UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Peter Kodermac

Slabosti preprostih uporabniških gesel

DIPLOMSKO DELO
NA VISOKOŠOLSKEM STROKOVNEM ŠTUDIJU

Mentor: prof. dr. Saša Divjak

Ljubljana, 2011

Univerza
v Ljubljani

Fakulteta za računalništvo
in informatiko

Tržaška 25
1000 Ljubljana, Slovenija
telefon: 01 476 84 11
faks: 01 426 46 47
www.fri.uni-lj.si
e-mail: dekanat@fri.uni-lj.si

Št. naloge: 00011/2010
Datum: 01.10.2010

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **PETER KODERMAC**

Naslov: **SLABOSTI PREPROSTIH UPORABNIŠKIH GESEL**
WEAKNESSES OF TOO SIMPLIFIED USER PASSWORDS

Vrsta naloge: Diplomsko delo visokošolskega strokovnega študija prve stopnje

Tematika naloge:

Analizirajte pomen gesel in nevarnosti, ki prežijo na uporabnike s preprostimi gesli. Proučite težave, ki se lahko pojavijo pri zlorabi osebnih podatkov in gesel. Podajte tipične načine oziroma mehanizme odkrivanja uporabniških gesel. Kot konkreten poligon uporabite možne napade na forum Fri- Info. Predstavite aplikacijo, ki je zmožna izvajati slovarski napad na tak forum.

Mentor:

prof. dr. Saša Divjak

Fakulteta za računalništvo in informatiko • UNIVERZA V LJUBLJANI • 2

Dekan:

prof. dr. Nikolaj Zimic

Besedilo je oblikovano z urejevalnikom besedil LaTeX.

Namesto te strani **vstavite** original izdane teme diplomskega dela s podpisom mentorja in dekana ter žigom fakultete, ki ga diplomant dvigne v študentskem referatu, preden odda izdelek v vezavo!

IZJAVA O AVTORSTVU

diplomskega dela

Spodaj podpisani/-a Peter Kodermac,

z vpisno številko 63040229,

sem avtor/-ica diplomskega dela z naslovom:

Slabosti preprostih uporabniških gesel

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal/-a samostojno pod mentorstvom
  prof. dr. Saše Divjak
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki "Dela FRI".

V Ljubljani, dne 15.04.2011 Podpis avtorja/-ice:

Zahvala

Rad bi se zahvalil mentorju prof. dr. Saši Divjak za strokovno pomoč in smernice pri pisanju diplomske naloge.

Hvaležen sem tudi za pomoč administratorju foruma Fri-Info, Ahacu Sedušaku, ki mi je omogočil analizo gesel na forumu.

Največja zahvala pa gre vsem mojim najbližjim.

Kazalo

Seznam uporabljenih kratic in simbolov

OCR	Optical character recognition
CAPTCHA	Completely Automated Public Turing test to tell Computers and Humans Apart
API	Application Programming Interface
PHP	Personal Home Page Tools
phpBB	PHP Bulletin Board
UNIX	UNiplexed Information and Computing System
HTML	HyperText Markup Language
Asirra	Animal Species Image Recognition for Restricting Access
DDOS	Distributed Denial Of Service
IP	Internet Protocol
TOR	The Onion Routing project
WEP	Wired Equivalent Privacy
HTTP	Hyper Text Transfer Protocol
GHz	Giga Hertz
MIT	Massachusetts Institute of Technology
IPB	Invision Power Board
URL	Uniform Resource Locator

Povzetek

Diplomsko delo opisuje pomen gesel in nevarnosti, ki prežijo na nevedne uporabnike s preprostimi gesli. Eden izmed glavnih namenov dela je osveščanje in praktičen prikaz, kako hitro se lahko dandanes najde takšne uporabnike.

Začetni del govori o težavah, ki se lahko pojavijo pri zlorabi osebnih podatkov in gesel. Vse preveč ljudi namreč svoje osebne podatke zaupa dvomljivim stranem ali pa koristi eno samo lahko uganljivo geslo. V ospredju je način iskanja takšnih ljudi in odkrivanju njihovih gesel. Uporabljen način za dosego takšnega cilja je poimenovan slovarski napad. Predstavljen je mehanizem takšnega napada, skupaj s predlogi za hitrejše doseganje boljših rezultatov. V nadaljevanju so prisotni rezultati izvedenega slovarskega napada na forum Fri-Info in analiza pridobljenih gesel.

Naslednja poglavja vsebujejo načine, kako do neke mere učinkovito preprečiti avtomatizirane procese. Obdelana je tudi tema, kako čim bolj uspešno zaobiti takšne zaščite. Sledeče poglavje se posveča predstavitvi aplikacije, ki je zmožna izvajati slovarski napad na forum Fri-Info. Natančno je predstavljena sestava in delovanje funkcij aplikacije ter skript, ki jih koristi. Izpostavljene so nekatere pomanjkljivosti aplikacije in tudi morebitne izboljšave.

Za konec so prikazane poti, ki jih je mogoče ubrati, z namenom anonimnega izvajanja slovarskega napada. Izdelava takšnih orodij in njihova uporaba je namreč protizakonita.

Ključne besede:

gesla, osebni podatki, avtomatiziran proces, slovarski napad, modul Mechanize, CAPTCHA

Abstract

This thesis is describing the meaning of passwords and dangers that ignorant users with simple passwords are faced with. One of the main purposes of this research is awareness and practical demonstration on how quickly one can find such users today.

The first section represents various issues that can result from the abuse of personal information and passwords. Too many people provide personal information to untrusted websites or continuously use the same simple password. One of the main focuses is a method used to identify such individuals and uncover their passwords. The method used for achieving this goal is called dictionary attack. The method demonstrates how this can be achieved, together with suggestions on how to achieve better results. Further in the thesis, the results of this method are presented using a specific Fri-Info forum, along with the analysis of the passwords discovered as a result.

Subsequent sections illustrate methods on how to effectively prevent automated processes as much as possible. In addition; the subject of bypassing such protection will also be covered. An application is also presented, which is capable of executing dictionary attack on Fri-Info forum. The composition and function of this application is discussed furthermore in detail together with additional scripts used by the same software. This application also lacks in some areas which are pointed out along with potential improvements.

Lastly, the ending section will demonstrate different ways one can can anonymously uncover weak passwords through the means of whats we know as dictionary attack. The creation and/or use of such tools is illegal.

Key words:

password, personal data, automated process, dictionary attack, Mechanize module, CAPTCHA

Poglavje 1

Uvod

Uporaba interneta in vsega kar je povezano z njim, se je dandanes močno razširila. Na spletu je dosegljivih ogromno različnih informacij, do katerih ima dostop vsak uporabnik. V primeru, da ne najde potrebnih odgovorov, se marsikdo zateče po pomoč na forum. Zaradi preproste namestitve, si lahko svoj forum postavi vsakdo, ki se vsaj malo znajde za računalnikom. Za vse ostale, pa obstajajo strani, ki samo z nekaj kliki ustvarijo želen forum, ki gostuje pri njih. To je prineslo porast števila forumov, pri čemer se marsikateri osredotočijo na določene informacije.

Za vsakega, ki hoče resno sodelovati na forumu, je potrebna registracija. Ob tem je potrebno vnesti geslo, elektronski naslov in ponekod tudi osebne podatke. S tem ni nič narobe, dokler si bodoči uporabnik foruma ne izbere lahko uganljivega gesla. Dodatno se nevarnost poveča, če nekomu uspe uganiti takšno geslo, saj potem dobi popoln nadzor nad računom. Takrat si lahko prebere morebitna zasebna sporočila, ki si jih je izmenjal z drugimi uporabniki. Sama vsebina sporočil je odvisna od uporabnikove nevednosti in neprevidnosti. Tako lahko obstaja možnost, da se iz sporočil razbere še marsikatero koristno informacijo, uporabno za nadaljnje zlorabe.

Smotrno je domnevati, da takšen neveden uporabnik koristi preprosta gesla tudi na drugih straneh, kjer je registriran. Če se razkrije njegov poštni naslov, je mogoče s pomočjo spletnih iskalnikov najti še kakšno stran, ki jo žrtev obiskuje. V najhujšem primeru pa se napadalec dokoplje do poštnega predala, kjer lahko povzroči največ škode. Preko poštnega predala lahko med drugim dobi dostop tudi do spletnega bančništva ali kaj enako pomembnega.

V časih, ko je širokopasovni internet dostopen skoraj vsakemu in ko cene zmogljivih računalnikov niso več pretirane, je mogoče ugibanje gesel, vnašanje podatkov in preverjanje rezultata lahko avtomatizirano. Takšna rešitev je

poimenovana bot, okrajšava iz besede robot. Robot je naprava, ki izvaja ponavljajoče se naloge, robot na osnovi programske opreme deluje na podoben način. S pomočjo botov se lahko pohitri iskanje nevednih uporabnikov s preprostimi gesli in hkrati tudi poveča možnost, da se takega uporabnika najde.

Poglavje 2

Osebni podatki na spletu

Nepremišljeni uporabniki spleta so hitro zavedeni v razkrivanje svojih osebnih podatkov ob registracijah, pri pošiljanju zasebnih sporočil in drugje. Nekdo, ki na kakršenkoli način, pridobi takšne podatke lahko naredi veliko škode. V najhujšem primeru lahko pride celo do kraje identitete. Po besedah raziskovalke pri informacijskem pooblaščencu, Eve Kalan, Slovenija[8] ni imuna na takšno obliko kriminala. Informacijski pooblaščenec pa se je v praksi že srečal s primeri klasične kraje identitete.

Zvijač, kako nepridipravi pridejo do osebnih podatkov je veliko. Eden izmed zadnjih načinov, ki je bil bolj opazen, se je zgodil pred kratkim. Na spletu se je pojavila stran[9], ki naj bi predstavljala švicarsko banko M&G. Za sodelovanje v nagradni igri, za katero so obljubljali deset tisoč evrov, je bilo potrebno vnesti osebne podatke in stran oglaševati na socialnem omrežju Facebook. Dokler je bila stran dosegljiva je bilo mogoče z imena domene razbrati, da gostuje na strani, ki nudi brezplačno gostovanje in ustvarjanje spletnih strani. Ime za takšna zavajanja je phishing ali spletno ribarjenje. Vendar pa to ni edini način za pridobivanje tovrstnih informacij. Iznajdljivi prevaranti se poslužujejo še različnih virusov, črvov in socialnega inženiringa.

Zato je izredno pomembno, da se tovrstne zvijače prepozna in da se uporabnike interneta osvešča o tej problematiki. Pri nas za osveščanje skrbi informacijski pooblaščenec. Na njegovi spletni strani si je mogoče ogledati različne brošure, ki so med drugim namenjene tudi preprečevanju kraje identitete. V takšnih zapisih so predstavljeni vzroki za krajo identitete, primeri resničnih zlorab, načini pridobivanja osebnih podatkov in navodila, kako se izogniti takšnim kriminalnim dejanjem.

2.1 Močna gesla

Namen gesel je, da onemogočijo dostop nepovabljenim osebam in dovolijo dostop samo lastniku računa. Da bi gesla opravljala svoje poslanstvo morajo biti ustrezne oblike in nikakor ne smejo biti lahko uganljiva. Ljudje si pogosto izberejo ravno takšna gesla, na primer njihovo osebno ime, rojstni datum, ime ljubljenčkov in podobno. Vsak, ki je sposoben memorizirati svojo ali drugo telefonsko številko, je zmožen tudi uporabljati močno geslo. Ravno močno geslo je nujnost na spletu, saj v najslabšemu primeru močno oteži delo nepridipravu.

Pri ustvarjanju močnih gesel je na spletu dostopnih mnogo orodij, ki svetujejo in ocenjujejo varnost vpisanega gesla. Takšno orodje bi bilo zelo priporočljivo vdelati na forum Fri-Info. Tako bi lahko bodoči uporabnik že med registracijo videl, kako primerno je njegovo geslo. Najbolj idealno pa bi bilo še, da bi takšno orodje preprečilo uporabo najbolj pogostih in preprostih gesel.

Močno geslo mora biti ustrezno dolgo, z dolžino vsaj šestih znakov. Vsebovati mora velike in male črke, številke ter različne simbole. Izogibati se je potrebno takšnim besedam, ki se pojavijo v slovarjih in sekvencam ponavljajočih se znakov. Priporočljiva je tudi uporaba mnemotehnike. To je veščina, ki poenostavi pomnjenje močnih gesel. Lažje si je namreč zapomniti geslo, če se posamezne črke iz gesla poveže z besedami, ki imajo določen pomen. Na primer „Zms1ezg!“ se lažje zapomni ob povezavi z „**Z m**nemotehniko **s1 e**nostavno **z**apomnim **g**eslo**!**“.

Izbrano močno geslo se ne sme hraniti v pisni obliki. Če ne gre drugače, se napisano hrani na varnem mestu. Nikakor pa ne v bližini računalnika, kjer lahko nekdo hote ali nehote zlahka najde napisano. Pomembna pa je tudi še uporaba več različnih gesel, na različnih straneh. Tako nepridiprav ob odkritju enega gesla, le-tega ne more uporabiti na drugih straneh, kjer je žrtev prijavljena.

2.2 Najbolj pogosta gesla na spletu

V mesecu oktobru leta 2009 se je na internetu pojavil spisek poštnih naslovov in pripadajočih gesel, pridobljen s pomočjo spletnega ribarjenja. Spisek je bil objavljen v treh delih, kar je skupno naneslo okrog 30 tisoč gesel. Zavedeni uporabniki so razkrili svoje podatke, misleč da se prijavljajo v poštni nabiralnik. Spisek se je pojavil na spletni strani www.pastebin.com, kamor se lahko anonimno objavljajo različna besedila. Poštni naslovi so pripadali različnim popularnim ponudnikom takih storitev, kot je na primer Hotmail, Gmail in Yahoo. Dokler spisek ni bil umaknjen s strani administratorjev, je bilo mogoče

shraniti preko 30 tisoč poštnih naslovov skupaj z gesli. Kljub temu pa je bila stran s pomočjo Google cache vidna še nekaj časa. V izognitev nadaljnjim težavam, so bili razkriti poštni naslovi zablokirani, gesla pa ponastavljena.

Na svoji spletni strani[2] je varnostni raziskovalec Bogdan Calin izvedel analizo 10 tisoč gesel iz prve objave. Najprej je bilo potrebno odstraniti ponavljajoča se gesla, saj so se nekateri uporabniki poizkusili večkrat prijaviti v svoj račun, misleč da so vnesli napačne podatke. Število računov se je tako zmanjšalo za 185 gesel. Najkrajše geslo je vsebovalo le en znak, najdaljše pa kar trideset. V odstotkih je največ gesel z dolžino 6 do 9 znakov. V povprečju pa je dolžina gesla obsegala 8 znakov.

Sestava gesel:

- Brez velikih tiskanih črk in številk, kot je na primer „iloveyou“: **3,713=42%**
- Samo črke, ki vsebujejo tudi velike tiskane črke, recimo „IloveYou“: **291=3%**
- Samo številke, od 0 do 9, na primer „123456“: **1707 = 19 %**
- Mešani znaki z velikimi tiskanimi črkami in številkami, primer je „iloveyou12“: **2655=30%**
- Mešani znaki, številke in ostali simboli, kot je „1LoveYou$%“: **565=6%**

Med dvajset najbolj pogosto uporabljanih gesel je veliko špansko zvenečih imen in gesel sestavljenih iz številk. Vendar pa te številke niso preveč varne, saj so lahko uganljive. Tako je na prvem mestu geslo „123456“, katero uporablja 64 ljudi. Drugo mesto je podobno, z 18 zadetki in geslom „123456789“. Sledijo imena in več enostavnih gesel sestavljenih iz številk, kot na primer „111111“,“000000“,“654321“, vendar pa so manj pogosta. Iz izsledkov je očitno, da se večina uporabnikov poslužuje preveč preprostih gesel. Še vedno si veliko ljudi izbere gesla, ki sestoje samo iz malih črk. Problematična pa je tudi še uporaba osebnih imen. Veliko poštnih naslovov namreč vsebuje uporabnikovo ime ter priimek. Tako se lahko ta dva podatka razbere in uporabi za vnos gesla.

Poglavje 3

Tehnika ugotavljanja gesel

Iznajdljivi napadalci se poslužujejo različnih poti za pridobivanje gesel. Prav tako je na spletu dosegljivo ogromno število vodičev in aplikacij za izvajanje raznovrstnih napadov, z namenom ugotavljanja gesel. Takšne aplikacije koristijo tako skriptni otroci, kot tudi strokovnjaki z namenom preverjanja varnosti na lastnem informacijskem sistemu. Moja aplikacija se poslužuje slovarskega napada, saj je en izmed boljših izbir za napad na forum.

3.1 Slovarski napad

Slovarski napad je eden izmed načinov, kako se uspešno prebiti skozi mehanizem, ki skrbi za avtentifikacijo. Takšen napad je bolj usmerjen na tisti krog ljudi, ki uporabljajo krajša gesla, besede iz slovarjev in drugačne ne varne oblike gesel. Uporablja se ga tudi pri preverjanju obstoja poštnih naslovov, z namenom pošiljanja vsiljene pošte. Za dosego tega cilja se uporabi vnaprej določene vrednosti, katere pravilnost se določa s poizkušanjem. Vnesene vrednosti so zapisane v slovarju in ponavadi niso na slepo izbrane. Za razliko od napada z grobo silo, izhajajo gesla iz besed, katere se lahko najde v slovarju – od tod tudi ime za napad. Za iskanje poštnih naslovov pa se uporabi slovar, ki vsebuje spisek najbolj pogostih uporabniških imen.

Napad poteka tako, da napadalčeva aplikacija iz slovarja zaporedno prebira vnose ter jih vnaša v obrazec za prijavo. V primeru, da je prijava uspela, se vneseni podatki shranijo, napad pa se nadaljuje. Večinoma pa so vneseni podatki nepravilni, zato aplikacija prebere naslednji zapis in nadaljuje z delom vse dokler ne pride do konca slovarja. Slovarski napad lahko ustavijo le močni zaščitni mehanizmi. Podobne metode se poslužujejo tudi razpošiljevalci reklamne pošte. Sprva določijo ime domene, ki postane tarča napada. Nato

se z uporabo slovarja, ki vsebuje različna uporabniška imena, sestavi poštni naslov, na katerega se pošlje sporočilo.

3.1.1 Trajanje in pohitritve slovarskih napadov

Čas izvajanja napada je odvisen od več dejavnikov. Najbolj očitna sta zmogljivost napadalčevega računalnika in internetna povezava. Zmogljivost računalnika dandanes ni več prevelik problem, saj so cene komponent sprejemljive. Aplikacijo za izvajanje tovrstnega napada se na zmogljivih računalnikih brez težav požene v več procesih. Na ta način se delo porazdeli. Seveda vsak proces uporablja svoj del slovarja.

Čas trajanja slovarskega napada je med drugim odvisen od dolžine slovarja, uspešnost pa prav tako od kvalitete uporabljanega slovarja. Ob takšnih napadih je mogoče uporabiti kombinacijo več računalnikov za hitrejše doseganje zadetkov. Na ta način se poveča število poizkusov gesel na sekundo.

3.1.2 Mavrične tabele

Zahvala za implementacijo gre Philippe Oechslinu, ki je priredil algoritem Martina Hellmana. Slednji je najbolj znan po Diffie-Hellman izmenjavi ključev. Vnesena gesla se ne shranjujejo kot goli tekst, pač pa se s pomočjo zgoščevalne funkcije pretvorijo v obliko, ki človeku ni več berljiva - hash. Tukaj pridejo v igro mavrične tabele. Vsak zapis v slovarju se pretvori z zgoščevalno funkcijo in shrani v datoteko, to je mavrična tabela. Takšni vnaprej pripravljeni hash-i se primerjajo z hash-om, katerega se želi pretvoriti v golo besedilo. Če se najde enak zapis, se posledično najde tudi golo besedilo. V nasprotnem primeru pa se iskanje nadaljuje.

Ideja je v večji uporabi prostora na trdem disku z namenom hitrejšega izvajanja napada. Mavrične tabele tako lahko zasedejo tudi po nekaj deset gigabajtov in vsebujejo ogromno število vnosov. Zaradi večje porabe prostora na disku, je mogoče napad izvesti hitrejše. Prav tako je takšen napad občutno hitrejši, kot pa napad z golo silo ali pa slovarski napad. Kvalitetne mavrične tabele so dosegljive na internetu, vendar pa si jih lahko kreira tudi uporabnik sam.

3.1.3 Kaj je salt

Najboljši način za izničenje napada z mavričnimi tabelami je uporaba naključnih vrednosti pri zgoščevanju gesel. Takšen proces je poimenovan salt.

Prvič se je pojavil v sedemdesetih letih prejšnjega stoletja, na operacijskem sistemu Unix. Algoritem vsebuje naslednje korake. Sprva se naključno ustvari neko besedilo, ki lahko sestoji iz poljubnih znakov, številk ali simbolov – tako imenovani salt. Le-tega se nikoli ne razkrije. Salt se nato zlepi skupaj z geslom ter uporabi v zgoščevalni funkciji. V primeru, da se uporablja naključno ustvarjen salt, se v bazo ločeno shranita hash in salt. Ob prijavi se iz baze prebere shranjen salt, izvede zgoščevanje in za konec primerja dobljeni hash z shranjenim. Če sta si enaka, je bilo vneseno geslo pravilno.

Uspešna uporaba mavričnih tabel je tako močno zmanjšana. Napadalec bi namreč moral za kreiranje uporabnih mavričnih tabel poznati salt. Da bi z grobo silo poizkusil ustvariti več različnih salt-ov za mavrične tabele, pa bi vzelo preveč časa in prostora.

3.1.4 Kako kreirati dober slovar

Izognil se bom raznim aplikacijam, ki na podlagi podanega teksta naredijo slovar. Tako bi namreč dobil preveč splošen slovar, pri katerem bi bil uspeh zelo dvomljiv. Na internetu je mogoče pridobiti že vnaprej narejene slovarje, vendar imajo v mojem primeru več negativnih kot pozitivnih lastnosti, zato jih ne bom uporabljal. Tuji slovarji zaradi več razlogov ne bi bili tako učinkoviti. Največji razlog je seveda jezik in s tem povezana tudi različna kultura. Verjetnost, da bi uporabnik imel geslo v tujem jeziku, je prav gotovo manjša, kot pa če bi imel geslo zapisano v materinščini. Nekdo ki ne govori nobenega tujega jezika, si verjetno ne bi izbral gesla v drugem jeziku. Prav tako bi lastnik računa ne uporabil gesla, ki je v njegovi kulturi neznan - na primer beseda iz žargona, ki je v drugih kulturah brez pomena. Dodatna slabost je tudi v temu, da je bolje uporabiti namenske slovarje, kot pa univerzalen slovar. Namenski slovarji namreč vsebujejo tematsko določena gesla, na primer pogosta moška osebna imena, citate, fraze in podobno. Vendar pa je ponovno potrebno misliti na jezikovno oviro.

Dobri slovarji lahko predvidijo pogoste vzorce gesel, kot je na primer dodajanje številk na začetku ali koncu gesla ter pogoste napake pri tipkanju določenih gesel. Pogosto se poslužujejo še tako imenovanih mutacijskih filtrov, ki so namenjeni preoblikovanju gesla v drugačno obliko. Eden izmed bolj uporabljanih filtrov je pretvarjanje določenih črk v številke (tako imenovani „l33t5p3ak“) ali pa dodajanje oziroma odstranjevanje črk z veliko začetnico. Pretirana uporaba takšnih filtrov lahko poveča obsežnost slovarja, s tem pa se tudi podaljša izvajanje napada.

3.1.5 Opis mojega slovarja

Moj slovar ni obsežen, saj vsebuje le nekaj ključnih gesel. Sicer bi lahko zgradil večji slovar, vendar je moj namen samo ilustrirati uporabo aplikacije ter preveriti, kako varna gesla si izbirajo uporabniki foruma Fri-Info. Ob dodajanju še kakšnega gesla, bi zagotovo našel še nekaj uporabnikov več, vendar zelo verjetno ne bi imel toliko zadetkov, kot pri sedanjem slovarju. Pri izbiri gesel sem imel v mislih prej opisane slabosti. Prav tako sem se zanašal predvsem na veliko verjetnost, da je dosti ljudi prelenih, da bi določili varna gesla in si jih tudi zapomnili.

Vsa gesla izpolnjujejo zahtevo foruma, da so dolga vsaj 6 znakov. Gesla so enobesedna, saj sem se na ta način izognil dodatnim zapletom ob vnašanju podatkov v forme. Slovar vsebuje le eno besedo v tujem jeziku, ki pa je zelo prepoznavna. Dva vnosa sta namenska in uporabna le na tej strani, saj vsebujeta ime spletne strani oziroma naziv foruma. Pomemben del slovarja je še vnos uporabniškega imena za geslo. Vendar pa je ta del mogoče narediti samo preko kode, zato ga tudi ni videti med drugimi zapisi. Preostala gesla sem izbral po zgledu najbolj pogostih gesel, ki jih izberejo uporabniki. Nekaj izmed teh pogostih gesel sem kombiniral ali pa uporabil njihov prevod.

Vsebina mojega slovarja:

- 123456
- 123123
- internet
- qwertz
- qwerty
- abc123
- geslo123
- password
- fri-info
- friinfo

3.1.6 Rezultati iskanja po slovarju

Zaradi varovanja podatkov uporabnikov, je po dogovoru preverjanje gesel izpeljal administrator foruma Ahac Sedušak. Pri tem ni uporabil moje aplikacije, ampak je izvedel iskanje po forumovi bazi s predlaganimi besedami iz slovarja. Za rezultat se je izpisalo samo število pojavitev, brez kakršnihkoli uporabniških imen. Na ta način so se zavarovali osebni podatki, hkrati pa sem dobil rezultate o uspešnosti slovarja.

Beseda	Število zadetkov
123456	20
123123	0
internet	2
qwertz	2
qwerty	1
abc123	5
geslo123	2
password	2
fri-info	3
friinfo	3
Uporabniško ime kot geslo	49

Tabela 3.1: Rezultati iskanja po slovarju z aplikacijo.

V času iskanja po geslih je bilo pregledanih 7600 uporabnikov. Sem spadajo tudi tisti uporabniki, ki so se samo registrirali, vendar pa niso potrdili svojega poštnega naslova. Skupno število odkritih gesel je **89** ali v odstotkih **1,17%**. Slovar obsega 10 različnih besed in vnos uporabniškega imena kot geslo. Večji odstotek bi prav gotovo dosegel z razširitvijo slovarja. Sedaj pa vsebuje resnično le nekaj najbolj pogostih in preprostih gesel. Rezultate bi lahko še dodatno izboljšal z uporabo mutacijskih filtrov, na primer z spreminjanjem velikih začetnic pri določenih geslih.

Največ uspeha je prineslo vnašanje uporabniškega imena za geslo z 49 zadetki. Verjetno bi ta možnost ostala na prvem mestu, tudi če se bi slovar razširilo z drugimi vnosi.

Takoj na drugem mestu je geslo „123456“, katerega na forumu uporablja 20 ljudi. Trditev, da je geslo „123456“ najbolj pogosto na internetu, se je na forumu Fri-Info izkazala za dokaj pravilno.

Pri preostalih zadetkih je pogostost precej manjša v primerjavi s številom

zadetkov prvih dveh gesel. Ta gesla se pojavljajo pri petih in manj uporabnikih. Verjetno obstaja še kakšno uganljivo geslo, ki je med tema dvema ekstremoma, vendar ga ni v mojem slovarju. Zanimivo je, da ima 6 uporabnikov za geslo kar ime spletnega foruma, ki ga obiskuje. Obe gesli, „friinfo“ ter „fri-info“, sta zapisani z malimi črkami. Zagotovo obstaja še kakšen uporabnik več z enakim geslom, vendar z drugačno kombinacijo velikih oziroma malih črk.

Razlog za izbiro preostalih uporabljenih preprostih gesel je verjetno njihova priročnost – hitro se jih je mogoče zapomniti in natipkati. Vsaj sodeč po medsebojni bližini črk, ki sestavljajo na primer geslo „qwertz“. Zanimivo bi bilo preveriti še moč izbranih gesel pri poštnih nabiralnikih takšnih uporabnikov. Obstaja namreč verjetnost, da so enako nedomiselni in nepremišljeni tudi pri izbiri preostalih gesel.

Poglavje 4

Zaščita pred avtomatiziranimi dostopi

Izraz CAPTCHA je kratica za Completely Automated Public Turing Test To Tell Computers and Humans Apart (Popolnoma avtomatiziran Turingov test, ki razloči med računalniki in ljudmi), kar poenostavljeno pomeni ločevanje ljudi od avtomatiziranih računalniških procesov. Metoda CAPTCHA ščiti internetne storitve pred računalniškimi procesi z zli nameni. Deluje po principu izziv – odgovor. Izziv je slika z naključno generiranim besedilom, odgovor pa je prepis tega teksta. Slika CAPTCHA ponavadi vsebuje neko besedilo, ki ne sme biti preveč lahko berljivo in je zapisano z različnimi pisavami ter velikostmi. Besedilo je tako lahko ukrivljeno, v različnih barvah, skozi njega potekajo različne črte ali pa je nameščen na neko ozadje. Na ta način se z veliko verjetnostjo zagotovi, da teksta ni mogoče računalniško obdelati in ga prebrati z ustreznimi OCR programi. Uporablja pa se tudi zvočna CAPTCHA, vendar je manj pogosta kot pa slikovna različica. Namenjena je ljudem, ki imajo težave z vidom. Ob pritisku na gumb se sliši glas, ki pove nekaj besed, uporabnik pa jih mora vnesti. Da je test varen, se v ozadje doda različne šume in uporablja različne glasove za branje.

Vendar pa tekstovni in zvočni način nista edina tipa CAPTCHA, čeprav sta najbolj pogosta. Počasi postajajo opazne tudi slikovne različice CAPTCHA. Namesto besedila se prikaže več slik, med katerimi mora uporabnik izbrati na primer slike letal. Eden izmed takšnih izdelkov je Microsoft Asirra. Prikazuje slike psov in mačk, ki so pridobljene iz www.petfinder.com, uporabnik pa mora označiti ali pse ali mačke. Zaradi velikega števila slik ni možno, da bi si napadalec naredil bazo slik, ki bi se lahko prikazale v testu. Takšen test je varen, saj je prepoznavanje slikanih objektov zaenkrat še slabo razvito. Prav

tako pa je zagotovo tudi manj suhoparen kot pa prepisovanje besed.

Eden izmed bolj pogostih ponudnikov tovrstne zaščite je dostopen na strani www.recaptcha.net. Od preostalih se razlikuje po eni pomembni lastnosti. Pri prepisovanju obeh besed v ustrezen prostorček se namreč ne izvaja samo CAPTCHA verifikacija. Lastniki te strani namreč želijo pomagati pri digitaliziranju različnih tekstov, saj ponavadi programi za prepoznavo teksta niso učinkoviti pri starejših in bolj obledelih besedilih. Tako je pri reCAPTCHA testu ena beseda kontrolna (pravilno prepoznana) druga pa neznana. V primeru, da je kontrolna beseda pravilno vnesena, se neznana beseda shrani kot možen pravilni odgovor. Ko se se za isto neznano besedo trikrat pojavi enak prepis, se ta beseda shrani kot kontrolna. V primeru, da se pri isti neznani besedi pojavljajo manjša odstopanja, se besedo prikaže še večjemu številu ljudi in po seštevku enakih odgovorov določi pravilno besedo. Uporabnik lahko vsako besedo, ki je neberljiva, nadomesti z drugo s pritiskom na določen gumb. Ko šest uporabnikov za isto besedo pritisne ta gumb, se ta beseda odstrani iz obtoka in je označena kot neberljiva. Tako dobijo veliko bolj natančne rezultate, kot pa pri navadnih programih za prepoznavanje besedila.

Ravno iz tega razloga je septembra 2009 Google kupil reCAPTCHA[1]. S takšno pridobitvijo so namreč izboljšali svojo tehnologijo za prepoznavanje teksta. V svojem času delovanja, je reCAPTCHA pridobil kar zajetno bazo prepisanih besed s pomočjo ljudi, ki na ta način učijo računalnike. Ocenjujejo namreč, da ljudje vsak dan pretipkajo približno sto milijonov testov CAPTCHA[14], ki se uporablja na sto tisočih straneh! Google z novim nakupom cilja predvsem na projekte kot so Google News in Google Books. Za razliko od skeniranega besedila se lahko po računalniško prebranem tekstu išče, se ga lažje ureja in podobno.

Slika 4.1: Primer CAPCHA testa iz www.recaptcha.net.

4.1 Primeri zlorab, ko ni zaščite CAPTCHA

Tovrstno zaščito pogosto najdemo ob registracijah novega računa, dodajanju komentarjev na nek zapis, spletnih nakupih, anketah in še marsikje drugje. V primeru, da se takšna zaščita ne uporablja, obstaja velika verjetnost različnih zlorab. Najbolj pogosti so pošiljatelji neželene pošte, ki reklamirajo določene izdelke ali spletne strani. Če na primer ob registraciji novega računa na forumu ni prisotna CAPTCHA, se lahko avtomatizirano registrira nov račun, ki po vseh delih foruma objavlja vnaprej določena reklamna sporočila. Dodatna težava se pojavi, ko se registrira več takih računov, zaradi česa lahko forum hitro uide izpod nadzora in vsebuje več reklamnih sporočil, kot pa ostalih objav. Podobno se dogaja pri komentarjih na blogih, kjer pa je tovrstnih zlorab še več, saj se za nov komentar ponavadi ni potrebno registrirati.

Ob registraciji novega poštnega računa je, pri vsakem večjem ponudniku te storitve, vedno prisotna CAPTCHA. V primeru, da je ne bi bilo, bi lahko nepridipravi izvajali avtomatizirano registriranje poštnih naslovov. To maso novih računov pa bi lahko uporabljali za nadaljnje ilegalne aktivnosti. Podoben problem bi nastal, če se CAPTCHA ne bi pojavila ob večkratnem nepravilnem vnosu gesla za dostop do pošte. S tem se prepreči slovarske napade, ki bi samodejno vnašali vnaprej določena gesla.

Zanimiv primer je anketa iz leta 1999, objavljena na www.slashdot.org, ki je spraševala po najboljši ameriški računalniški fakulteti. Hitro so se našli študenti fakultete Carnegie Mellon in našli način goljufanja, ki je za njihovo šolo oddal veliko število glasov. Ko so nenadno rast glasov opazili študenti MIT, so tudi sami našli svojo pot za avtomatizirano oddajanje glasov. Na koncu je vsaka izmed fakultet imela[19] preko dvajset tisoč glasov, preostale pa okrog tisoč. Enak primer se je zgodil tudi pri nas v letu 2008. Na domači strani časopisa Žurnal[6] je v enem samem dnevu z enega IP naslova prišlo ogromno število glasov za isto izbiro. Kasneje se je izkazalo, da je lukнjo v sistemu izkoristila določena politična stranka, ki je glasovala sebi v korist.

Vedno bolj se uveljavlja tudi zaščita poštnega naslova, pred nezaželenimi pošiljatelji. Pošiljatelji neželene pošte najdejo osebne poštne naslove preko dopisnih seznamov, klepetalnic, različnih imenikov in na številne druge načine. Stran www.recaptcha.net med drugim omogoča varno javno objavljanje osebnega poštnega naslova. Ta ni javno viden vsakomur, vendar je potrebno vnesti CAPTCHA tekst, da se razkrije celoten naslov. Na ta način je elektronski naslov varen pred zbiralci naslovov.

4.2 Kako zaobiti CAPTCHA zaščito

Glavni razlog za razbijanje CAPTCHA zaščite je avtomatizirano izpolnjevanje form brez kakršnihkoli ovir. Zaradi vse pogostejših zlorab se za tovrstno zaščito odloča vse več skrbnikov spletnih strani. Ravno zato so različne oblike CAPTCHA testov pod drobnogledom ljudi, ki iščejo načine kako jo zaobiti. Poti za dosego tega cilja je več, nekatere so bolj uspešne, druge spet manj. V nadaljevanju so opisani trije takšni načini.

4.2.1 Uporaba drugih ljudi s pomočjo zvijač

Napadalci poznajo več možnosti, kako zaobiti CAPTCHA zaščito po čimhitrejši poti. Način, kako zlomiti takšno zaščito, pa ni nujno omejen samo na kodo, ki po določenih postopkih prepozna besedilo.

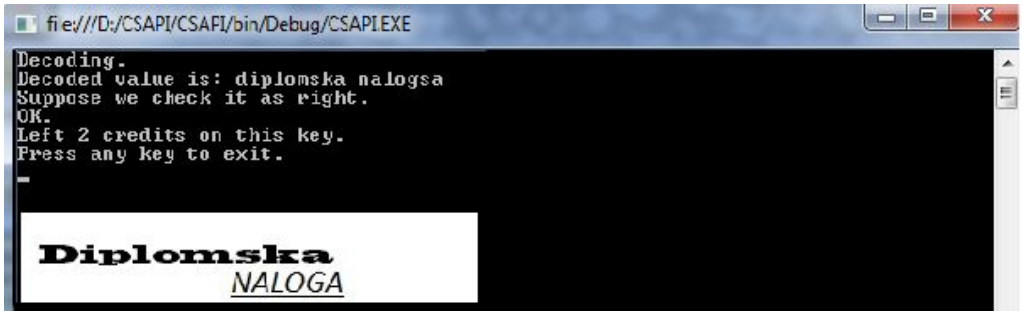
Eden izmed bolj duhovitih načinov se je pojavil s trojanskim konjem, poimenovan TROJ_CAPTCHAR.A, bolje znan kot Melissa Strip (slika 4.2). Ob namestitvi se uporabniku prikaže okno, ki vsebuje preprosto igro. Na samem začetku se prikaže slika ženske. Navodila povejo uporabniku, da bo „Melissa"po vsaki pravilno prepisani besedi odvrgla en kos svojega oblačila. Pri tem se na zaslonu izpiše CAPTCHA koda, katero je potrebno prepisati in pritisniti gumb za potrditev. V ozadju se zatem tekst prenese na neznan strežnik, dekodirani tekst pa lahko napadalec uporabi za svoje namene. Zavedeni uporabnik na ta način ne vedoč pomaga pri razbijanju CAPTCHA zaščite. V PandaLabs, kjer so med prvimi zasledili tega trojanskega konja, so ugotovili da testi CAPTCHA izhajajo iz spletne strani Yahoo[13]. Eden izmed možnih namenov zbiranja prepisanih besed je, da si nekdo izdeluje bazo Yahoojevega varnostnega mehanizma CAPTCHA.

Slika 4.2: Posnetek zaslona ob izvajanju TROJ_CAPTCHAR.A.

4.2.2 Človeško vnašanje besedila

Obstaja možnost, da se za majhen denar najame skupino ljudi, ki prepisujejo CAPTCHA besedilo. Možno pa je tudi plačati za reševanje testov tipa Asirra. Za podjetja, ki nudijo takšne storitve, se je prijel izraz Captcha Farm ali Turing Farm. Cene za takšne usluge niso visoke. Za tisoč rešenih CAPTCHA zaščit najcenejši ponudnik zaračuna samo dva dolarja[17]. Ob registraciji dobi uporabnik API, katerega vgradi v svojo že obstoječo kodo. Tako lahko koristi plačane storitve. Stran www.bypasscaptcha.com omogoča brezplačen preizkus delovanja API z enourno omejitvijo in desetimi prepisanimi CAPTCHA besedili. Kolikor sem preizkusil delovanje, nisem opazil nobenih težav, razen če sem naredil CAPTCHA besedilo in pri tem uporabil šumnike. Odziv na resnične CAPTCHA primere je bil v skoraj vseh primerih pravilen (slika 4.3).

Dobra stran takšne rešitve je neprestana dosegljivost delavcev in preprosta uporaba API-jev. Slabost pa je v počasnosti prejemanja odgovora. Ker tekst prepisujejo ljudje, je normalno da je za reševanje enega testa potrebnih kar nekaj sekund. Ta čas pa se še poveča, če je v čakalni vrsti še več slik potrebnih reševanja. Naslednja slabost pa leži v stroških. Ti stroški se sicer ne poznajo preveč ob manjših napadih, v mojem primeru bi za en obhod vseh pet tisoč uporabnikov foruma Fri-Info (če bi imeli vsi vključeno CAPTCHA zaščito zaradi prevečkratnega neuspešnega prijavljanja), plačal minimalno 10 dolarjev. Vendar pa samo z enim obhodom ne bi dosegel cilja – preizkusiti več različnih gesel in tako povečati verjetnost, da se najde uporabniški račun s preprostim geslom. Tako bi potreboval več obhodov skozi vse uporabnike, kar bi še povečalo skupni račun, vendar pa še vedno ne bi prišel do takšne številke, kot pri večjih projektih.



Slika 4.3: Posnetek zaslona ob izvajanju API z www.bypasscaptcha.com, skupaj s poslanim CAPTCHA testom.

4.2.3 Vnašanje besedila programabilno

Zaradi velike razširjenosti reCAPTCHA je ta toliko bolj pod drobnogledom ljudi, ki jo poizkušajo premagati. Trenutno je najbolj uspešen Chad Houck, katerega koda dosega do 30% pravilno računalniško prebranega teksta. Vendar pa kljub temu, da je opis algoritma in preostali pripomočki javno dostopen na njegovi spletni strani, zaenkrat še ni bilo opaziti, da bi se aktivno uporabljal. Če se bi uporaba algoritma[3] močno razširila, bi Google zagotovo poskrbel za ustrezne spremembe testa. Tako bi postal algoritem za reševanje reCAPTCHA testov ničen. Čas porabljen za prepoznavo besedila je primerljiv s človeškim prepoznavanjem, na računalniku z 2.53 GHz procesorjem, je bilo potrebnih skoraj osem sekund.

Postopek:

1. Algoritem prične z odstranjevanjem popačenja do neke mere. Popačenje besedila je vidno v valovitem zapisu teksta, samo besedilo pa ima ob robu še malo šuma. Valovito besedilo je potrebno zravnati za lažjo nadaljnjo obdelavo. Algoritem, ki skrbi za to, je poimenovan „blanket algorithm“. Deluje tako, da išče najbolj prilegajočo se tangento na spodnjem delu slike. S primerjavo med sosednjimi slikovnimi pikami izdeluje naklone, katere potem med seboj povezuje, za konec pa sledi še glajenje črte. Težave povzročajo črke s podaljšanim spodnjim delom, kot na primer q, y ali g. Ko je določena meja, se slikovne pike prenašajo po osi y navzdol tako, da se dobi izravnano besedilo.

Slika 4.4: Vizualni prikaz delovanja „blanket“ algoritma.

2. Sledi razdeljevanje slike na manjše dele, ki verjetno vsebujejo znake. Ločevanje črk med seboj je dosti težavno, saj so črke enakih barv, prav tako pa se lahko tudi prekrivajo. Uporabi se algoritem podoben prejšnjemu, ki išče upadanja linije, in zagotovi, da se ločna linija črke čim tesneje oklepa. Zatem se lažje razmeji posamezne črke med seboj.

3. Kose slike se analizira za določitev čim bolj podobnega znaka in nato še besed. Vsaka izmed teh ločenih črk se potem primerja z vnaprej

Slika 4.5: Ločevanje besed na posamezne črke.

pripravljenimi črkami, ki so bile že prepoznane. Preostali tipi CAPTCHA uporabljajo naključno kombinacijo črk, ki naj nebi imeli nobenega pomena. Pri reCAPTCHA je vsaka prikazana beseda tudi v slovarju angleškega jezika. To omogoča uporabo slovarskega napada, sam slovar za napad pa je prav tako dosegljiv na spletni strani od pisca kode. Uporabi se dinamičen algoritem, ki za vsako besedo vzeto iz slovarja, naredi spisek vsake posamezne črke. Vsaka izmed teh črk se potem primerja s prej razrezano črko. Beseda, ki po določenem postopku doseže največji rezultat, se smatra kot prepoznana beseda.

Poglavje 5

Pregled foruma

Nove različice spletnih forumov vsebujejo veliko že vgrajenih varnostnih mehanizmov. Te omogočajo zaznavanje raznih nepooblaščenih vstopov in podobno. V nadaljevanju se bom osredotočil na neuraden forum fakultete FRI. Forum Fri-Info je namenjen izmenjavi informacij med študenti Fakultete za računalništvo. Trenutno ima nekaj več kot pet tisoč osemsto uporabnikov.

Spada v vrsto phpBB, je brezplačen, odprto koden ter spada med najbolj razširjene tipe forumov. Zaradi teh razlogov je na voljo veliko število modifikacij kode, ki omogočajo uporabo različnih novih funkcij. Med drugim tudi številni dodatki za preprečevanje avtomatiziranih dostopov, ki omogočajo enostavno objavljanje neželenih sporočil. Ne obstaja pa dodatek, ki bi lahko zaznal abnormalno obnašanje uporabnikov.

5.1 Zaščita uporabljena na forumu

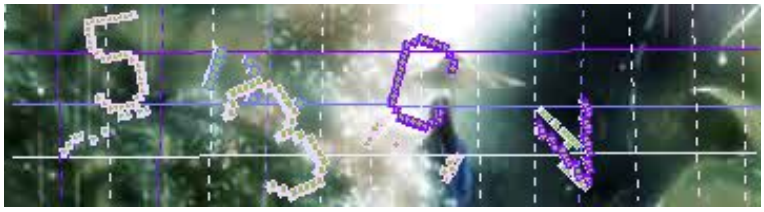
Fakultetin neuraden forum nima posebnih dodatkov za večjo varnost, uporablja pa varnostni sistem CAPTCHA. Sprva ni bila v uporabi, vendar so jo omogočili, ko se je na forumu močno povečalo pojavljanje novih tem, v katerih se je reklamiralo različne produkte. Ravno zaradi uporabe CAPTCHA je izvajanje moje aplikacije malo oteženo. Za zajezitev problema reklamnih sporočil ni mogoče objavljati prispevkov, če uporabnik ni registriran. Onemogočen je tudi ogled spiska obstoječih uporabnikov za neprijavljene goste.

Prva uporabljana CAPTCHA je bila bolj preproste narave (slika 5.1) in jo je bilo možno računalniško prebrati, tako torej ni služila svojemu namenu. Po valu avtomatiziranih registracij in nezaželenih sporočilih je prišla v uporabo bolj kompleksna CAPTCHA (slika 5.2). Ta je zagotovo veliko bolj varna, saj se pošiljatelji reklamnih sporočil niso več pojavljali. Uporabljajo se različna

ozadja, ki pa nikoli niso v eni barvi. V ozadju je prav tako videti šum. Črke so izpisane pod različnimi koti in so v različnih barvah. Prekrivajo pa jih še vodoravne in navpične črte.

Slika 5.1: Primer slabe CAPTCHA zaščite.

Forum deluje namreč tako, da je ob vsaki novi registraciji potrebno prepisati CAPTCHA, ki se ob nadaljnji uporabi foruma ne prikazuje več. Ni prisotna ob uporabi iskanja po forumu, dodajanju novih objav, spreminjanju gesla ter ostalih podatkov in pošiljanju zasebnih sporočil. S tem se odprejo nove možnost za uporabo botov.



Slika 5.2: Primer zaščite CAPTCHA uporabljene na forumu.

V mojem primeru ni potrebno registrirati novih računov, saj bi z ugibanjem ugotavljal gesla že obstoječih uporabnikov. Ravno iz tega razloga se ob večkratnem neuspešnem prijavljanju začne prikazovati CAPTCHA. Za vsak račun se lahko trikrat vnese napačno geslo, potem pa se vklopi prikazovanje zaščitnega mehanizma. Lastnik računa bi moral ob naslednji prijavi uspešno rešiti CAPTCHA test, kljub temu da bi uporabil pravilne podatke. Kaj takega pa bi zagotovo sprožilo različne pritožbe uporabnikov in pritegnilo pozornost upravljavcev strani. Toda tudi ta zaščita ima svojo pomanjkljivost. Števec nepravilnih vnosov gesel se ponastavi, ko lastnik računa vnese pravilno geslo in pretipka CAPTCHA besedilo. Zatem se lahko ponovno izvede tri obhode skozi takšne račune. Tako se tudi izogne dodatni kodi, katere namen bi bil branje CAPTCHA slik, kar bi bil velik zalogaj sodeč po videnih slikah CAPTCHA. Res pa je, da bi v krajšem času bilo ponovno aktiviranih veliko manj računov. Večinoma bi to bili samo tisti, ki so bolj redni obiskovalci foruma. Vendar pa v primeru, da za takšen napad ne bi bilo nobene časovne omejitve, ta pomanjkljivost ni prehuda.

Poglavje 6

Pridobivanje gesel z aplikacijo

Grafični vmesnik ter koda v ozadju je spisana v programskem jeziku C#. Uporabljajo pa se tudi skripte narejene v programskem jeziku Python. Namenjene so preprostemu prijavljanju uporabnikov na forum, shranjevanju uporabniških imen in zasebnih sporočil. Vsaka izmed skript se poslužuje modula Mechanize. Za uspešno izvajanje je tako potrebno imeti najprej nameščen Microsoft .Net Framework. Naslednja zahteva je instalacija Pythona, potrebno pa je določiti še pot do mesta, kjer je nameščen prevajalnik. Zaradi tega se lahko izvaja skripte, ne glede na njihovo lokacijo. Zadnja nujnost je namestitev modula Mechanize. Najlažji način je uporaba modula easy_install, ki skrbi za preprosto shranjevanje, nameščanje ter urejanje Pythonovih paketov. Ob izpolnitvi teh zahtev imamo na razpolago delujoč bot, prirejen za delovanje na forumu Fri-Info.

6.1 Kaj je bot

Glavni del moje aplikacije so skripte narejene v programskem jeziku Python in uporabljajo modul Mechanize[7]. Vsaka izmed petih skript izvaja svojo nalogo. Združitev grafičnega dela, vključno s kodo v ozadju, s skriptami sestavlja bot.

V to kategorijo spadajo programi, ki se povezujejo na splet ter pridobivajo podatke, katere se nato obdeluje. Boti ponujajo različne funkcije za avtomatizirano opravljanje ponavljajočih se nalog. Hitrost opravljanja takšnih nalog je neprimerljivo večja od človeškega opravljanja enakih nalog. Na spletu se uporabljajo boti z različnimi nameni. Spletni iskalniki se poslužujejo botov, ki na primer indeksirajo spletne strani z namenom ponujanja čim boljših iskalnih rezultatov. Seveda pa uporaba ni omejena samo na dobro miselne naloge. Bote se izrablja za goljufanje pri spletnih igrah, koordiniranih DDOS napadih,

širjenje zlonamerne kode in še drugje.

6.2 Opis in uporaba skript

Vse Python skripte so dodane med Resources in so pravzaprav vdelane v izvršljivo datoteko. Na ta način se izogne komplikacijam, saj skript ni potrebno posebej prenašati in jih postavljati na določeno mesto, ki ustreza vnaprej zapisani poti. Skripte se na začetku izvajanja prenesejo v mapo, kamor tudi operacijski sistem začasno shranjuje preostale datoteke. Ob izhodu iz aplikacije le-ta za seboj počisti in izbriše ustvarjene datoteke.

Za izvajanje Pythonovih skript znotraj programskega jezika C# sem izbral eno izmed lažjih poti (priloga A.1). Uporabi se razred Process, ki je del System.Diagnostics imenskega prostora. Omogoča tako dostop do lokalnih procesov, kot tudi zagon in ustavitev lokalnih procesov. Sprva je potrebno določiti lastnosti, ki se bodo posredovale metodi za pričetek procesa. Najpomembnejša lastnost je določitev imena datoteke, ki se bo zagnala. V tem primeru je to python.exe, zato je tudi pomembno, da je določena njegova pot v spremenljivkah okolja.

V nadaljevanju se določi argumente, ki se bodo podali klicu datoteke. Na prvem mestu je vedno pot do skripte, ki se bo uporabila. Na preostalih mestih pa so ostale spremenljivke, ki so potrebne v želeni skripti. Na primer uporabniško ime, geslo ali pa pot za shranjevanje datoteke. Skoraj vse skripte vračajo nekatere vrednosti, tako da jih izpišejo, torej jih je potrebno ujeti in obdelati. To omogoča lastnost, ki poskrbi za preusmerjanje standardnega izpisa. Enako se omogoči preusmerjanje napak. Sedaj je vse pripravljeno in lahko se kliče metodo Start(), ki sproži izvajanje procesa.

Pri klicu nekaterih skript je potrebno zajeti izpis, pri drugih pa je potrebno samo počakati na njihovo zaključitev. Prvo možnost se reši s klicem ustrezne metode, ki spada v imenski prostor System.Diagnostics. Metoda WaitForExit() skrbi za to, da trenutna nit čaka toliko časa, dokler se novo ustvarjeni proces ne zaključi. Uporablja se na primer pri skripti, ki shranjuje zasebna sporočila in poštni naslov uporabnika. Ta skripta ne izpisuje ničesar, za razliko od skripte ki skrbi za prijavljanje v forum. Ob uporabi takšne skripte, je potrebno za klicem metode Start() uporabiti while zanko. Ta zanka koristi metodo StandardOutput, ki omogoča branje posamezne vrstice, katero izpiše novo ustvarjeni proces. Zanka se izvaja vse dokler ni prebrana vrstica nična. Tako se zajame ves izpis iz skript in se ga uporabi za izpisovanje ter ostalo. Na koncu je potrebno proces še zapreti in sprostiti sredstva.

6.3 Grafični vmesnik

Za lažje delo z skriptami za prijavljanje sem naredil grafični vmesnik (slika 6.1). Uporabniku omogoča večjo preglednost, enostavno poganjanje skript v ozadju in nazoren izpis rezultatov, ki pa se prikazujejo realno časovno. Glavni funkciji v aplikaciji sta namenjeni vnašanju podatkov za prijavo, z namenom pridobivanja računov s preprostimi gesli, kar je rdeča nit diplomskega dela. Na razpolago pa sta tudi še dve uporabni funkciji.

Prva omogoča shranjevanje zasebnih sporočil ter poštnega naslova od uporabnika, katerega geslo je znano. Naslednja funkcija je uporabno orodje, ki ustvari spisek vseh registriranih uporabnikov foruma Fri-Info.

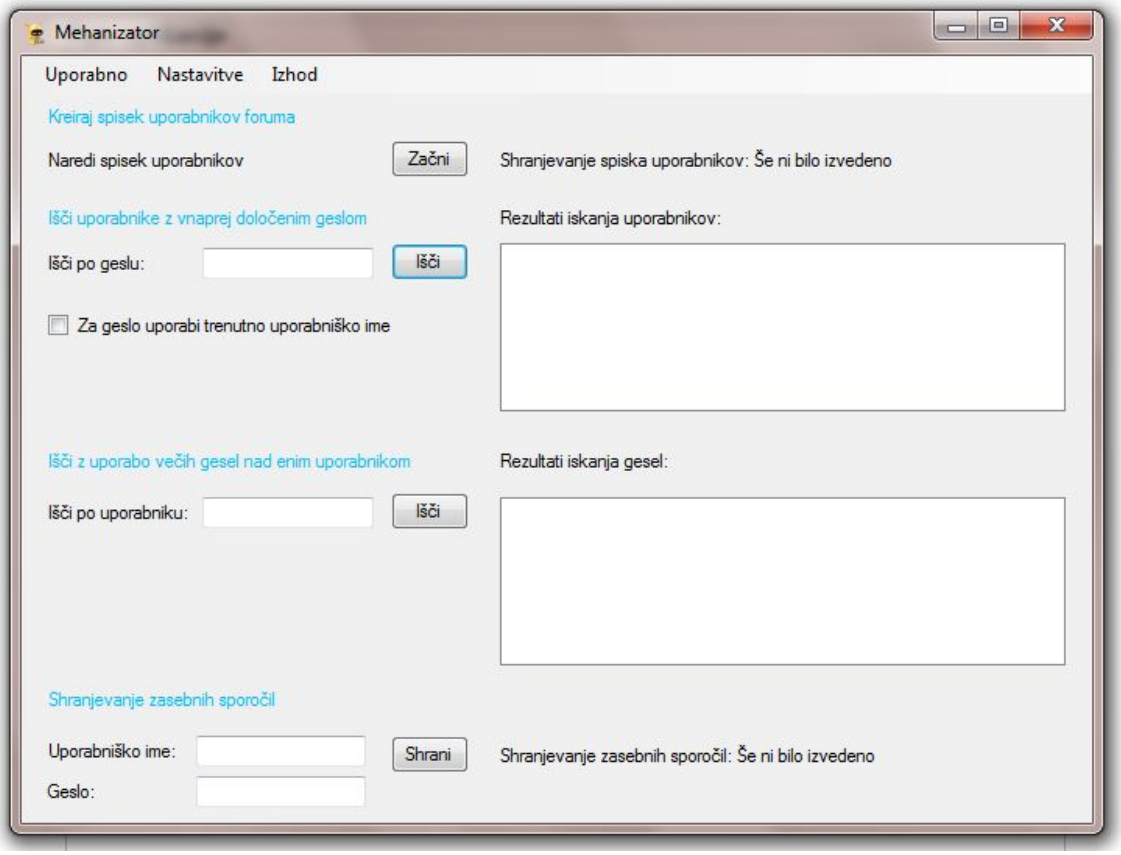
Rezultati se prikazujejo v dveh seznamskih poljih. V prvemu se vnaša eno samo geslo več različnim uporabnikom foruma. Drugo seznamsko polje pa prikazuje rezultate iskanja različnih gesel za samo enega uporabnika. V primeru, da polja nista prazna je mogoče z desnim klikom nad označenim elementom izvesti prijavo. Ob tem se odpre nova forma, ki služi kot brskalnik, na naslovu za prijavo v forum. Podatki označenega elementa se avtomatično vnesejo v ustrezna polja, uporabnik mora samo še pritisniti na gumb za prijavo.

Menijska vrstica ponuja nekaj manj pomembnih izbir, kot na primer odpiranje privzetega brskalnika na strani za prijavo v forum in pa možnost za izhod iz aplikacije. Izbira „Nastavitve“ vsebuje bolj pomembne elemente. S klikom na ustrezen element lahko uporabnik nastavi pot za shranjevanje najdenih uporabnikov in gesel. Vnaprej je mogoče tudi izbrati pot do obstoječih spiskov gesel in uporabniških imen, ki služita kot slovarja pri napadu. Zaradi forumove omejitve, ki onemogoča ogled spiska uporabnikov gostom, je dodan še element za določanje podatkov obstoječega računa. Ob kliku na to izbiro se odpre novo okence, v katerega se vnese uporabniško ime ter geslo. Ob potrditvi se vneseno shrani in uporabi pri ustvarjanju spiska uporabnikov foruma.

Za boljšo informiranost uporabnika so v glavnem oknu oznake, ki obveščajo o uspešni ali neuspešni zaključitvi izvajanja skript. Številna sporočilna okna služijo kot obvestila ob nepravilni uporabi.

6.4 Začetno preverjanje in config datoteka

Ob vsakem zagonu se sprva preveri, če je prisotna config datoteka. Njena naloga je ohranjanje nastavitev med večkratnimi izvajanji. V njej se namreč nahajajo različni ključi, ki se uporabijo med izvajanjem aplikacije. Če datoteke ni, se izvajanje zaključi, še prej pa se prikaže ustrezno obvestilo. Pomemben konfiguracijski ključ v config datoteki vsebuje „true“ ali „false“ in aplikaciji



Slika 6.1: Posnetek zaslona z aplikacijo.

pove, če uporabljani računalnik izpolnjuje vse pogoje, za uspešno izvajanje. V primeru, da ni nastavljen na „true“, se ob vsakem zagonu sproži preverjanje. Sprva se preveri, če pot vsebuje besedo „Python“, saj je nastavljena pot obvezna za izvajanje skript ne glede na njihovo lokacijo na trdem disku. V nadaljevanju se sproži izvajanje preproste Python skripte z imenom *checkComputerSettings*, ki vključuje modul Mechanize. Če ta modul ni nameščen, je izvajanje te skripte in vseh preostalih nemogoče. Za konec se skripta poizkusi povezati na spletno stran in s tem preveri, če obstaja povezava z internetom. V primeru, da se vse izvede uspešno, se v konfiguracijski ključ vpiše „true“, in ob naslednjih izvajanjih se to preverjanje ne bo pojavilo.

Naslednji ključ vsebuje pot za delovno mapo, v katerega se prednastavljeno shranjujejo rezultati in zasebna sporočila. Shranjena sta tudi podatka obstoječega računa, ki se uporabi pri shranjevanju spiska uporabnikov. Ta dva

podatka se lahko posodobita tudi med izvajanjem aplikacije, če jih uporabnik spreminja. Katerikoli element iz config datoteke je mogoče spremeniti pred izvajanjem in se ga uporabi kasneje.

6.5 Ustvarjanje spiska uporabnikov foruma

Sprva je potrebno ustvariti spisek uporabnikov. Zaradi forumove zaščite je potrebno uporabiti obstoječ račun, katerega se določi ali v config datoteki ali pa preko grafičnega vmesnika. Pri slednji izbiri se prikaže nova forma z dvema besedilnima poljema. Namenjena sta vnosu uporabniškega imena ter pripadajočega gesla. Mogoče pa je tudi onemogočiti uporabo računa s klikom na potrditveno stikalo. Sama forma vsebuje tri lastnosti objektov, preko katerih je mogoče iz glavne forme pridobiti stanje potrditvenega stikala in vnesenega teksta. Spremembe se shranijo samo ob kliku na potrditveni gumb, drugače pa se forma zapre.

Ob kliku na gumb se odpre pogovorno okno za shranjevanje, preko katerega uporabnik določi pot za shranjevanje spiska. Sledi klic skripte *downloadFirstPage* skupaj s potrebnimi argumenti, med katere spadajo uporabniško ime in geslo, povezava ter pot za shranjevanje prve strani, ki je v obliki HTML. Skripta odpre sprejeto povezavo, vnese uporabniške podatke, dokonča prijavo in je preusmerjena na prvo stran spiska uporabnikov foruma. Ta se shrani v začasno mapo operacijskega sistema in čaka na nadaljnjo obdelavo. V primeru, da se je prijava zgodila uspešno, se prva stran odpre in iz nje razbere število strani. Ta številka pove, na koliko straneh se nahaja celoten spisek uporabnikov. Zatem se naslednja skripta, *downloadMemberlist*, poveže na vsako stran posebej. Uporabi se for zanka, ki je navzgor omejena s prej pridobljenim številom strani. V zanki se sproti gradijo povezave, ki pripeljejo do naslednje strani. Vsaka tako zgrajena povezava se, po enakem principu kot v prejšnji skripti, odpre s pomočjo modula Mechanize in shrani na disk. Ob zaključitvi tega procesa je potrebno obdelati nove datoteke.

Vsako datoteko je potrebno odpreti in brati po vrsticah. Pri tem se išče določena beseda, ki se pojavi samo pred uporabniškim imenom. Iz vrstice, ki vsebuje to besedo, se izlušči uporabniško ime ter ga zapiše v spisek na disku. Ko pride zanka do zadnje HTML datoteke se shranijo zadnja uporabniška imena, aplikacija počisti za seboj in odstrani ustvarjene datoteke. Uporabnik tako dobi tekstovni dokument vseh trenutnih uporabnikov foruma, ki so med seboj ločeni z novimi vrsticami. Takšen spisek je mogoče posodobiti kadarkoli, saj so registracije novih članov dokaj pogoste.

6.6 Shranjevanje zasebnih sporočil

Pred začetkom mora uporabnik vnesti v besedilna polja podatke za prijavo. Po kliku na gumb za shranjevanje se prične izvajanje shranjevanja zasebnih sporočil ter poštnega naslova uporabnika. Če sta bila v besedilnima poljema vnesena podatka, se pokliče skripta *savePrivateMessage*. Poleg teh dveh vrednosti se, kot argumenta, skripti posreduje še pot do delavne mape, kamor se shranijo sporočila ter poštni naslov. Datoteke se shranijo v pod mapo, poimenovano po uporabniku, ki bo prijavljen. Na ta način se izogne zmešnjavi ob večkratnih izvajanjih z različnimi uporabniki.

Skripta deluje tako, da odpre povezavo, preko katere se prijavi v forum. Modul Mechanize omogoča hrambo piškotkov. To omogoča, da ostane račun prijavljen in lahko dostopa do svojih zasebnih sporočil. Povezava do nabiralnika je znana in skripta shrani spisek prikazanih sporočil v HTML datoteko. Nato se shrani še stran, ki med drugim vsebuje tudi poštni naslov, na katerega je registriran prijavljen račun. Sledi branje in obdelava novo ustvarjenih datotek. Sprva se prebere datoteko, ki vsebuje poštni naslov. V vsaki vrstici se išče prisotnost besed „name="email" “, ki se pojavi pred poštnim naslovom. Ko je vrstica najdena, je potrebno samo še izluščiti naslov ter ga shraniti na ustrezno mesto. Vsako zasebno sporočilo ima svojo identifikacijsko številko. Takšno številko se uporabi pri sestavljanju povezave do posameznega sporočila. Uporabi se namreč začetni del nespremenljive povezave do zasebnega sporočila, na konec pa se doda identifikacijsko številko. Na ta način se ustvari toliko povezav, kolikor je zasebnih sporočil. Vsako povezavo se odpre in shrani v delovno mapo.

6.7 Iskanje uporabnikov po geslu

Uporabnik sprva s pomočjo pogovornega okna določi pot do tekstovne datoteke, ki vsebuje uporabniška imena. Ta so v spisku med seboj ločena z novimi vrsticami. Nad temi imeni se vrši napad z geslom, ki je vpisan v ustrezno besedilno polje. Mogoče pa je tudi nastaviti, da se za geslo vnese uporabniško ime trenutnega uporabnika. To možnost se izbere s klikom na potrditveno stikalo. Zatem je mogoče napad sprožiti. V primeru, da spisek ni določen, se pred nadaljevanjem prikaže pogovorno okno, preko katerega uporabnik nastavi pot. Aplikacija ustvari tekstovno datoteko, v katero se zapisujejo morebitni zadetki, nato pa se napad prične.

Glavni del je obdan z zanko, ki iz spiska bere eno uporabniško ime za drugim. Uporabi se skripta *findUsers*, ki sprejme uporabniško ime ter geslo

kot argumente (priloga A.2). Med izvajanjem odpre povezavo, preko katere se uporabniki foruma prijavijo. Izbere ustrezni formi in vnese prejeto uporabniško ime in geslo. Po potrditvi vnešenega se rezultat shrani v ločeno spremenljivko. To spremenljivko se pregleda in če ta vsebuje besedo „napaka“ ali „limit“, prijavljanje ni uspešno. Beseda „napaka“ se namreč prikaže v primeru, če je vneseno geslo napačno, „limit“ pa takrat, ko se vklopi CAPTCHA. V teh dveh primerih skripta izpiše vrednost „0“, v nasprotnem primeru pa „1“. Izpise se ujame v aplikaciji, ki je klicala skripto. Glede na izpisane vrednosti v skripti se v seznamsko polje vnese obvestilo z ustreznim besedilom. Uspešno najdene račune se skupaj z geslom shrani v tekstovno datoteko (slika 6.2). Nato se vpisovanje v tekstovno datoteko zaključi in zapre.

6.8 Iskanje gesel po uporabniku

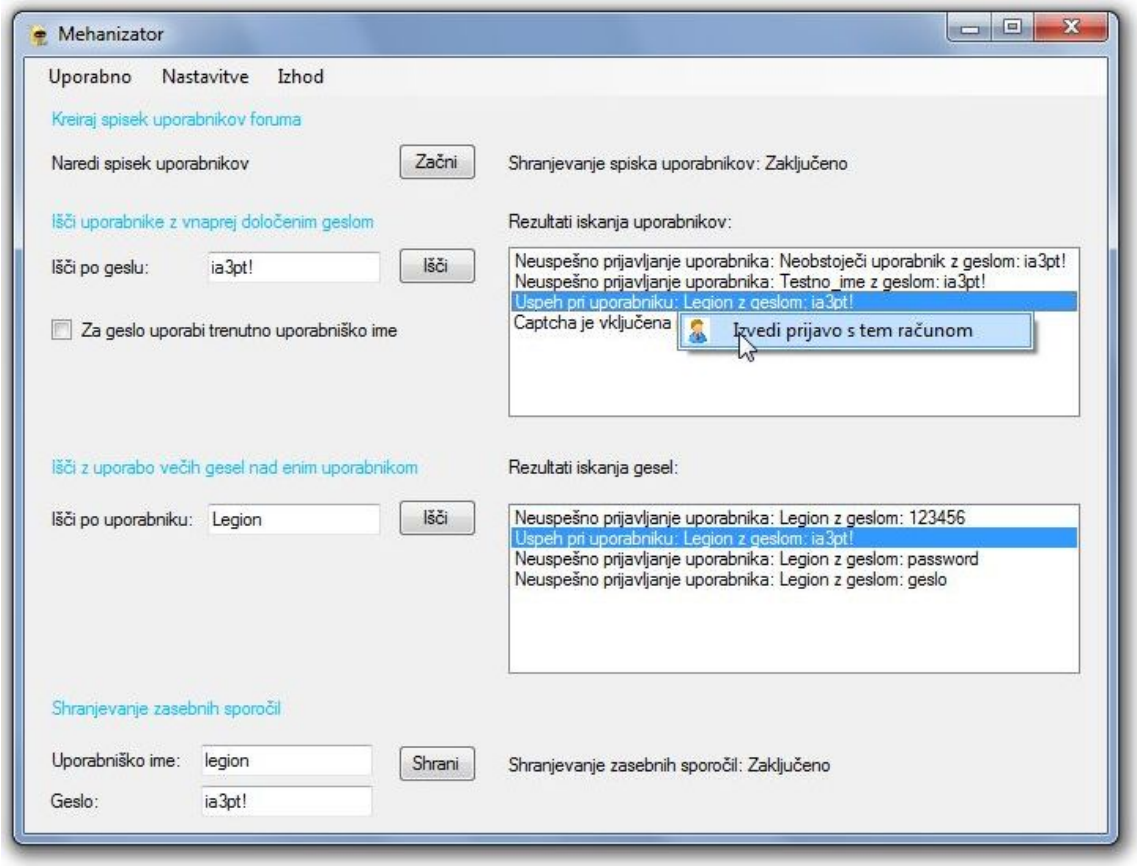
Namen te funkcije je enak kot pri iskanju uporabnikov po geslu, vendar z manjšimi razlikami. Uporablja se spisek gesel, ki so med seboj ločena z novimi vrsticami. Uporabnik določi pot do takšnega spiska in vnese uporabniško ime v besedilno polje. Po kliku na gumb se izvrši preverjanje prisotnosti poti do spiska gesel. Hkrati se preveri če je besedilno polje izpolnjeno. Sledi izvajanje skripte *findUsers*. Glavna razlika je torej pri načinu iskanja, vse ostalo je zelo podobno zgoraj opisani funkciji. Rezultati se prav tako zapisujejo v seznamsko polje, zadetki pa se shranijo v tekstovni dokument.

6.9 Slabosti aplikacije

Manjša slabost je odvisnost od modula Mechanize. Uporaba aplikacije brez namestitve Pythona, modula Mechanize in nastavljanja spremenljivke okolja ni mogoča. To bi od uporabe odvrnilo marsikaterega uporabnika, če bi bila aplikacija javno dostopna. Manj vešči uporabniki bi verjetno imeli nekaj težav pri nameščanju modula, če se nebi seznanili z navodili.

Ob vključitvi varnostnega mehanizma CAPTCHA je aplikacija nemočna. V kodo ni vključena nobena rešitev za prepoznavanje besedila. Prva uporabljena CAPTCHA je bila preprosta, vendar je bila nadomeščena z drugo po navalu avtomatizirano registriranih uporabnikov. Naslednja verzija je takšne težave preprečila in je torej trd oreh za samodejno reševanje.

Največja negativna lastnost leži v neprenosljivosti aplikacije na druge forume. Aplikacija je prirejena za izvajanje samo na forumu Fri-Info. Razlogov za neprenosljivost je več, najbolj očiten je spletni naslov foruma. Za uporabo



Slika 6.2: Posnetek zaslona med izvajanjem aplikacije.

na drugih spletnih forumih, bi bilo potrebno menjavati spletni naslov. Vendar pa se izkaže, da je to eden najmanjših problemov in do določene mere rešljiv. Največja težava leži v različnih tipih forumov. Moja aplikacija je prirejena za uporabo na forumu tipa phpBB, vendar pa so v uporabi prav tako druge vrste. Zelo so popularni tudi drugi tipi, kot je na primer IPB, vBulletin in še mnogo drugih. Različni tipi forumov pa pomenijo različne vsebine v HTML datotekah.

Aplikacija in določene skripte po takšnih datotekah iščejo točno določene besede. Lep primer je funkcija za ustvarjanje spiska uporabnikov. Ko so ustvarjene datoteke HTML (vsaka izmed njih vsebuje uporabnike foruma), je potrebno iz njih pridobiti posameznega uporabnika. To poteka tako, da se išče vrstico, ki vsebuje besedo „viewprofile“. Podoben postopek se uporablja pri preostalih skriptah, razlikujejo se samo iskane besede. Za uspešno uporabo

na različnih tipih forumov bi bilo potrebno uporabljati točno take besede, kot se uporabljajo na želenem forumu. Prav tako bi bilo potrebno spremeniti način shranjevanja HTML datotek, v katerih so zapisani uporabniki. Za vsako takšno datoteko je potrebno zgraditi povezavo. Gradnja teh povezav pa seveda ni enaka na vseh tipih forumov.

Naslednja težava so imena form. Vse skripte uporabljene v aplikaciji, z izjemo tiste, ki preveri računalniške nastavitve, izvajajo prijavo na forum. Da se prijava uspešno zgodi, je potrebno poznati ustrezni imeni form – za uporabniško ime in za geslo. Ta dva podatka pa se dobi z vpogledom v izvor kode strani in se razlikujeta pri različnih verzijah foruma. Vse skupaj pa bi se lahko še dodatno zakompliciralo z nameščanjem morebitnih modifikacij, ki omogočajo lastno poimenovanje form.

6.10 Morebitne izboljšave

Če bi želel aplikacijo uporabljati z namenom pridobivanja čim več gesel in resnih zlorab, bi dodal še nekaj izboljšav in vložil nekaj denarja. Aplikacija je že sedaj popolnoma funkcionalna, vendar nima rešitve pred varnostnim sistemom CAPTCHA. Najbolj učinkovita rešitev bi bila uporaba plačljivih storitev za človeško reševanje takšnih testov. V aplikacijo bi bilo potrebno samo implementirati GUI, ki je dostopen na osebni strani združbe, ki ponuja takšne storitve. Na ta način se izogne tudi morebitnim spremembam prikazovanja tega varnostnega mehanizma.

Dobrodošla bi bila tudi izboljšava prenosljivosti na druge tipe forumov. Popolne prenosljivosti ne bi dosegel, lahko pa bi poskrbel, da aplikacija opravlja svoje delo na treh najbolj pogostih tipih forumov. Takšni dodatki bi zmanjšali preprostost uporabe in uporabniku naložili dodatno delo. Za začetek bi moral vnesti vse potrebne povezave do želenega foruma. Nato bi bilo potrebno na teh povezavah pregledati še izvorno kodo strani in poiskati določene vrednosti. S tem je mišljeno na primer iskanje ustrezne vrstice, ki vsebuje zaporedje znakov, ki naznanja prisotnost uporabniškega imena. Tako najdene nujno potrebne podatke bi potem bilo potrebno vnesti ali v aplikacijo ali pa v config datoteko.

Uporabno bi bilo dodati še izboljšavo iz področja varnosti. Do sedaj ni bila vključena iz preprostega razloga – ni bilo potrebno skrivati spletne identitete. V grafičnemu vmesniku bi se omogočil vnos proxy naslovov, te pa se bi v skriptah nastavilo z metodo set_proxies. Pri brskanju po forumu preko aplikacije bi za to poskrbel razred WebProxy.

Za hitrejše izvajanje bi bilo koristno skrajšati spisek uporabnikov. Med

ustvarjanjem spiska bi se izključilo določen tip uporabnikov. Ob predpostavki, da so administratorji ter moderatorji osveščeni o uporabi močnih gesel, se takšnih računov ne bi dodalo na spisek. Najbolj opazna razlika med ekipo, ki skrbi za delovanje foruma, in preostalimi uporabniki je uporaba forumske slike. Drugi uporabniki si slike namreč ne morejo nastaviti. Preko te podrobnosti je tako zlahka narediti selekcijo. Izvajanje se bi lahko še dodatno pohitrilo, če bi aplikacija hkrati tekla na večih računalnikih. Vsak računalnik bi dobil svoj del slovarja in breme se bi porazdelilo na več delov.

Tudi za slovar je možnih nekaj dodatnih izboljšav. Za resne napade bi bilo potrebno najprej močno razširiti vsebino slovarja. Pametno bi bilo vključiti na primer celoten spisek najpogostejše uporabljanih gesel na spletu. Dobrodošel dodatek pa bi bila še gesla, spremenjena s pomočjo mutacijskih filtrov. Takšne spremembe obstoječega slovarja bi sicer povzročile daljše izvajanje napadov, vendar pa bi prinesle več najdenih uporabniških računov s preprostimi gesli.

Poglavje 7

Načini varne uporabe aplikacije in pridobljenih računov

Slovenska zakonodaja prepoveduje neupravičene vstope ali vdore v informacijske sisteme. Zato bi bilo potrebno ob izvajanju aplikacije uporabiti kakšnega izmed večih načinov za prikrivanje spletne identitete. V kazenskem zakoniku republike Slovenije 221. člen[18] govori o napadu na informacijski sistem:

(1) Kdor vdre v informacijski sistem ali kdor neupravičeno prestreže podatek ob nejavnem prenosu v informacijski sistem ali iz njega, se kaznuje z zaporom do enega leta.

Vdor je mišljen kot vstop v informacijski sistem na takšen način, da se izogne oviri v obliki varnostnega mehanizma. V primeru povzročitve velike škode, je zagrožena zaporna kazen od treh mesecev pa do petih let.

Ob javni objavi kode moje aplikacije in navodil za uporabo, bi prav tako kršil zakon. O tem govori 306. člen - Izdelovanje in pridobivanje orožja in pripomočkov, namenjenih za kaznivo dejanje[18]. V tretjem odstavku je zapisano:

(3) Enako kot v prejšnjem odstavku se kaznuje, kdor z namenom storitve kaznivega dejanja poseduje, izdeluje, prodaja, daje v uporabo, uvaža, izvaža ali kako drugače zagotavlja pripomočke za vdor ali neupravičen vstop v informacijski sistem.

V izognitev morebitnim težavam z zakonom je smotrno uporabiti enega izmed naslednjih varnostnih ukrepov.

7.1 Uporaba javnih računalnikov

Dandanes ima skoraj vsak večji nakupovalni center svoj kotiček, v katerem so prosto dostopni računalniki. Te ponavadi uporabljajo širokopasovne povezave in niso pod strogim nadzorom. Če skrbnik ni poskrbel za onemogočanje nameščenja programske opreme, ni skoraj nobene ovire več. Napadalec lahko namesti vse potrebno in prične z izvajanjem aplikacije.

7.2 Proxy strežnik in TOR omrežje

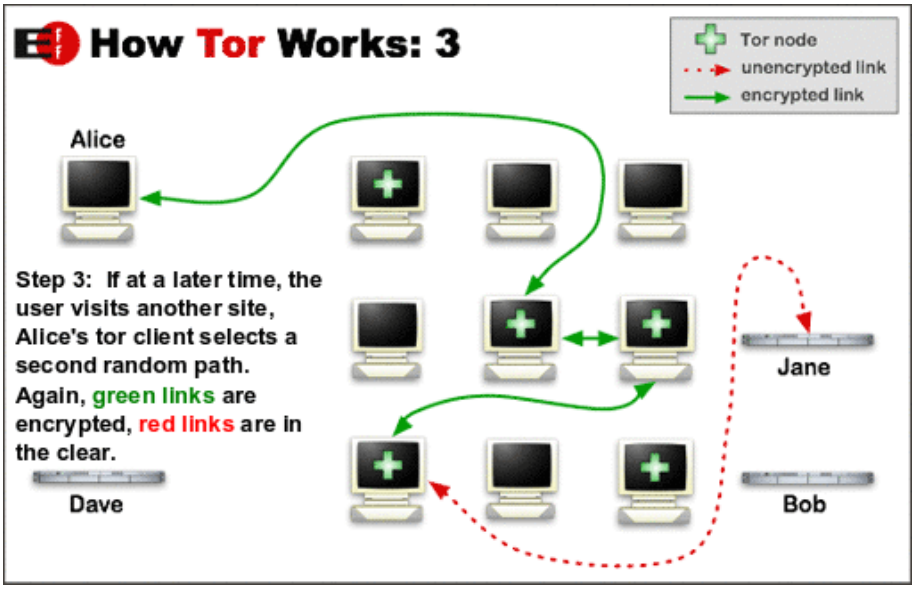
Takšen strežnik služi uporabniku kot posrednik, ki izvaja zahteve v njegovem imenu in mu zakrije identiteto. Deluje torej kot vmesni člen med uporabnikom in preostalimi strežniki, ki jih želi uporabnik obiskati. Na obiskanih straneh pa je prikazan IP naslov proxy strežnika. Splet ponuja veliko brezplačnih proxy strežnikov. Slabost le-teh je njihova slaba odzivnost zaradi velikega števila ljudi, ki ga uporablja.

Implementacija proxy strežnika v skripte ni zapletena. Modul Mechanize namreč omogoča preprosto dodajanje takšnega varnostnega mehanizma. To se naredi s pomočjo metode set_proxies, ki za argumente sprejme vrsto protokola ter IP naslov za proxy. Tako je proxy nastavljen in se bo uporabljal pri nadaljnjemu brskanju po spletu.

Za navadno brskanje se z lahkoto uporabi anonimno proxy stran ali pa še bolj zanesljivo TOR omrežje. Slednji sistem pogosto uporabljajo etični uporniki pri anonimnem razkrivanju tajnih podatkov novinarjem. Sistem ne deluje po principu sledenja neposredne poti od izvora do želenega cilja. Namesto tega aplikacija gradi zaporedje kriptiranih povezav med računalniki v omrežju (slika 7.1). Tako posamezen računalnik pozna samo izvor in ponor podatkov, nikoli pa mu ni znana celotna pot. Za dodatno varnost, se iste povezave uporabljajo le približno deset minut. Naslednje zahteve se izvedejo preko novih povezav.

7.3 Nezaščitena brezžična omrežja

Uporaba brezžične opreme je pri nas dokaj razširjena. Poleg zasebnih brezžičnih omrežij so na razpolago še javna, ki so dostopna vsakomur. Takšna odprta omrežja ponujajo različne spletne kavarne, restavracije in tudi mestna občina Maribor. Ponavadi nimajo časovnih omejitev, morebitne težave lahko povzroča le šibkost signala. Dodatne možnosti se odprejo še pri zlorabah zasebnih brezžičnih omrežij.



Slika 7.1: Zadnji korak primera uporabe TOR omrežja.

Revija Moj mikro je že drugič ponovila analizo o zaščitenosti brezžičnih omrežij v Ljubljani[10]. Za tako imenovani „war driving“ so se posluževali prenosnika in vgrajene brezžične antene. Samo zaznavanje brezžičnih omrežij ni protizakonito. Vzrok za takšno odločitev je potrebno pripisati dejstvu, da se lahko računalnik samodejno poveže na nezaščiteno omrežje. Pri nas lastniki takih omrežij nimajo težav, v Nemčiji pa je lastnik lahko denarno kaznovan[4].

Prvič so preverjali število nezaščitenih omrežij v letu 2004. Takrat so zasledili 587 brezžičnih dostopnih točk, od katerih jih je bilo 276 nešifriranih. V šestih letih se je število dostopnih točk namnožilo na 5963. Nezaščitenih omrežij v lanskem letu je bilo 848, torej **14,22%**. Več kot dovolj velika izbira za ilegalno početje. Določena zaščitena omrežja pa prav tako ne povzročajo preveč težav. Verjetnost uspešnega dostopa v zaklenjeno omrežje je odvisen od načina šifriranja brezžičnih povezav. Tako je na primer šifriranje WEP mogoče razbiti v času 10 minut. Takšno početje pa že spada med kazniva dejanja.

7.4 Najboljša izbira v mojem primeru

Za anonimno poganjanje moje aplikacije bi bila najbolj preprosta in učinkovita možnost uporabe nezaščitenega brezžičnega omrežja. Izbira javnega računalnika ali proxy strežnika ni izključljiva, vendar je bolj tvegana od uporabe nezaščitenega

omrežja. Slabost javnih računalnikov je v temu, da mora biti uporabnik ves čas fizično prisoten. Pri proxyjih pa obstaja možnost, da so odgovorni prisiljeni v razkritje svojih uporabnikov, ob predložitvi sodnega naloga. Poleg tega pa bi si za hitro izvajanje aplikacije moral omisliti plačljivi proxy.

Največ dobrih strani ima torej tretja opcija. Odprtih brezžičnih omrežij je na izbiro dovolj in jih ni težko najti. Če se uspe najti kakšnega v predelu mesta, ki je pokrito z optičnim omrežjem, je poskrbljeno še za hitrost. Ob morebitnih težavah pa je malo verjetno, da se odkrije pravega storilca. Na omrežje se poveže z lastnim prenosnikom, ki ima že nameščeno obvezno programsko opremo, tako da je potrebno samo še sprožiti izvajanje aplikacije.

Poglavje 8

Sklepne ugotovitve

Diplomsko delo prikazuje pomembnost izbire močnih gesel in nevarnosti, ki pretijo preostalim geslom. Rezultat dela je funkcionalna aplikacija, ki je zmožna izbrskati uporabnike foruma Fri-Info s preprostimi gesli. Rezultati najdenih gesel so zelo podobni tujim analizam, narejenim na večjem krogu ljudi. Na forumu je eno izmed najbolj pogostih gesel „123456“ in ogromno ljudi si za geslo izbere kar svoje uporabniško ime. Razlogov za takšne izbire gesel je veliko, od lenobe pa vse do nezavedanja nevarnosti in računalniške nevednosti.

Kot je razvidno iz prejšnjih razdelkov diplomske naloge, je mogoče brez večjih težav narediti preprost bot. Njegove zmogljivosti so v največji meri omejene z varnostnimi mehanizmi, uporabljanimi na spletnih straneh. Tako kot bo naraščalo število botov, iskalnih programov in ostalih avtomatiziranih procesov, se bodo v prihodnosti pojavljali tudi novi varnostni mehanizmi. Takšne zaščite pa bodo nadobudni nepridipravi ves čas poizkušali zaobiti.

Uporabo preprostih gesel bo mogoče počasi zmanjšati z osveščanjem ljudi o takšnih nevarnostih. Še vedno pa se bo veliko število ljudi srečalo s takšnimi nevarnostmi po težji poti. Najboljša rešitev za ljudi, ki neprestano uporabljajo preprosta gesla, je uporaba programske opreme, ki hrani vsa gesla. Primer takšne aplikacije je LastPass.

Ta aplikacija hrani vsa gesla, ki jih ima uporabnik, tako da mu jih ni potrebno vsakič vnašati. Do njih lahko dostopa iz kateregakoli računalnika, samo če pred tem v brskalnik namesti takšen dodatek. Vendar pa je pred prvo uporabo potrebno najprej vpisati glavno geslo. Takšno glavno geslo mora biti izredno močno in je pravzaprav edino geslo, ki ga mora uporabnik poznati. Preostala gesla, kot je na primer geslo za dostop do foruma ali pošte, pa so shranjena na strežniku. Da so ta gesla res močna, poskrbi generator močnih gesel, ki je vključen v takšno aplikacijo. Za dodatno varnost se vsa

gesla šifrirajo na uporabnikovem računalniku, preden se shranijo na strežniku. Tako ima lahko do njih dostop res le njihov lastnik, ki pozna glavno geslo. Preostala gesla so tako dovolj kompleksna, da izpolnjujejo zahteve o močnih geslih, uporabnikom pa jih ni potrebno poznati na pamet.

Dodatek A

Priloge

A.1 Klicanje skript iz C#

```
Process loginScript = new Process();
loginScript.StartInfo.FileName = ("python.exe");
loginScript.StartInfo.Arguments = (String.Format("{0} {1} {2}",scriptPath,user,pwd));
loginScript.StartInfo.CreateNoWindow = true;
loginScript.StartInfo.UseShellExecute = false;
loginScript.StartInfo.RedirectStandardOutput = true;
loginScript.StartInfo.RedirectStandardError = true;
loginScript.Start();
```

A.2 Osnovni del skripte za prijavo na forum

```
browser = mechanize.Browser()
browser.open("http://fri-info.net/forum/ucp.php?mode=login")
browser.select_form(nr=1)
browser.form['username'] = receivedUsername
browser.form['password'] = receivedPassword
response=browser.submit().read()
```

Slike

Tabele

Literatura

[1] L. von Ahn,"Teaching computers to read: Google acquires reCAPTCHA,". Dostopno na: http://googleblog.blogspot.com/2009/09/teaching-computers-to-read-google.html

[2] B. Calin, "Statistics from 10,000 leaked Hotmail passwords,". Dostopno na: http://www.acunetix.com/blog/news/statistics-from-10000-leaked-hotmail-passwords

[3] C. Houck, "Decoding reCAPTCHA Paper,". Dostopno na: http://n3on.org/projects/reCAPTCHA/docs/reCAPTCHA.docx

[4] K. Grieshaber, "German court orders wireless passwords for all,". Dostopno na: http://www.msnbc.msn.com/id/37107291/ns/technology_and_science-security/

[5] S.P. Corell, "A new way of social engineering,". Dostopno na: http://pandalabs.pandasecurity.com/a-new-way-of-social-engineering

[6] B. H, "Bitka na Žurnalovem polju,". Dostopno na: http://www.zurnal24.si/slovenija/bitka-na-zurnalovem-polju-108389/clanek

[7] J. J. Lee, "Mechanize,". Dostopno na: http://wwwsearch.sourceforge.net/mechanize/

[8] E.Kalan, "Vaša identiteta JE pomembna in je samo VAŠA!,". Dostopno na: http://www.dnevnik.si/objektiv/1042248041

[9] A. Lukić,“ Banka Slovenije razkrila resnico,”. Dostopno na: http://zurnal24.si/gospodarstvo/banka-slovenije-razkrila-resnico-183701/clanek

[10] J. Mele, “Ljubljana, govori z menoj!,”. Dostopno na: http://www.mojmikro.si/v_srediscu/podrobneje_o/ljubljana_govori_z_menoj

[11] Microsoft research, “Asirra,”. Dostopno na: http://research.microsoft.com/en-us/um/redmond/projects/asirra

[12] P. Oechslin, “Making a Faster Cryptanalytic Time-Memory Trade-Off,”. Dostopno na: http://lasecwww.epfl.ch/~oechslin/publications/crypto03.pdf

[13] R. Ordoñez, “CAPTCHA Wish Your Girlfriend Was Hot Like Me?,”. Dostopno na: http://about-threats.trendmicro.com/ArchiveMalware.aspx?language=us&name=TROJ_CAPTCHAR.A

[14] “reCAPTCHA: Human-Based Character Recognition via Web Security Measures,”. Dostopno na: http://www.google.com/recaptcha/static/reCAPTCHA_Science.pdf

[15] Tor project, uradna stran. Dostopno na: http://www.torproject.org/about/overview.html

[16] Uradna stran podjetja za prepisovanje CAPTCHA. Dostopno na: http://bypasscaptcha.com

[17] Uradna stran podjetja za prepisovanje CAPTCHA. Dostopno na: http://www.decaptcher.com/client

[18] “UKAZ o razglasitvi Kazenskega zakonika (KZ-1),“. Dostopno na: http://www.uradni-list.si/1/objava.jsp?urlid=200855&stevilka=2296

[19] “What is a Captcha”. Dostopno na: http://www.google.com/recaptcha/captcha