh_bind_new()` a nastavia parametre servera pomocou `ssh_bind_options_set`. Otvorí sa logovací súbor a server začne načúvať.

Tak isto ako aj v FTP serveri, server po príchod klienta skontroluje či kapacita obsluhy nie je plná. Z `ssh_session` sa získa socket pomocou `ssh_get_fd` a výsledok sa predá do `getpeername` ktorá naplní `struct sockaddr` pomocou ktorej `getnameinfo` získa adresu klienta a uloží.

Následne nastane výmena kľúčov funkciou `ssh_handle_key_exchange`. Ak prebehne úspešne začne server dostávať spravy pomocou `ssh_message_get`. Server obdrži od klienta správu `SSH_MSG_SERVICE_REQUEST` a odpovie na nu `SSH_MSG_SERVICE_ACCEPT` toto zabezbeči funkcia `ssh_message_service_reply_success`

Funkcie `ssh_message_type` a `ssh_message_subtype` zistia o akú správu sa jedná. Ak sa jedná o typ `SSH_REQUEST_AUTH` a nejedná sa o podtyp `SSH_AUTH_METHOD_PASSWORD` teda autorizáciu heslom, pošle klientovi správú že chce autorizáciu heslom.
Nastaví to funkcia `ssh_message_auth_set_methods` s parametrom `SSH_AUTH_METHOD_PASSWORD`.

Ked obdrží autorizačné údaje spolu s heslom, uloží ich a zapíše do logovacieho súboru aj spolu s časom. Zvýši počet pokusov o zadanie hesla. Ak prekročí zadaný maximalný počet tak odpojí klienta, v opačnom prípade zašle klientovi žiadosť o heslo.

4 Základne informácie o programe

Program spustí jednoduchý honeypot erver, buď FTP alebo SSH, ktorý loguje zhromaždené údaje, najmä autorizačné do logovacieho súboru.

5 Návod na použitie

Synopsis: ./fakesrv -m mode -a addr -p port -l logfile [-r rsakey] [-c maxclients] [-t maxattempts]
Viz. Spracovanie argumentov 2.1

Literatúra

[1] https://en.wikipedia.org/wiki/Honeypot_(computing)

[2] https://sk.wikipedia.org/wiki/Protokol_prenosu_súborov

[3] https://en.wikipedia.org/wiki/Secure_Shell

[4] http://long.ccaba.upc.es/long/045Guidelines/eva/ipv6.html

[5] https://www.ietf.org/rfc/rfc959.txt

[6] https://www.ietf.org/rfc/rfc4253.txt

[7] http://api.libssh.org/master/group__libssh.html

[8] https://github.com/substack/libssh/blob/master/examples/samplesshd.c