CARNet
HRVATSKA AKADEMSKA I ISTRAŽIVAČKA MREŽA
CROATIAN ACADEMIC AND RESEARCH NETWORK

Metode za poboljšanje sigurnosti web preglednika

CCERT-PUBDOC-2009-09-276

+CERT.hr u suradnji s LSS

Sigurnosni problemi u računalnim programima i operativnim sustavima područje je na kojem CARNet CERT kontinuirano radi.

Rezultat toga rada je i ovaj dokument, koji je nastao suradnjom CARNet CERT–a i LS&S–a, a za koji se nadamo se da će Vam koristiti u poboljšanju sigurnosti Vašeg sustava.

CARNet CERT, www.cert.hr

Nacionalno središte za **sigurnost računalnih mreža** i sustava.

LS&S, www.LSS.hr

Laboratorij za sustave i signale pri Zavodu za elektroničke sustave i obradbu informacija Fakulteta elektrotehnike i računarstva Sveučilišta u Zagrebu.

Sadržaj

1. Uvod

Web preglednik je programski paket namijenjen dohvaćanju i prezentaciji web stranica, odnosno slikovnih datoteka, video zapisa ili neke druge vrste sadržaja koji se na njoj nalazi. Osim toga, može se koristiti i za pristup informacijama s web poslužitelja u privatnim mrežama ili datotekama u datotečnim sustavima. Prema statističkim podacima, proteklih se godina najviše korisnika opredijelilo za sljedeće preglednike: Internet Explorer, Mozilla Firefox, Apple Safari, Google Chrome i Opera.

Većina korisnika svakodnevno koristi web preglednike, što ih čini posebno pogodnima za izvršavanje napada na sigurnost računala. U ovom će dokumentu biti dan pregled najčešćih ranjivosti web aplikacija koje se često mogu iskoristiti putem preglednika te usporedba najčešće korištenih preglednika (Internet Explorer i Mozilla Firefox), kao i opis sigurnosnih metoda ugrađenih u njihove najnovije inačice. Osim toga, biti će riječi i o dvije najpoznatije metode za poboljšanje sigurnosti web preglednika – tzv. "*sandbox*" metodi i metodi virtualizacije aplikacije, kao i o popularnim alatima za njihovo provođenje – *Sandboxie* i *Thinapp*. Na kraju će biti dan i pregled web preglednika u razvoju, od kojih se očekuje da će korisnicima osigurati jednostavnije i sigurnije surfanje Internetom.

2. Sigurnost web preglednika

Postoje dva najčešća načina iskorištavanja ranjivosti web preglednika – proučavanje pregledničkih datoteka i navođenje korisnika na postupke koji ga dovode u kompromitirajuću situaciju.. Obje metode najčešće rezultiraju otkrivanjem povjerljivih korisničkih podataka.

Informacije o korisniku moguće je otkriti pomoću nekoliko pregledničkih datoteka – datoteka iz priručne memorije (eng. *cache files*), datoteke s poviješću surfanja (eng. *history file*) i tzv. *bookmark* oznaka. U tzv. *history* datoteci čuvaju se poveznice koje je korisnik posjetio (kod većine preglednika pamte se podaci za prethodnih 30 dana), što napadaču daje informacije o korisničkim interesima. *Bookmarks* oznake predstavljaju sličan problem jer otkrivaju napadaču koje stranice korisnik najčešće posjećuje. Priručna memorija omogućava brže pristupanje informacijama kojima je korisnik nedavno pristupio. Budući da se pregledom neke web stranice ona sprema lokalno, kod sljedećeg pristupa preglednik ne mora dohvaćati podatke preko mreže. Na taj način *Bookmarks* oznake i priručna memorija predstavljaju sigurnosnu prijetnju - ako je označena neka stranica koja zahtjeva korisničko ime i zaporku za pristup, napadač najčešće može uvidom u priručnu memoriju otkriti te zaporke ili barem korisnička imena.

Navođenje korisnika na postupke koji ga dovode u kompromitirajuću situaciju najčešće se svodi na preusmjeravanje prema zlonamjerno oblikovanoj web stranici na kojoj tada korisnik ostavlja svoje osobne podatke. Npr. korisnik se nalazi na web stranici banke u kojoj ima otvorene račune. Želi pregledati stanje svojih računa, ali klikom na gumb koji ga je trebao preusmjeriti na stranicu s računima zapravo je preusmjeren na stranicu sličnog izgleda, ali pod kontrolom napadača.

Ranjivosti uočene u popularnim preglednicima se svakodnevno uklanjaju, ali isto tako hakeri svakodnevno otkrivaju nove sigurnosne rupe.

U nastavku će biti opisano 5 najčešćih ranjivosti web aplikacija. Web aplikacija uglavnom sadrži skripte pomoću kojih komunicira s krajnjim korisnikom. Možemo reći da se sastoji od tri komponente:

- web poslužitelj šalje stranice korisničkom pregledniku,
- aplikacijski poslužitelj obrađuje te podatke i
- potrebni se podaci spremaju u bazu podataka.

Budući da web poslužitelj potrebne stranice šalje pregledniku, napadač može iskoristiti preglednik kao sredstvo napada.

2.1. Trendovi ranjivosti

Postojanje ranjivosti u web preglednicima predstavlja sigurnosni rizik za korisnika. Poznato je da hakeri često dolaze do novih metoda napada, ali kad korisnički preglednik ne bi sadržavao ranjivosti koje je moguće iskoristiti, tada bi sigurnosni rizik za korisnika prilikom surfanja bio znatno manji. Dodatno, mnoge web stranice posjeduju XSS (eng. *cross-site scripting*) ranjivosti. Trenutno se najčešće provode sljedeći napadi na web aplikacije:

- umetanje proizvoljnog SQL koda,
- pokretanje proizvoljnog programskog koda,
- iskorištavanje grešaka u formatiranju znakovnih nizova,
- XSS i
- pogađanje korisničkih imena.

2.1.1. Umetanje proizvoljnog SQL koda (eng. *SQL injection*)

Umetanje proizvoljnog SQL koda je tehnika izvođenja napada na web aplikacije koje konstruiraju SQL upite iz korisnički unesenih podataka (najčešće korisničkog imena i zaporke). Takvi se podaci najčešće unose preko korisničkog web preglednika. Problem se javlja kad aplikacija ne obradi korisnički unesene podatke na ispravan način, čime napadaču omogućava izmjenu konstrukcije pozadinskog SQL upita. Tako izmijenjen SQL upit tada se može pokrenuti s ovlastima komponente koja je pokrenula SQL naredbu, a to može biti poslužitelj baze podataka, web poslužitelj i sl. Posljedice koje uspješno izveden napad može imati jesu krađa informacija, preuzimanje kontrole nad bazom podataka i izvršavanje proizvoljnih naredbi u sustavu.

Primjerice, web aplikacija može za autentikaciju koristiti sljedeći SQL kod:

```
String SQLQuery = “SELECT Username FROM Users WHERE Username = ‘”
+
username + “’ AND Password = ‘” + password + “’”;
```

Time se korisnički uneseni podaci direktno ubacuju u SQL upit pomoću varijabli *username* i *password*. Ako napadač umjesto korisničkog imena upiše niz: `' OR ''='`, a umjesto zaporke niz: `' OR ''='`, tada će SQL upit izgledati ovako:

```
SELECT Username FROM Users WHERE Username = '' OR''=''
AND Password = '' OR ''=''
```

Na taj način, umjesto uspoređivanja korisničkih podataka, uspoređivat će se prazan niz s praznim nizom, rezultat čega je uvijek istinit pa će se napadač moći prijaviti na sustav kao prvi korisnik u korisničkoj tablici *Users.*

2.1.2. Pokretanje proizvoljnog programskog koda

Kao što i sam naziv kaže, riječ je o ranjivosti koja napadaču omogućava pokretanje proizvoljnog koda na ranjivom poslužitelju i dohvaćanje željenih informacija koje se na njemu nalaze. Do ovakvih ranjivosti najčešće dolazi uslijed pogrešaka nastalih u kodiranju, koje je teško otkriti metodama penetracijskog testiranja. Pokretanje proizvoljnog programskog koda najčešće se postiže preuzimanjem kontrole nad pokazivačem na sljedeću naredbu (eng. *instruction pointer*). Tako se napad uglavnom svodi na umetanje proizvoljnog koda te iskorištavanje ranjivosti aplikacije kako bi se izmijenila vrijednost pokazivača na sljedeću instrukciju tako da pokazuje na umetnute naredbe (umjesto originalnih). Osim toga, pokretanje proizvoljnog koda često je posljedica prepisivanja spremnika (eng. *buffer overflow*). Do prepisivanja spremnika može doći ako web aplikacija očekuje određenu duljinu korisničkog unosa (npr. 10 znamenki telefonskog broja) pa razvojni programeri zauzmu samo onoliko memorijskih resursa koliko je potrebno za obradu tih podataka. Ako napadač umjesto očekivanih 10 znamenaka upiše npr. 100, može doći do prepisivanja spremnika, odnosno prepisivanja memorije rezervirane za izvođenje nekih drugih zadataka. Posljedice takve situacije mogu biti otkrivanje informacija o web poslužitelju kao što je: popis prethodnih transakcija, informacija o privatnim SSL ključevima, podacima iz baze podataka i sl.

Za izbjegavanje prepisivanja spremnika tipično se koriste HTML i *JavaScript* jezici za ograničavanje broja znakova koje je moguće unijeti putem preglednika. Međutim, napadač može izmijeniti HTML kod i isključiti *JavaScript* zaštitu i tada pokrenuti napad. Kako bi zaštita od ovakve vrste napada bila moguća, potrebno je dobro provjeriti sve korisnički unesene podatke

2.1.3. Greške u formatiranju

Greške u formatiranju znakovnih nizova pripadaju grupi programskih ranjivosti otkrivenih oko 1999. godine. Iako se najprije smatralo da su bezazlene, pokazano je kako mogu uzrokovati rušenje aplikacije ili pokretanje zlonamjerno oblikovanog koda [29]. Problem nastaje kad se nefiltrirani korisnički uneseni podaci predaju kao parametri određenim funkcijama za formatiranje (kao što je npr. funkcija *printf()* u programskom jeziku C). Do ovakvih pogrešaka najčešće dolazi kad programer želi ispisati znakovni niz koji sadrži korisnički unesene podatke. Ako pritom umjesto potrebne naredbe `printf("%s", spremnik)` greškom napiše `printf(spremnik)`, spremnik se interpretira kao znakovni niz i analiziraju se svi posebni formatirajući znakovi koje sadrži (kao npr. "%d"). Pomoću tih znakova moguće je doći do informacija o vrijednostima stoga programa. Time je omogućen uvid u memoriju na temelju kojeg napadač može doći do informacija korisnih za izvođenje daljnjih napada.

Napadač može iskoristiti ovakve pogreške navođenjem korisnika na zlonamjerno oblikovanu web stranicu ili podmetanjem posebno oblikovanog SSL certifikata (certifikate koji sadrže određene znakove za formatiranje). Nakon što korisnički preglednik učita takvu posebno oblikovanu stranicu ili SSL certifikat, dolazi do pogrešne interpretacije znakova za formatiranje. Tipične posljedice ovakvog napada mogu biti uskraćivanje usluga (eng. *Denial*

of Service), pristup potencijalno osjetljivim informacijama (za koje napadač nema ovlasti) i prepisivanje pokazivača na sljedeću instrukciju (te tako omogućavanje pokretanja proizvoljnog programskog koda).

2.1.4. XSS (eng. *cross-site scripting*)

XSS je napadačka tehnika koja se temelji na prosljeđivanju zlonamjerno oblikovanog izvršnog koda korisniku. Taj se kod učitava u korisnikov web preglednik gdje se i izvršava, a najčešće je pisan skriptnim jezikom *JavaScript*. Korištenjem ove napadačke tehnike napadaču je omogućeno čitanje, izmjena i prosljeđivanje osjetljivih podataka koji su na raspolaganju korisničkom web pregledniku, što dalje omogućava krađu korisničkih računa, prosljeđivanje štetnog sadržaja, usmjeravanje preglednika na druge lokacije i sl. Slijedi primjer koda ranjivog na XSS napade:

```
<form action="search.php" method="GET" />
Dobrodosli!
<p>Upisite svoje ime: <input type="text" name="ime" /><br />
<input type="submit" value="Go" /></p><br>
</form>

<?php
echo "<p>Vase ime <br />";
echo ($_GET[ime]);

?>
```

Problem je što se u ovom kodu korisnički unesena vrijednost varijable *ime* ne obrađuje prije vraćanja korisniku, što je moguće iskoristiti za pokretanje proizvoljne skripte. Neki od zlonamjerno oblikovanih kodova namijenjenih iskorištavanju ranjivosti ovog koda su sljedeći:

```
http://ranjiva_stranica/clean.php?ime=<script>kôd</script>
ili
http://ranjiva_stranica/clean.php?ime=<script>alert(document.cooki
e);</script>
```

Prvi se kod može iskoristiti za pokretanje proizvoljne posebno oblikovane skripte, dok se drugi oblik koristi za otvaranje tzv. *pop-up* prozora s ispisom korisničkog kolačića (eng. *cookie*), što omogućava krađu korisničkog identiteta.

2.1.5. Pogađanje korisničkih imena (eng. *username enumeration*)

Pogađanje korisničkih imena je tip napada kod kojeg se iskorištava činjenica da pozadinska validacijska skripta ispisuje poruke o ispravnosti unesenog korisničkog imena. Napadaču je time omogućeno eksperimentiranje s različitim korisničkim imenima i otkrivanje ispravnih pomoću različitih poruka koje ta skripta ispisuje. Ova se napadačka tehnika uglavnom koristi za otkrivanje trivijalnih korisničkih imena i zaporki kao što su kombinacije: test/test, admin/admin, gost/gost i sl. Takve račune obično stvaraju razvojni programeri prilikom testiranja aplikacije, a često ih zaborave izbrisati ili izmijeniti zaporku. Slijedi primjer poruka koja se ispisuju u web pregledniku ako je upisano korisničko ime pogrešno (ne postoji) ili korisničko ime postoji, ali je upisana pogrešna zaporka:

```
Authentication failure: entered username does not exist.
```

```
Authentication failure: incorrect password entered.
```

Zaštita od ovog tipa napada provodi se korištenjem istih poruka za različite vrste pogrešaka i zabranom korištenja trivijalnih korisničkih imena i zaporki.

2.2. Uspješni napadi

U ožujku 2009. godine održana je konferencija "CanSecWest" u sklopu koje je organizirano natjecanje "Pwn2Own". Cilj tog natjecanja bio je iskoristiti ranjivosti web preglednika za pokretanje proizvoljnog programskog koda na klijentskom sustavu. Za izvođenje napada bila su dostupna dva prijenosna računala - Sony Vaio s preglednicima Internet Explorer 8, Google Chrome i Mozilla Firefox te Macbook s preglednicima Mozilla Firefox i Safari. Jedno od pravila bilo je da se otkrivena ranjivost mora moći iskoristiti isključivo korisničkim klikom na poveznicu. Natjecanje je trajalo 3 dana, pri čemu su za svaki dan bila definirana dodatna pravila:

- prvi dan je bilo dopušteno koristiti samo podrazumijevanu konfiguracija preglednika, bez dodataka,
- drugi dan je bilo moguće korištenje dodataka (*Flash, Java, .Net, Quicktime*) i
- treći dan dopušteno je i korištenje popularnih aplikacija, kao što je *Acrobat Reader*.

Od spomenutih preglednika, samo je Chrome prošao test, dok su preglednicima Internet Explorer 8, Firefox i Safari već prvi dan natjecanja otkrivene (i iskorištene) sigurnosne ranjivosti. Svi su operacijski sustavi i web preglednici nadograđeni nakon natjecanja.

3. Popularni web preglednici

Danas korisnici imaju puno opcija prilikom biranja web preglednika. Microsoft Internet Explorer polako gubi prednost koju je imao pred ostalim preglednicima, a korisnici se sve češće odlučuju na preglednike kao što su Mozilla Firefox, Google Chrome i dr. Ipak, trenutno se i dalje najviše koriste preglednici Internet Explorer i Mozilla Firefox, što je moguće vidjeti i na slici 1:

2009	IE7	IE6	IE8	Firefox	Chrome	Safari	Opera
srpanj	15.9%	14.4%	9.1%	47.9%	6.5%	3.3%	2.1%
lipanj	18.7%	14.9%	7.1%	47.3%	6.0%	3.1%	2.1%
svibanj	21.3%	14.5%	5.2%	47.7%	5.5%	3.0%	2.2%
travanj	23.2%	15.4%	3.5%	47.1%	4.9%	3.0%	2.2%
ožujak	24.9%	17.0%	1.4%	46.5%	4.2%	3.1%	2.3%
veljača	25.4%	17.4%	0.8%	46.4%	4.0%	3.0%	2.2%
siječanj	25.7%	18.5%	0.6%	45.5%	3.9%	3.0%	2.3%

Slika 1: Usporedba popularnih preglednika u 2009. godini

Izvor: Browser Statistics, w3schools.com

3.1. Mozilla Firefox

31. ožujka 2009. godine objavljeno je kako je Mozilla Firefox prvi put postao najpopularniji preglednik u Europi, pri čemu je s 35.05% korisnika preuzeo vodstvo od preglednika Internet Explorer. Radi se o besplatnom pregledniku pisanom u programskim jezicima C++, XUL, XBL i JavaScript. Podržan je na većini današnjih popularnih platformi (Microsoft Windows, Mac OS X i Linux /Unix), a dostupan je za korištenje na više od 45 jezika.

3.1.1. Prednosti

Prednosti koje se najčešće pripisuju pregledniku Firefox (u odnosu na druge preglednike) su sljedeće:

- brzina (inačica 3.5 je dva puta brža od inačice 3 te deset puta brža od inačice 2),
- brzi odgovor na sigurnosne prijetnje na Internetu (u obliku izdanih zakrpa),
- brojne mogućnosti i dodaci za personalizaciju preglednika te
- stabilnost i dobre performanse.

Sigurnosne metode koje se koriste u pregledniku Firefox su sljedeće:

- provjera informacija o web stranici prije upisivanja osobnih podataka (klikom na ikonu stranice u adresnoj traci),
- integracija s antivirusnim programom,
- privatnost prilikom pretraživanja,
- brisanje podataka o posjećenim web stranicama,
- provjera web stranica s ciljem zaštite od virusa, crva, trojanskih konja i drugih zlonamjernih programa,
- detektor tzv. *phishing* napada,
- program za upravljanje zaporkama i dr.

3.1.2. Nedostaci

Jedino što korisnici zamjeraju pregledniku Mozilla Firefox je pretjerano trošenje memorijskih resursa. Što se tiče sigurnosnih problema, iako je Firefox poznat kao jedan od sigurnijih preglednika, u inačicama 3.5.x otkriveno je nekoliko kritičnih sigurnosnih nedostataka.

Prvi se kritični nedostatak očituje kad korisnik ima instalirane dodatke za blokiranje reklama i oglasa (*AdBlock Plus)* ili za blokiranje zlonamjerno oblikovanih skripti (*NoScript,)* a napadaču

omogućava pokretanje proizvoljnog *JavaScript* koda s ovlastima *chrome* korisnika. Ovaj nedostatak ne posjeduju ranije inačice preglednika.

Otkriveno je i nekoliko nespecificiranih problema u pokretačkom programu preglednika (eng. *browser engine*). Poznato je kako mogu uzrokovati izmjenu proizvoljnih memorijskih lokacija, pri čemu može doći do rušenja ranjive aplikacije u određenim uvjetima. Pretpostavlja se da je takvu situaciju moguće iskoristiti i za pokretanje proizvoljnog koda.

Sljedeći je problem koji se može iskoristiti za izvođenje tzv. *phishing* napada. Pogreška koja se javlja prilikom otvaranja novog prozora za zlonamjerno oblikovanu domenu može se iskoristiti za prikaz proizvoljne URL adrese u adresnoj traci novootvorenog prozora. Za uspješnu zlouporabu potrebno je metodi "window.open()" podmetnuti naziv domene s npr. znakovima "%20".

Svi spomenuti propusti su uklonjen u inačici 3.5.2.

3.2. Internet Explorer

Preglednik Internet Explorer je grafički web preglednik tvrtke Microsoft. Od 1995. godine dolazi u paketu s operacijskim sustavima Microsoft Windows, a od 1999. godine postaje najrašireniji web preglednik. Najnovija inačica preglednika, Internet Explorer 8, izdana je u ožujku 2009. godine.

3.2.1. Prednosti

Neke od mogućnosti najnovije inačice preglednika koje korisnicima omogućuju lakše i sigurnije surfanje su sljedeće:

- poboljšanje performansi (veća brzina obavljanja zadataka),
- "pametnija" adresna traka koja na temelju nekoliko znakova dopunjava ostatak URL adrese prethodno posjećenih stranica,
- novi akceleratori za brže obavljanje svakodnevnih zadataka (prevođenje, definiranje riječi i sl.),
- tzv. *SmartScreen* filter za zaštitu od napada socijalnim inženjeringom,
- tzv. *Web slices* metoda za praćenje novosti na web stranicama,
- brzi oporavak od rušenja preglednika,
- privatnost prilikom surfanja pomoću *InPrivate Browsing* funkcionalnosti i dr.

3.2.2. Nedostaci

U najnovijoj inačici preglednika – Internet Explorer 8, u 2009. godini zabilježeno je 11 propusta, od kojih je jedan označen manje ozbiljnim, dok ih je 10 označeno ozbiljnim.

Teži propusti uzrokovani su dereferenciranjem oslobođene memorije, neispravnim rukovanjem tabličnim operacijama i podacima u priručnoj memoriji, neodgovarajućom analizom predložaka (eng. *style sheet*) dokumenata, nepravilnostima u rukovanju pozivima DHTML objekata te "XMLHttpRequest" zahtjevima i sl. Napadač ih može iskoristiti za korupciju memorije, pokretanje proizvoljnog programskog koda, zaobilaženje sigurnosnih ograničenja te pristup osjetljivim informacijama.

Manje je ozbiljan nedostatak uzrokovan pogreškom prilikom otvaranja novog prozora funkcijom "*window.open()*", pri čemu može doći do prikaza proizvoljnog sadržaja u prozoru preglednika.

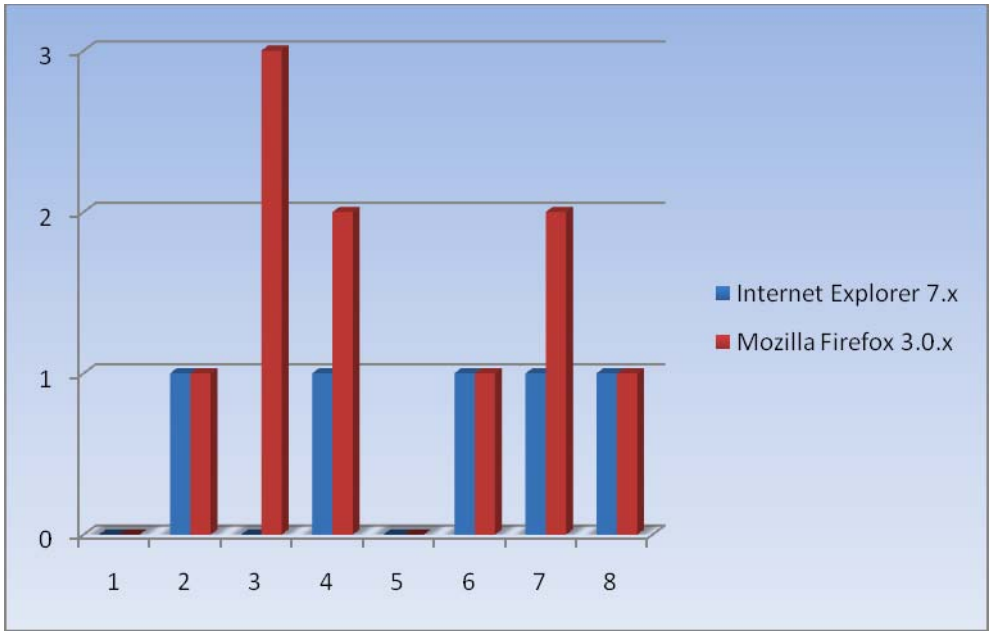
Proizvođač je uklonio sve kritične ranjivosti, dok je posljednji, manje ozbiljan problem, zasad još uvijek neriješen.

3.3. Mozilla Firefox vs. Internet Explorer

Budući da većina korisnika i dalje koristi nešto starije (stabilne) inačice preglednika u nastavku će biti data usporedba preglednika Internet Explorer 7.x i Mozilla Firefox 3.0.x.

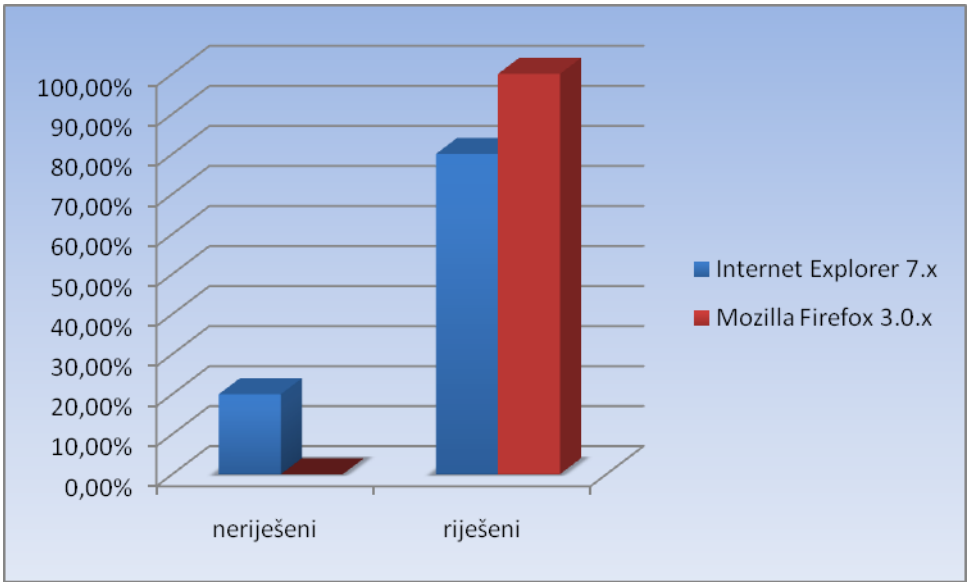
3.3.1. Statistike

Prema statistikama tvrtke Secunia, 2009. godine otkriveno je 5 sigurnosnih propusta preglednika Internet Explorer 7.x te 10 propusta preglednika Mozilla Firefox 3.0.x. Usporedbu broja ranjivosti po mjesecima moguće je vidjeti na sljedećem dijagramu.



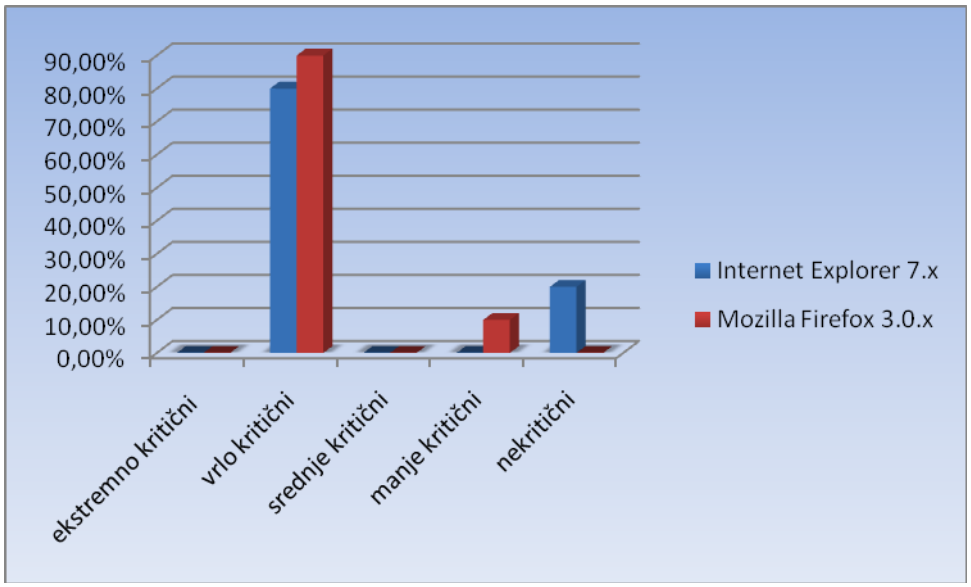
Dijagram 1: Usporedba broja ranjivosti u 2009. Godini
Izvor: Secunia.com

Dijagram 2 pokazuje odnos riješenih i neriješenih propusta za oba preglednika do kolovoza 2009. godine. Vidljivo je kako su svi otkriveni propusti preglednika Firefox ispravljeni, dok za 20% propusta preglednika Internet Explorer još uvijek nisu izdane zakrpe.



Dijagram 2: Usporedba riješenih i neriješenih propusta u 2009. Godini
Izvor: Secunia.com

Prema dijagramu 3, u 2009. godini nije otkriven nijedan ekstremno kritičan propust niti u Mozilli Firefox, niti u Internet Exploreru. 90 % propusta preglednika Mozilla Firefox označeno je vrlo kritičnima, dok je 10% označeno manje kritičnim. Kod preglednika Internet Explorer 80% otkrivenih propusta je vrlo kritičnih, dok je 20% (zasad neriješenih) nekritičnih.



Dijagram 3: Kritičnost otkrivenih propusta
Izvor: Secunia.com

Do prije nekoliko mjeseci prevladavalo je mišljenje da je preglednik Mozilla Firefox puno sigurniji od Internet Explorera. Međutim, prema prikazanim statističkim podacima, ta razlika više nije toliko očita.

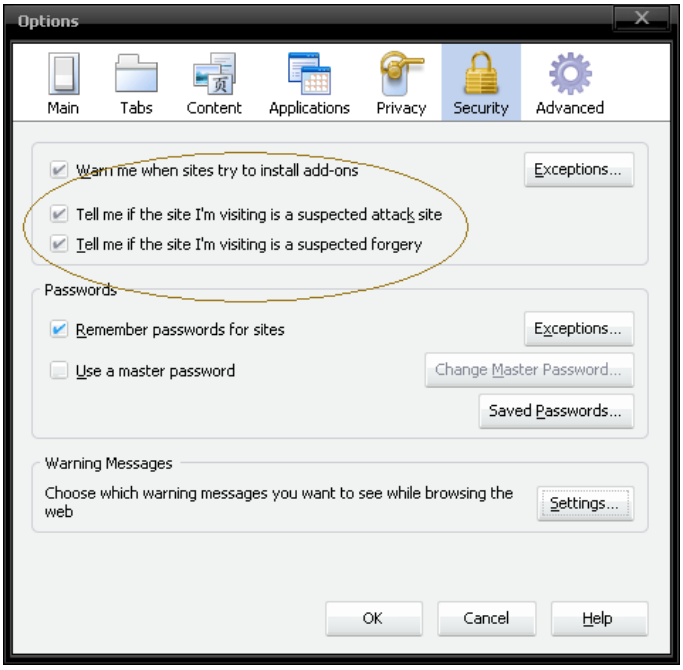
4. Metode za povećanje sigurnosti web preglednika

U metode za povećanje sigurnosti web preglednika ubrajaju se akcije koje sami korisnici mogu provoditi kako bi se što više zaštitili od napada, kao i metode koje je moguće primijeniti na popularne web preglednike korištenjem nekih od programskih paketa za virtualizaciju. Riječ je o metodama *sandbox* i virtualizacija aplikacije.

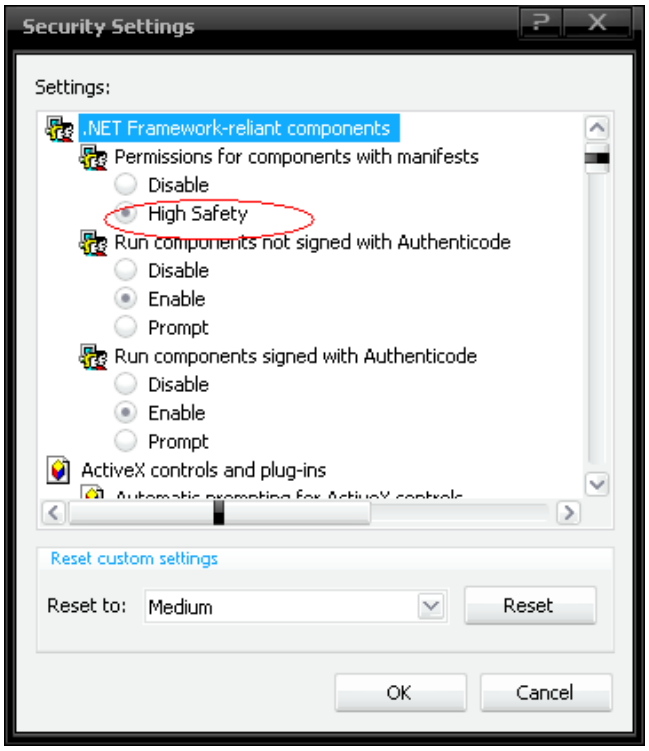
4.1. Sigurno surfanje

Svaki korisnik mogao bi povećati sigurnost svog preglednika i time smanjiti količinu sigurnosnih prijetnji koristeći sljedećih nekoliko metoda:

- Učestala nadogradnja preglednika na najnovije inačice sa svim potrebnim zakrpama – korisnici obično prime obavijest o izdavanju nove inačice.
- Nadogradnja korištenog operacijskog sustava – ako se koristi operacijski sustav Windows, najbolje je uključiti automatsku nadogradnju sustava.
- Korištenje najnovijih inačica antivirusnih i drugih zaštitnih programa (npr. vatrozida), pri čemu je važno da se nove inačice programa preuzmu sa službenih (a time i sigurnih) stranica.
- Redovita nadogradnja programa korištenih za pregled multimedijskih sadržaja (*Flash, Java* i dr.) – najčešće na web stranicama dodataka postoji mogućnost provjere korištene inačice i najnovije izdane inačice, kao na primjer na stranici za *Adobe Flash Player* - http://www.adobe.com/software/flash/about/. Budući da su takvi dodaci često uzroci sigurnosnih ranjivosti, potrebno je redovito provjeravati je li izdana nova inačica (jer ona obično sadrži sigurnosne zakrpe).
- Blokiranje *pop-up* prozora radi sprečavanja preuzimanja zlonamjernih programa na korisničko računalo. P*op-up* prozori se najčešće javljaju prilikom preuzimanja slika, besplatne glazbe i sl. Većina popularnih preglednika posjeduje mogućnost blokiranja takvih prozora:
  - Firefox: *Tools->Options->Content->Block pop-up windows*
  - Internet Explorer: *Tools->Pop-up Blocker*
  - Opera: *Tools->Preferences->General->Block Unwanted Pop-ups/Block All Pop-ups*
- Provjera sigurnosnih postavki web preglednika i postavljanje istih na barem srednju razinu zaštite - svaki od preglednika posjeduje neke sigurnosne metode za zaštitu od lažiranih stranica, stranica sa zlonamjernim programima i sl. Na korisniku je da, u ovisnosti o potrebama, odabere u kojim će ga slučajevima preglednik upozoriti na sigurnosnu prijetnju. Sa stajališta sigurnosti, najbolje bi bilo uključiti sve zaštitne opcije, što međutim može postati i smetnja prilikom korištenja alata. Na sljedećim su slikama prikazane neke od zaštitnih opcija preglednika Firefox i Internet Explorer.



Slika 2: Sigurnosne postavke preglednika Firefox



Slika 3: Sigurnosne postavke preglednika Internet Explorer

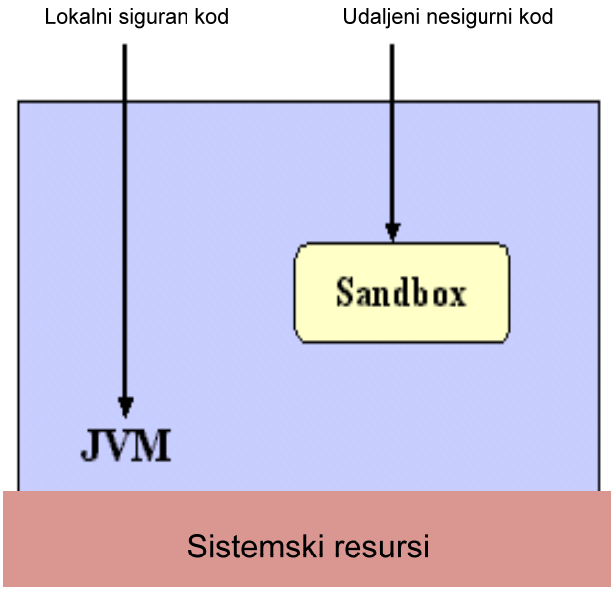
- Onemogućavanje *JavaScript*-a i nepotrebnih *ActiveX* kontrola (osim u slučaju da je nešto od toga korisniku zaista potrebno, kada je bolje izmijeniti postavke preglednika na način da upozorava korisnika prilikom svakog pokretanja skripte). Za preglednik Mozilla Firefox za potrebe blokiranja skripti izdan je dodatak *NoScript*, dok je kod preglednika Internet Explorer potrebno ručno u izborniku *Tools->Internet Options->Security* odabrati blokiranje skripti i ActiveX kontrola.

4.2. "Sandbox" metoda

Razlog zbog kojeg je iznimno važna implementacija neke sigurnosne metode u preglednicima jest sljedeći – web preglednik je postao značajan za izvođenje većine funkcija koje su korisnicima potrebne, a s druge strane je vrlo česta meta sigurnosnih napada, odnosno i sama njegova funkcionalnost najčešće se iskorištava u svrhu izvođenja napada.

U računalnoj sigurnosti, *sandbox* je sigurnosni mehanizam namijenjen razdvajanju pokrenutih programa. Često se koristi prilikom izvođenja nesigurnih programa dobivenih od neprovjerenih izvora. Ova metoda se uglavnom svodi na osiguravanje dobro kontroliranog skupa resursa (diskovnog prostora ili memorije) potrebnih za rad takvim gostujućim programima. Većina ostalih funkcija, kao što je npr. mrežni pristup, najčešće je zabranjena ili strogo ograničena.

Sam koncept *sandbox* sigurnosnog modela vrlo je dobro definiran u sklopu ranijih inačica Java aplikacija. Ideja je dopustiti gostujućem programu pokretanje i dodijeliti mu pristup određenim sistemskim resursima, ali ograničiti njegovo djelovanje na određeno područje kako bi se umanjio njegov utjecaj na ostatak sustava. Shema takvog jednog sustava prikazana je na sljedećoj slici:



Slika 4: Primjer *sandbox* modela zaštite

Sandbox metoda sa stajališta sigurnosti pregledniku omogućava ograničen pristup kritičnim sistemskim datotekama, bibliotekama i sl. Drugim riječima, preglednik ima pristup samo komponentama potrebnim za njegovu potpunu funkcionalnost. Na taj način, iskorištavanje neke njegove ranjivosti imat će znatno manji utjecaj na ostatak sustava. Isto tako, *sandbox* stvara prividni spremnički prostor potreban za rad preglednika, budući da bi sam preglednik mogao napraviti trajne izmjene na tvrdom disku. Zato mu se pokušava stvoriti virtualni spremnički prostor čije je izmjene kasnije moguće poništiti. Upravo ta mogućnost je posebno važna za sigurnost preglednika, budući da napadači kod većine napada korištenjem web preglednika pokušavaju trajno spremiti izmjene na sustavu, osigurati interakciju sa sustavom ili ubaciti neki zlonamjerno oblikovan program. Iako ova metoda ne može spriječiti sve napade, može znatno ograničiti njihov utjecaj na sustav.

Najčešći primjeri implementacije *sandbox* metode su *applet*-i. Riječ je o samostalnim programima koji se pokreću na virtualnom računalu ili u interpreteru nekog skriptnog jezika koji tada simulira *sandbox*. *Applet*-i su česti u web preglednicima gdje se ovaj mehanizam koristi za sigurno izvođenje potencijalno nesigurnog koda ugrađenog u web stranice. Preglednik Mozilla Firefox koristi sigurnosne pretince (eng. *sandboxes*) za izvođenje *Java* i *JavaScript* koda. Time bi se trebalo spriječiti nanošenje štete korisničkom računalu.

4.3. Metoda virtualizacije aplikacije

Alternativna metoda za poboljšanje sigurnosti web preglednika je metoda virtualizacije aplikacije. Jedan od najvećih izazova vezanih uz poboljšavanje sigurnosti preglednika jest svakodnevno održavati njegovu konfiguraciju sigurnom. Problem je u tome što je osim samog preglednika potrebno održavati, nadograđivati i osiguravati konfiguracije svih pomoćnih aplikacija i dodataka, kao što su *Acrobat Reader, Flash, Quicktime, Windows Media Player, Webex, Skype,* itd. Svaka od tih aplikacija namijenjenih proširenju funkcionalnosti preglednika ujedno povećava njegovu ranjivost.

Sama definicija virtualizacije aplikacije je sljedeća: virtualizacija aplikacije odvaja program od operacijskog sustava na kojem se izvodi te osigurava povećanje prenosivosti, upravljivosti i kompatibilnosti aplikacija.

Iako virtualizirane aplikacije nisu zaista instalirane na operacijskom sustavu, izvode se kao da jesu. Na taj je način veća mogućnost postizanja homogenosti i konzistentnosti web preglednika. Korištenjem virtualizacije jednu centralnu instancu aplikacije (u ovom slučaju preglednika) moguće je nadograditi i konfigurirati na željeni način prije pokretanja, ali isto tako i održavati tijekom korištenja.

Postoje razne aplikacije koje osiguravaju ovaj tip funkcionalnosti preglednika - *Citrix XenApp, Microsoft Application Virtualization* (prethodno poznata pod nazivom *SoftGrid*), *VMWare ThinApp* (prethodno poznat pod nazivom *Thinstall*) i dr. U ovom će dokumentu biti opisana funkcionalnost alata *VMWare ThinApp*.

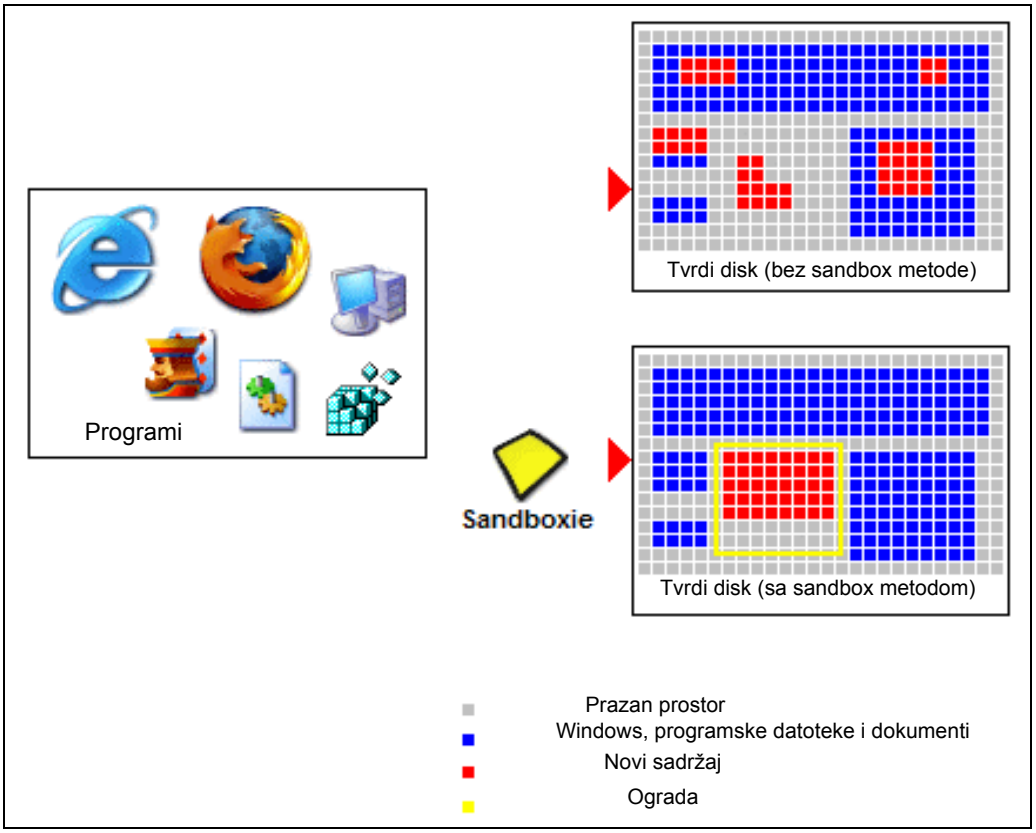
5. Alati za povećanje sigurnosti preglednika

5.1. Sandboxie

Kao što i sam naziv kaže, *Sandboxie* je alat koji implementira zaštitnu metodu *sandbox*. Njegov autor, Ronen Tzu, razvio ga je 2004. godine. Iako mu je u početku jedina zadaća bila pokretanje preglednika Internet Explorer u izoliranom prostoru, danas mu je funkcionalnost proširena na mnoštvo drugih programa i njihovu interakciju. Alat je dostupan u dvije inačice – besplatnoj i komercijalnoj, pri čemu besplatna inačica sadrži neka ograničenja (nemogućnost korištenja više zaštitnih "pretinaca" i sl.).

Pokretanje programa u izoliranom prostoru pomoću alata *Sandboxie* onemogućava spremanje trajnih izmjena nad drugim programima ili podacima na računalu. Razlika između alata *Sandboxie* i tradicionalnih alata namijenjenih zaštiti od zlonamjernih programa jest što *Sandboxie* stvara zaštitni sloj koji blokira trajnu interakciju svih programa sa sustavom, dok ostali alati jednostavno pokušavaju blokirati poznate zlonamjerno oblikovane programe.

Na sljedećoj je slici ilustrirana funkcionalnost alata.



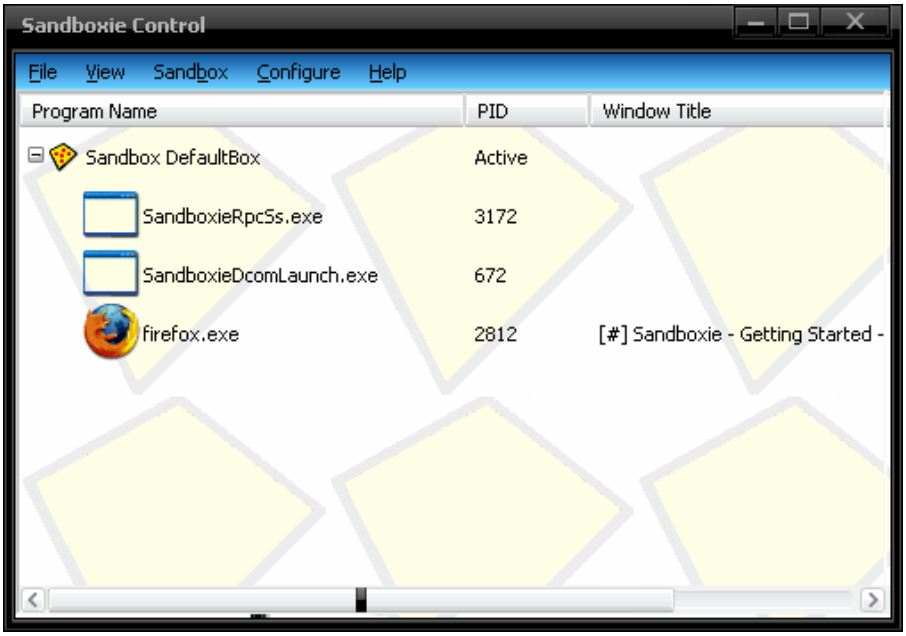
Slika 5: Ilustracija rada alata *Sandboxie*

Prednosti korištenja alata *Sandboxie* su sljedeće:

- Sigurnije surfanje – svi zlonamjerni programi preuzeti pomoću preglednika spremaju se na isto mjesto i moguće ih je uništiti sve odjednom.
- Veća privatnost – povijest surfanja, kolačići (eng. *cookie*) i datoteke iz privremene memorije ostaju unutar granica određenih alatom i ne djeluju na operacijski sustav.
- Sigurna elektronička pošta – virusi i drugi zlonamjerni programi iz privitaka ne mogu utjecati na sustav.
- Programi se instaliraju u izolirani prostor pa se tako ne povećava "veličina" samog operacijskog sustava.

5.1.1. Instalacija i korištenje

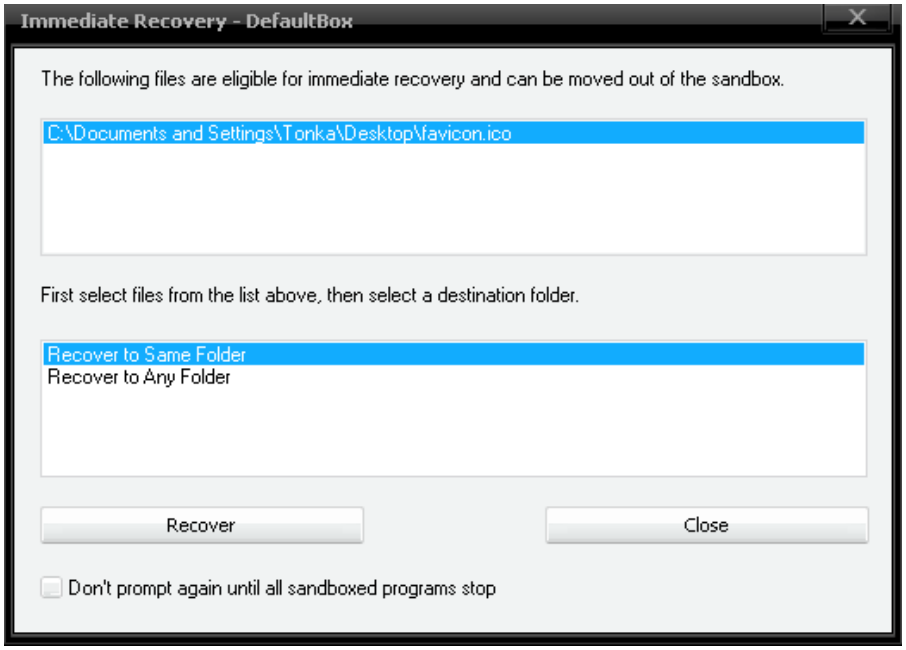
Instalacija alata provodi se u nekoliko jednostavnih koraka kroz koje korisnike vodi instalacijski čarobnjak. Sučelje alata prikazano je na sljedećoj slici:



Slika 6: Sučelje *Sandboxie* alata

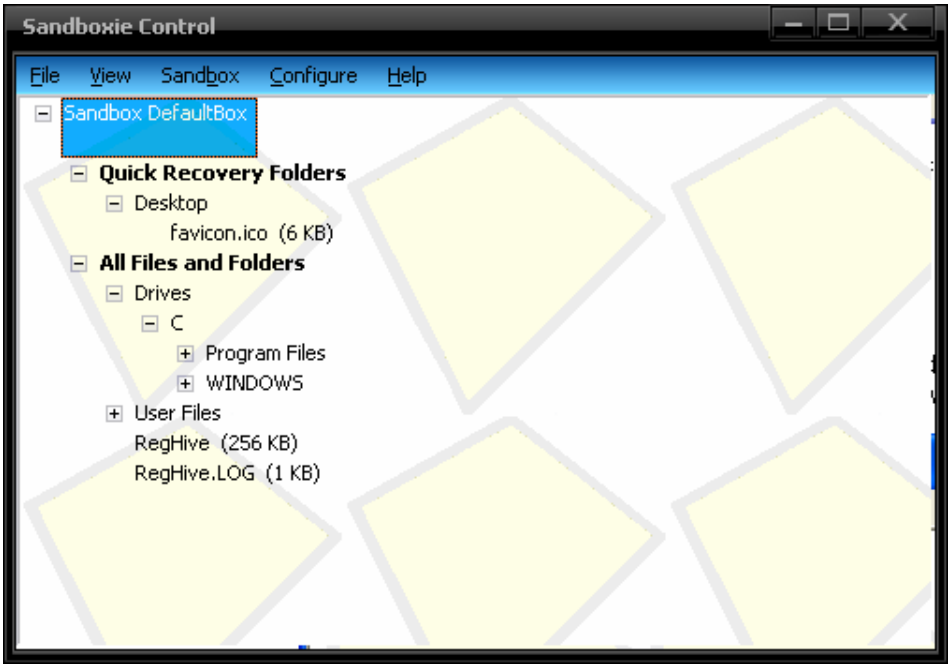
Unutar sučelja je prikazana lista trenutno pokrenutih programa u *sandbox* načinu rada. Sve izmjene koje ti programi izazovu, mogu biti spremljene u podrazumijevani (eng. *default*) "pretinac" ili je moguće stvoriti nove "pretince" za određene programe.

Sve preuzete datoteke tako će biti spremljene u izolirani prostor, dok god web preglednik radi u *sandbox* načinu rada. To podrazumijeva da će prilikom brisanja izmjena biti izbrisane i datoteke koje su korisnici namjerno preuzeli pomoću preglednika. Zato *Sandboxie* omogućava izbacivanje tih željenih datoteka prije brisanja "pretinca", pri čemu pretpostavlja da je korisnik namjerno preuzeo datoteke spremljene na radnu površinu, te direktorije *Favorites* i *My Documents*. Iz tog razloga se prilikom spremanja datoteka na neku od spomenutih lokacija korisniku nudi da ju odmah spremi izvan "pretinca", kao što je prikazano na sljedećoj slici.



Slika 7: *Sandboxie* – obnavljanje preuzetih datoteka

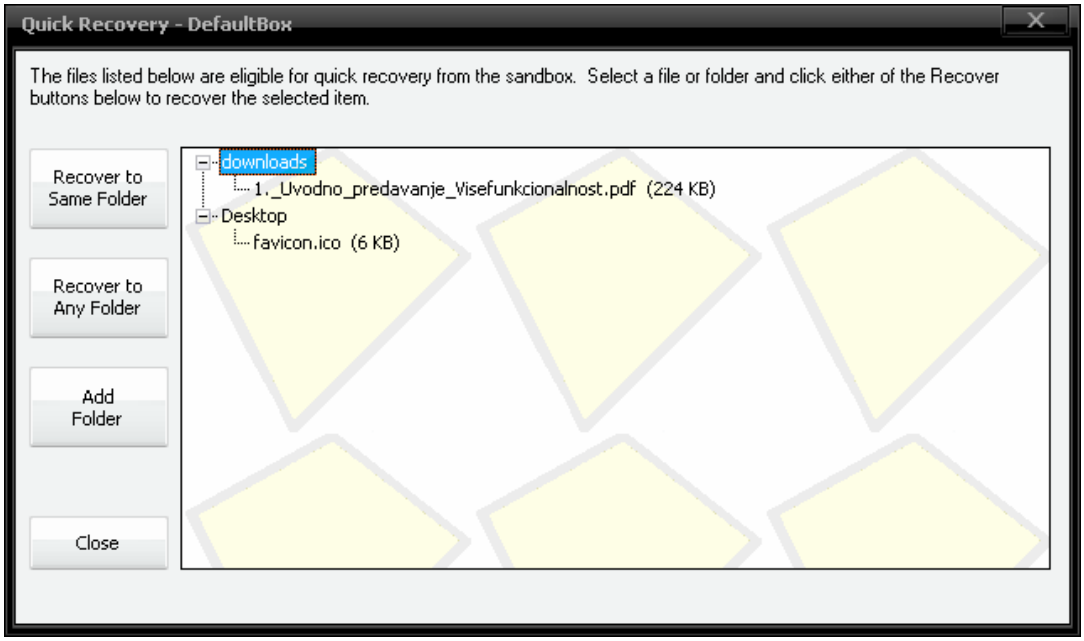
Ostale datoteke, koje se ipak spreme u pretinac, korisnik može pregledati odabirom *View->Files And Folders View*, kao što je prikazano na sljedećoj slici.



Slika 8: Pregled *Sandboxie* pretinca

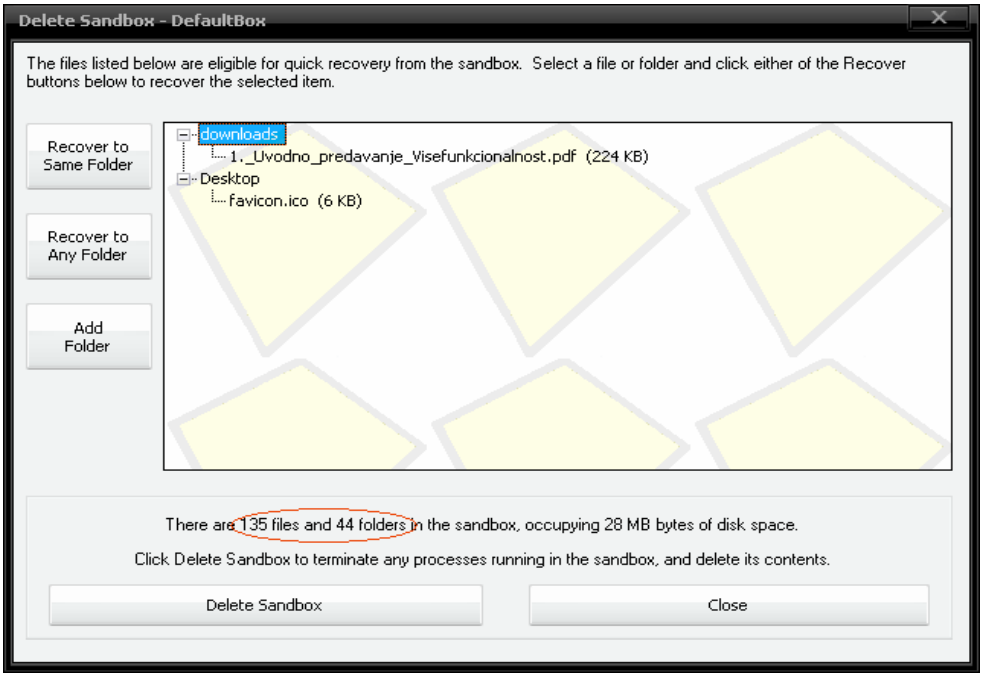
Na isti način *Sandboxie* može ograničiti i rad drugih programa, odnosno funkcionalnost mu nije ograničena samo na preglednike.

Korištenjem opcije *Quick Recovery*, korisnicima je omogućeno spremanje datoteka izvan pretinca, pri čemu *Sandboxie* inicijalno omogućava trajno spremanje samo datoteka iz prethodno spomenutih direktorija. Ako korisnik želi imati mogućnost trajnog spremanja datoteke koju je spremio u neki drugi direktorij, tada taj direktorij mora dodati na listu lokacija na koje je spremanje omogućeno.



Slika 9: *Sandboxie Quick Recovery*

Nakon što korisnik završi s radom u pregledniku, preporučljivo je izbrisati sadržaj *Sandboxie* pretinca.



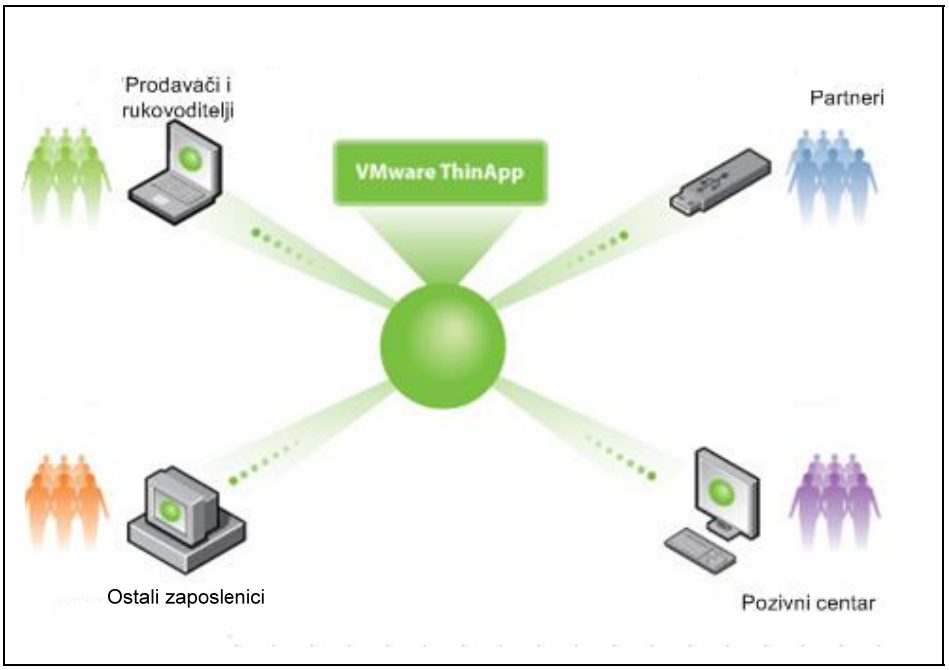
Slika 10: *Sandboxie - b*risanje pretinca

Sadržaj pretinca ne briše se automatski, osim ako korisnik ne odabere opciju *Automatically delete contents of sandbox*.

5.2. ThinApp

Alat *ThinApp* je, za razliku od *Sandboxie* alata, više orijentiran na organizacije. Riječ je o komercijalnom alatu, kojeg je moguće besplatno testirati 60 dana. Njegova funkcionalnost organizacijama omogućava centralizirano upravljanje aplikacijama i kontrolu pristupa. Osigurava mogućnost pokretanja programa bez potrebe za izmjenama lokalnog operacijskog sustava, datotečnog sustava ili *registry* zapisnika. Pomoću virtualizacije pomaže organizacijama u konfiguraciji, održavanju i nadogradnji korištenog web preglednika i svih pomoćnih aplikacija.

Korištenjem alata *ThinApp* moguće je na centralnoj lokaciji kreirati prilagodljivi prenosivi aplikacijski paket web preglednika – čitava aplikacija i njezin virtualni operacijski sustav pakiraju se u jednu izvršnu datoteku, koju tada svi klijenti mogu koristiti. Takva "virtualizirana" aplikacija može se pokretati u korisničkom načinu rada, bez administrativnih ovlasti, čime se osigurava dodatna zaštita. Nadalje, može se konfigurirati te joj se mogu dodati sigurnosni dodaci i nadograđene inačice pomoćnih aplikacija kao što su *Flash, Quicktime* i sl. Bez virtualizacije aplikacija, kad bi neka organizacija odlučila instalirati novi preglednik, morala bi ga instalirati na sustave prije korištenja te pustiti samog korisnika da se bavi nadogradnjom preglednika, dodataka i pomoćnih aplikacija. To očito nije zadovoljavajuće rješenje. Korištenjem alata *Thinapp*, dio zaposlenika zaduženih za održavanje jednostavno nadograđuje centralni preglednik, a ostali korisnici prilikom sljedećeg pokretanja koriste tako nadograđeni preglednik. Ilustracija korištenja alata *ThinApp* prikazana je na sljedećoj slici.



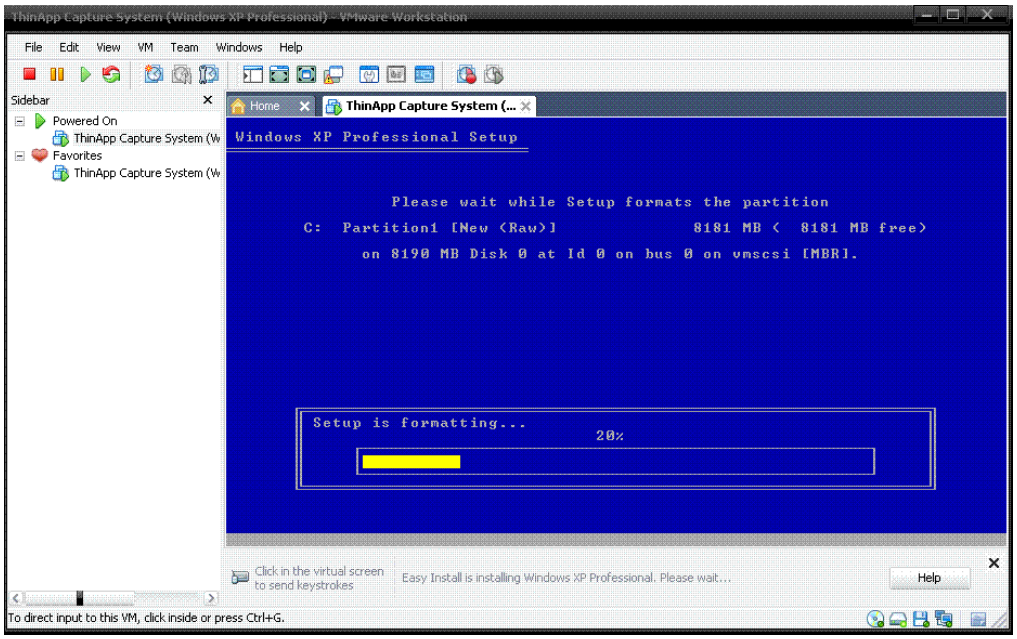
Slika 11: Ilustracija korištenja alata *ThinApp*

5.2.1. Instalacija i korištenje

Korištenje alata *ThinApp* nešto je kompliciranije od korištenja alata *Sandboxie*. Prilikom izrade prenosivog aplikacijskog paketa prikupljaju se sve datoteke koje aplikacija (u našem slučaju preglednik) koristi u jedan prenosivi paket neovisan o operacijskom sustavu. Stvaranje prenosive aplikacije preporuča se na čistom operacijskom sustavu jer se proces stvaranja aplikacije svodi na provjeru stanja sustava prije i poslije njene instalacije. Ako pritom u pozadini radi neki drugi program, može se dogoditi da i njegove datoteke budu uključene u aplikacijski paket, što nije poželjno. Proizvođač za kreiranje (čistog) virtualnog operacijskog sustava preporuča korištenje paketa *Vmware Workstation* što međutim nije neophodno za korištenje alata *ThinApp*.

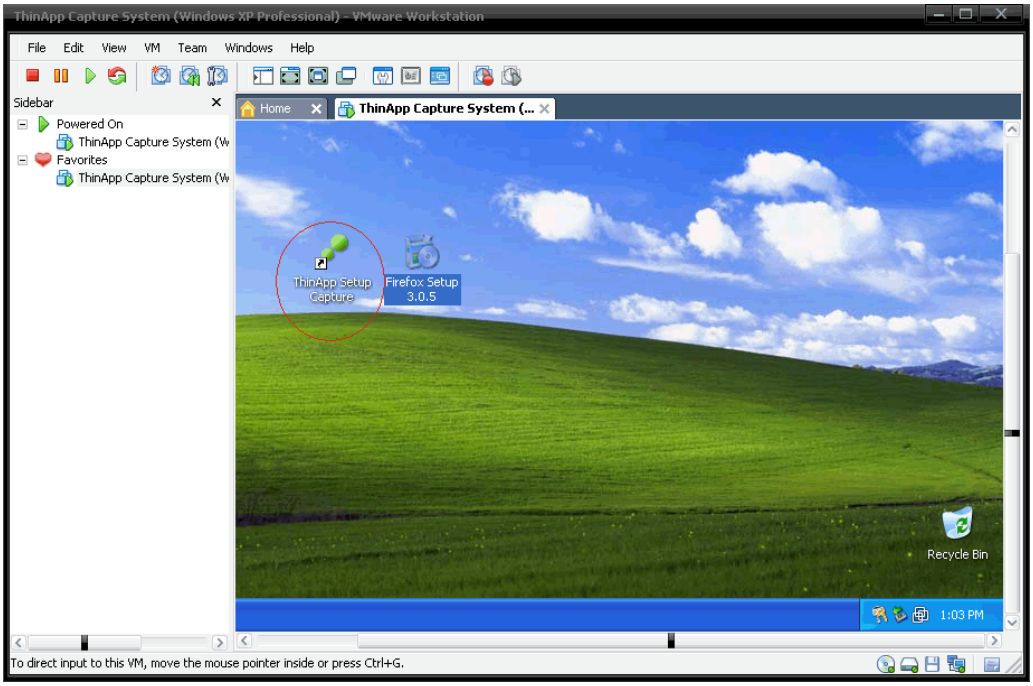
Preporučeni koraci koje bi korisnici trebali slijediti prilikom izrade prenosive aplikacije su:

- instalacija programskog paketa *Vmware Workstation*,
- stvaranje novog virtualnog računala s operacijskim sustavom Windows,
- pokretanje kreiranog virtualnog računala, što je prikazano na sljedećoj slici:



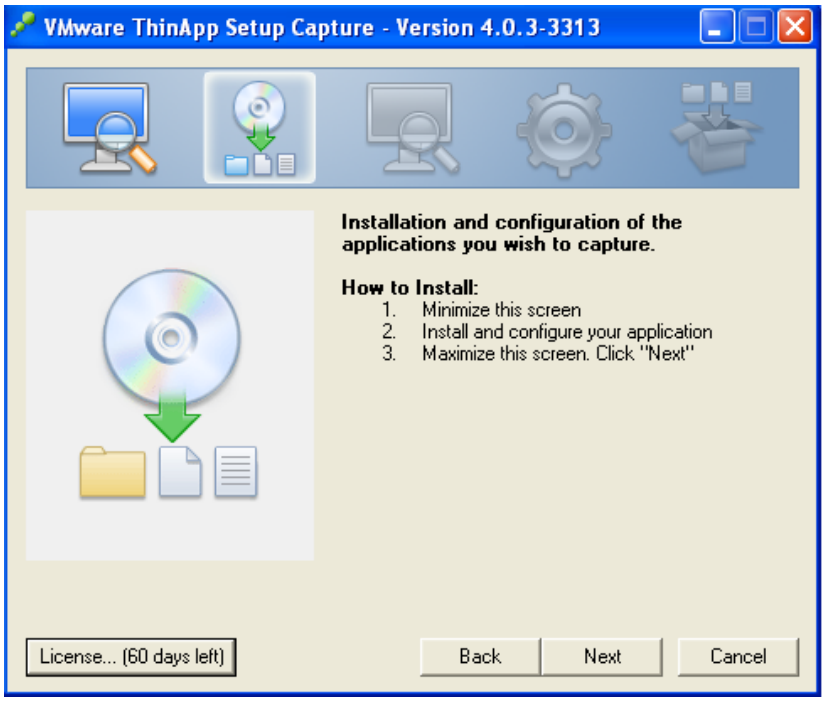
Slika 12: Pokretanje virtualnog računala

- konfiguriranje virtualnog računala,
- instalacija programskog paketa *ThinApp* na virtualnom računalu,



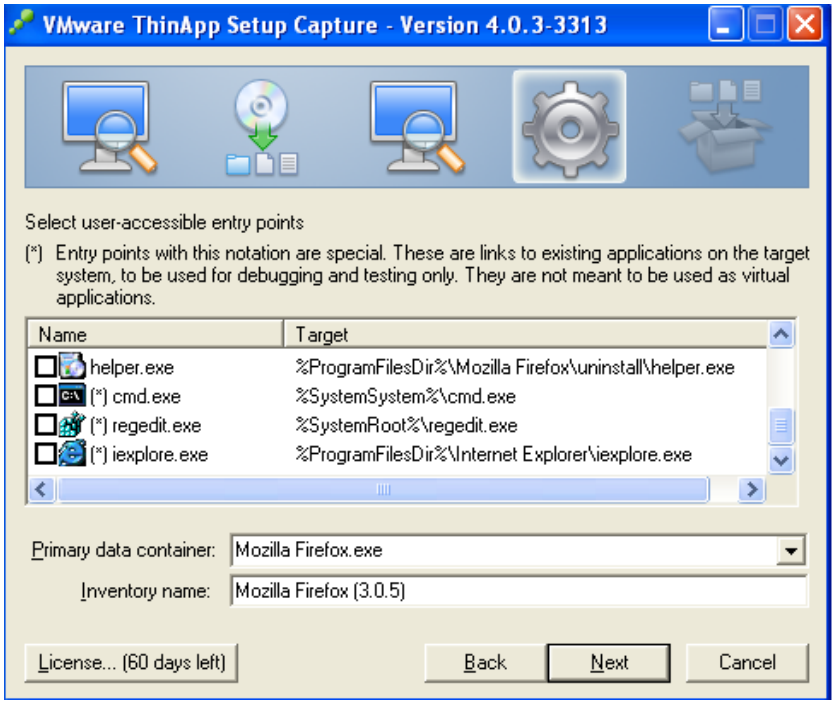
Slika 13: Instalacija paketa *ThinApp* na virtualnom računalu

- stvaranje aplikacijskog paketa web preglednika pomoću alata *ThinApp*, što uključuje:
  - pokretanje *ThinApp* paketa prije instalacije preglednika,



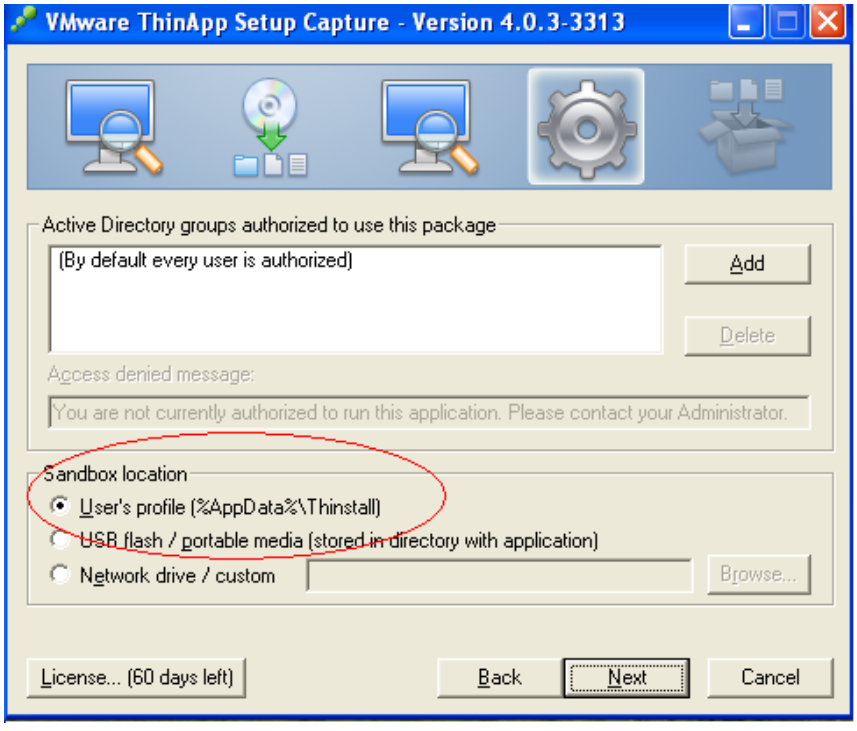
Slika 14: Pokretanje alata *ThinApp*

  - pokretanje instalacije preglednika,
  - konfiguracija preglednika na željeni način,
  - instalacija dodatnih aplikacija (*Adobe Flash, Java* i sl.),
  - ponovno pokretanje alata *ThinApp* nakon instalacije i konfiguracije preglednika,



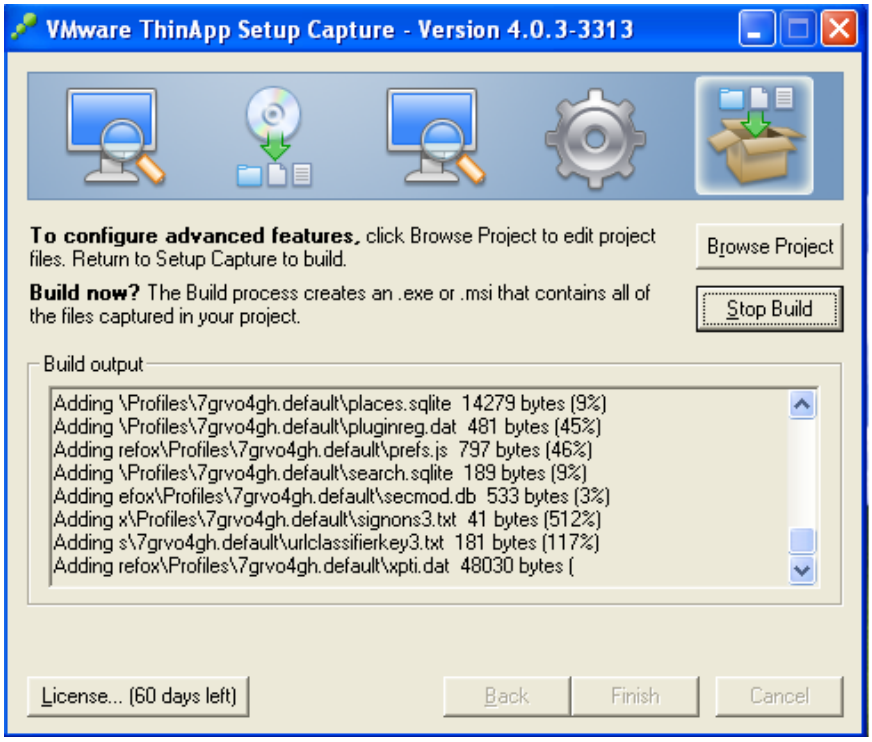
Slika 15: Pokretanje alata nakon instalacije preglednika

- odabir lokacije u koju će se spremati preuzete datoteke (slično kao kod alata *Sandboxie*),



Slika 16: Odabir lokacije "pretinca" za spremanje

- provjera dodatnih mogućnosti i brisanje nepotrebnih podataka prije spremanja te
- spremanje projekta.



Slika 17: Spremanje projekta

- Posljednji je korak kopiranje stvorenog projekta na fizičko računalo, CD/DVD, usb disk ili neki drugi uređaj.

Tako izrađeni paket željene konfiguracije može se koristiti na klijentskim računalima pokretanjem izvršne datoteke preglednika (u *bin* datoteci paketa), bez potrebe za instalacijom na lokalno računalo.

6. Budućnost

Novi web preglednici i nove inačice već postojećih preglednika razvijaju se sa sve većim naglaskom na sigurnost. U predviđanjima o najsigurnijim preglednicima budućnosti prevladavaju sljedeća 3 preglednika:

- Internet Explorer 8,
- Google Chrome i
- Gazelle.

6.1. Internet Explorer 8

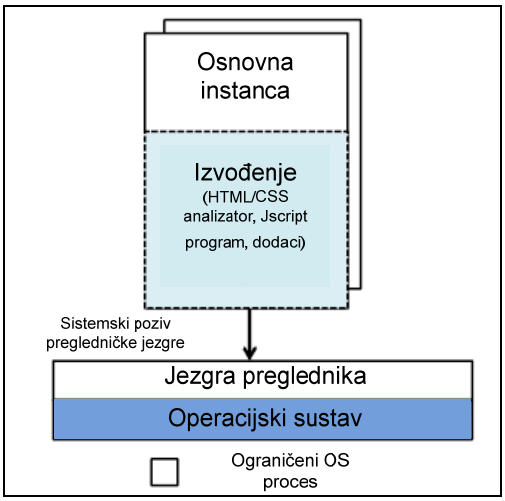
Najnoviju inačica preglednika Internet Explorer moguće je preuzeti sa službenih stranica. Radi se o poboljšanoj inačici koja bi, sa stajališta sigurnosti, korisnicima trebala osigurati sljedeće:

- zaštitu od zlonamjernih programa i *phishing* napada,
- *SmartScreen* zaštitu od lažiranih web stranica,
- filter za zaštitu od XSS napada,
- zaštitu od otvaranja neželjenih stranica izvođenjem tzv. "*Clickjacking*" napada,
- DEP (eng. *Data Execution Prevention*) mogućnost za zaštitu od pisanja po određenom memorijskom prostoru,
- dodatke za povećanje privatnosti i brisanje povijesti pregledavanja (*InPrivate Filtering* i *Delete Browsing History*) te
- automatski oporavak od rušenja preglednika.

Neke od spomenutih mogućnosti su postojale i u prethodnoj inačici preglednika (zaštita od *phishing* i XSS napada, brisanje povijesti pregledavanja i oporavak od rušenja preglednika), ali im je u novoj inačici unaprijeđena funkcionalnost. Ostale sigurnosne mogućnosti dostupne su tek od najnovije inačice preglednika.

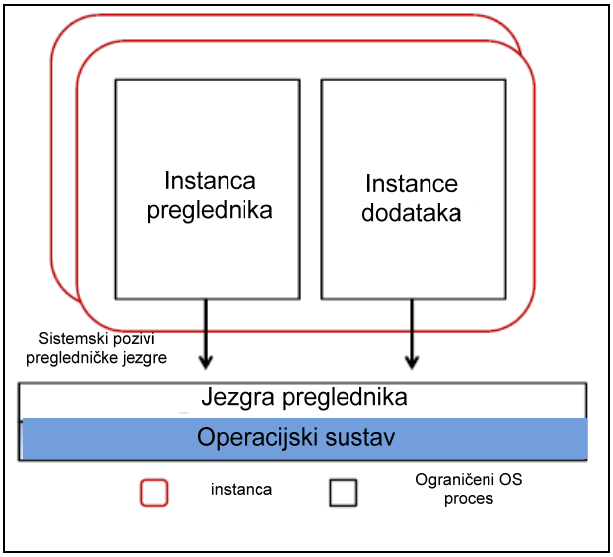
6.2. Gazelle

Microsoft radi na razvoju novog web preglednika koji bi trebao biti sigurniji od popularnih preglednika kao što su Firefox, Google Chrome i različite inačice Internet Explorera. Riječ je o web pregledniku koji koristi tehnike operacijskog sustava s ciljem razdvajanja aplikacija i njihove izolacije od sustava na kojem su pokrenute. Može se reći da je ovaj preglednik sam po sebi operacijski sustav (s vlastitom jezgrom i resursima) pokrenut unutar nekog drugog sustava. Ovakva arhitektura osigurava da bilo kakav zlonamjerni kôd (skripte, virusi, štetni dodaci i sl.) djeluju samo na vlastiti proces, a na druge web aplikacije, jezgru preglednika i sam operacijski sustav nemaju utjecaj. Na slikama koje slijede prikazana je zamišljena arhitektura novog preglednika.



Slika 18: Prikaz osnovne korištene arhitekture

Slika 18 prikazuje osnovnu arhitekturu razdvajanja aplikacija s ciljem zaštite od njihovog međusobnog utjecaja.



Slika 19: Arhitektura preglednika Gazelle

Slika 19 prikazuje nešto izmijenjenu (poboljšanu) arhitekturu, u kojoj se kao dodatna razina zaštite koristi i razdvajanje samog preglednika od dodataka.

Zasad je ovaj projekt tek u razvoju i spomenuti su samo koncepti koji će se pritom koristiti. Međutim, ako se te ideje uspiju realizirati, nema sumnje da bi Gazelle mogao biti opasna konkurencija današnjim preglednicima (pogotovo po pitanju sigurnosti)

6.3. Google Chrome

Google Chrome je web preglednik namijenjen bržem, lakšem i sigurnijem surfanju, razvijen s naglaskom na minimalističkom dizajnu. Između ostalih mogućnosti koje nudi (jedan okvir za sve, stranica u novoj kartici, prečaci aplikacije, dinamičke kartice i dr.), neke od sigurnosnih postavki koje posjeduje su sljedeće:

- Zaštita od pada sustava – svaka kartica (eng. *tab*) koju korisnik koristi u pregledniku izvodi se zasebno. Na taj se način sprječava rušenje drugih kartica u slučaju pada jedne aplikacije.
- Pregledavanje *inkognito* - postavka namijenjena povećanju privatnosti prilikom pregledavanja, onemogućava spremanje odabranih stranica u povijesti weba.
- Sigurno pregledavanje – preglednik upozorava korisnika u slučaju posjeta stranici s mogućom aktivnošću krađe identiteta, mogućnošću zaraze zlonamjernim programom i sl.

U korist ovog preglednika govori i činjenica kako je na prethodno spomenutom natjecanju *Pwn2Own* jedino Chrome uspio proći testiranje bez otkrivanja ranjivosti, dok su kod ostalih preglednika odmah uočeni sigurnosni propusti.

Početkom srpnja 2009. godine Google je objavio kako radi na novom operacijskom sustavu otvorenog koda – Google Chrome OS. Riječ je o pregledniku Chrome pokrenutom na jezgri operacijskog sustava Linux. Planira se potpuni redizajn sigurnosne arhitekture operacijskog sustava tako da se korisnici ne moraju brinuti o virusima, zlonamjernim programima i sigurnosnim zakrpama. Svi bi se poslovi trebali izvoditi i spremati na webu, tako da tradicionalni štetni programi ne bi mogli utjecati na takav sustav.

7. Zaključak

Svakodnevno se javljaju novi trendovi sigurnosnih prijetnji i ranjivosti web preglednika, što povećava rizik kojeg korisnici moraju biti svjesni prilikom surfanja. Isto tako, razvija se sve više web aplikacija koje zahtijevaju dodatne funkcionalnosti preglednika, što također predstavlja sigurnosni problem. Najveći problem u implementaciji sigurnosnih ideja jest postići da se istovremeno poveća sigurnost preglednika, ali bez utjecaja na potrebnu funkcionalnost. U ovom su dokumentu opisane neke od tehnika korištenih za poboljšanje sigurnosti preglednika – *sandbox* i metoda virtualizacije aplikacije. Obje metode imaju isti cilj, a temelje se na različitim konceptima. Može se reći da, iako se virtualizacijske tehnike koriste na području informacijske sigurnosti već godinama, na području osiguravanja web preglednika primjenjuju se tek odnedavno.

Osim spomenutih metoda, stručnjaci rade i na razvoju novih, sigurnijih preglednika. Riječ je o sasvim novim idejama korištenja operacijskog sustava kao web preglednika (Google) i web preglednika kao operacijskog sustava (Microsoft). Ti bi projekti mogli pokrenuti revoluciju u računalnoj sigurnosti jer tradicionalni zlonamjerni programi na njih neće moći utjecati. To će zasigurno rezultirati novom generacijom zlonamjernih programa temeljenih na webu, što bi čak moglo utjecati i na poslovanje antivirusnih organizacija.

Ipak, unatoč sigurnijim web preglednicima i dalje je potrebno koristiti antivirusne i druge zaštitne programe te korisnici moraju paziti na svoje postupke prilikom surfanja jer upravo oni sami sebe nepromišljenim ponašanjem najviše dovode u opasnost.

8. Reference

1. Web browser, http://en.wikipedia.org/wiki/Web_browser, kolovoz 2009.
2. The Hack FAQ, Web Browser as Attack Point, http://www.nmrc.org/pub/faq/hackfaq/hackfaq-08.html, kolovoz 2009.
3. SQL injection, http://en.wikipedia.org/wiki/SQL_injection, kolovoz 2009.
4. Sigurnost web aplikacija, Mario Kozina, http://www.zemris.fer.hr/~sgros/publications/diploma_thesis/kozina_mario_seminar.pdf, kolovoz 2009.
5. Arbitrary code execution, http://en.wikipedia.org/wiki/Arbitrary_code_execution, kolovoz 2009.
6. Five common Web application vulnerabilities, http://www.securityfocus.com/infocus/1864, kolovoz 2009.
7. Format string attack, http://www.securityfocus.com/infocus/1864, kolovoz 2009.
8. Cross-site scripting, http://en.wikipedia.org/wiki/Cross-site_scripting, kolovoz 2009.
9. Username enumeration vulnerabilities, http://www.gnucitizen.org/blog/username-enumeration-vulnerabilities/, kolovoz 2009.
10. Web browser attacks, http://www.digitalbond.com/index.php/2009/04/15/web-browser-attacks/, kolovoz 2009.
11. Browser Statistics, http://www.w3schools.com/browsers/browsers_stats.asp, kolovoz 2009.
12. Mozilla Firefox, http://www.mozilla.com/, kolovoz 2009.
13. Security Advisories for Firefox 3.5, http://www.mozilla.org/security/known-vulnerabilities/firefox35.html, kolovoz 2009.
14. Internet Explorer 8: Home page, http://www.microsoft.com/windows/internet-explorer/default.aspx, kolovoz 2009.
15. Internet Explorer 7 vs. Mozilla Firefox 3, http://www.cert.hr/documents.php?kw=internet+explorer+7+vs.+mozilla+firefox+3&lang=hr&cat=9, kolovoz 2009.
16. Vulnerability report: Microsoft Internet Explorer 7.x, http://secunia.com/advisories/product/12366/?task=statistics, kolovoz 2009.
17. Vulnerability Report: Mozilla Firefox 3.0.x, http://secunia.com/advisories/product/19089/?task=statistics, kolovoz 2009.
18. Security tips for Firefox users, http://www.squarefree.com/securitytips/users.html, kolovoz 2009.
19. A Virtually Secure Browser, http://www.sans.org/reading_room/whitepapers/hsoffice/a_virtually_secure_browser_33124, rujan 2009.
20. Sandbox (computer security), http://en.wikipedia.org/wiki/Sandbox_(computer_security), rujan 2009.
21. Application virtualization, http://en.wikipedia.org/wiki/Application_virtualization, rujan 2009.
22. Future of Secure Web Browsing, http://blog.unmaskparasites.com/2009/07/08/future-of-secure-web-browsing/, rujan 2009.
23. Gazelle Web Browser: Microsoft To Devise The Most Secure Web Browser?, http://www.spywareremove.com/security/gazelle-web-browser-microsoft-to-devise-the-most-secure-web-browser/, rujan 2009.
24. The Multi-Principal OS Construction oft he Gazelle Web Browser, http://research.microsoft.com/apps/pubs/default.aspx?id=79655, rujan 2009.
25. Google Chrome, http://www.google.com/chrome, rujan 2009.
26. Usporedba "sandbox" programskih alata, http://www.cert.hr/documents.php?kw=sandbox&lang=hr&cat=&x=0&y=0, rujan 2009.
27. VUPEN, http://www.vupen.com/english/advisories/2007/0075
28. The Multi-Principal OS Construction oft he Gazelle Web Browser, http://research.microsoft.com/pubs/79655/gazelle.pdf, rujan 2009.