CARNet
HRVATSKA AKADEMSKA I ISTRAŽIVAČKA MREŽA
CROATIAN ACADEMIC AND RESEARCH NETWORK

Web 2.0 – sigurnosni rizici

CCERT-PUBDOC-2008-11-245

+CERT.hr u suradnji s LSS

Sigurnosni problemi u računalnim programima i operativnim sustavima područje je na kojem CARNet CERT kontinuirano radi od 1996. godine.

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 boljem razumijevanju informacijske sigurnosti i poboljšanju sigurnosti Vašeg sustava.

CARNet CERT, www.cert.hr

Nacionalno središte za sigurnost računalnih mreža i sustava.

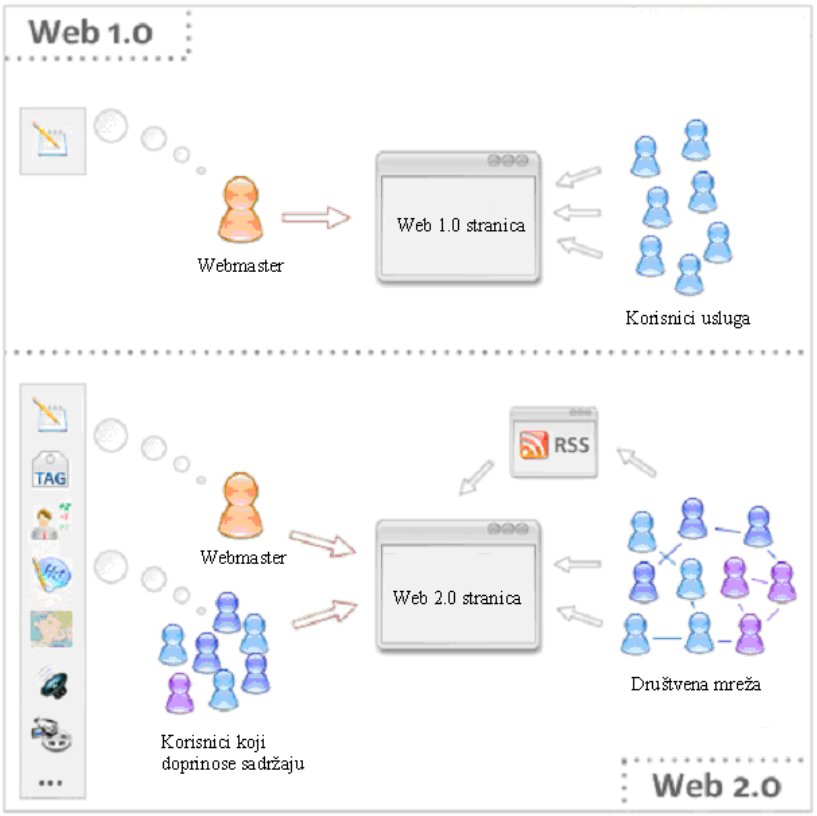
LS&S, www.LSS.hr

Laboratorij za sustave i signale pri Zavodu za elektroničke sustave i obradbu informacija Fakulteta elektrotehnike i računarstva Sveučilišta u Zagrebu bavi se informacijskom sigurnošću od 1995. godine.

Sadržaj

1. Uvod

Veliki i brzi razvoj Internetskih usluga zahtjeva ubrzan napredak tehnologija na kojima se one zasnivaju. U današnje vrijeme korisnici zahtijevaju pouzdane i stalno dostupne usluge, ali također i što bolju mogućnost suradnje, razmjene informacija i međusobne interakcije. Rane web tehnologije, poznate pod nazivom Web 1.0, nisu omogućavale navedene značajke. Jedan od osnovnih nedostataka bila je činjenica da su korisnici imali mogućnost samo pregleda sadržaja web stranica. Rezultat toga bio je relativno mali broj korisnika, kao i dosta manji broj Internet stranica (u usporedbi sa današnjim stanjem). Pojava Web 2.0 tehnologija uklonila je nedostatke ranijih tehnologija i donijela revoluciju u shvaćanju web usluga. Jedna od osnovnih novina je bilo samo shvaćanje web-a kao platforme na kojoj se izgrađuju usluge. Korisnicima se omogućilo aktivno sudjelovanje u kreiranju sadržaja web stranica. Osim toga, uvedene su razne nove funkcionalnosti kao što su RSS (eng. Really Simple Syndication) polja i označavanje (eng. tagging), a dolazi i do razvoja društvenih stranica (npr. Facebook, MySpace), blogova i foruma. Sve to dovelo je do velikog rasta broja korisnika Internet usluga. Opisane razlike između Web 1.0 i Web 2.0 tehnologija prikazane su na slici 1.



Slika 1. Razlika Web 1.0 i Web 2.0 tehnologija

Kako novo shvaćanje web usluga donosi mnoge pogodnosti korisnicima, ono također zahtjeva i primjenu novih tehnologija kao što je AJAX, Adobe Flash, Flex i sl. Osim toga, korisnici imaju veću slobodu i veća prava prilikom pristupa sadržaju web stranica. Sve to dovodi do velikih sigurnosnih problema s kojima se susreću sve Web 2.0 stranice. Statistički podaci o napadima na web stranice pokazuju da većina Web 2.0 stranica sadrži neku od ranjivosti koja napadačima omogućuje izvođenje nekog od raznih napada (npr. XSS napad). Zbog toga treba puno pažnje usmjeriti na pravilno programiranje Web 2.0 aplikacija, kao i pravilno korištenje svih novih tehnologija od strane krajnjih korisnika. Također, važnu ulogu igra i ispitivanje Web 2.0 aplikacija kako bi se otkrile moguće sigurnosne ranjivosti.

2. Web 2.0

Izraz Web 2.0 opisuje promjenu trendova u korištenju World Wide Web tehnologija pri izradi web stranica. Cilj mu je povećati kreativnost, sigurnu razmjenu informacija, suradnju i funkcionalnost. Temelji se na zamisli da se korisnicima omogući sudjelovanje u kreiranju sadržaja web stranica te podrazumijeva interaktivnu dvosmjernu komunikaciju između korisnika i računala te korisnika i drugih korisnika. Na taj način svaki korisnik postaje aktivni sudionik u komunikaciji.

Web 2.0 koncepti su doveli do razvoja i evolucije web usluga. Izraz je postao primjetan nakon prve „O'Reilly Media Web 2.0" konferencije 2004. godine. Iako pojam sugerira novu verziju World Wide Web tehnologija, on se ne odnosi na ažuriranje bilo koje tehničke specifikacije, nego na promjenu u načinima korištenja web usluga.

2.1. Definicije

Definicija Web-a 2.0:

> „Filozofija uzajamnog povećanja kolektivne inteligencije i dodane vrijednosti za svakog sudionika dinamičkim stvaranjem i dijeljenjem informacija."

Cilj Web 2.0 tehnologije:

> „Posložiti stvari tako da samousluga kupca i algoritamsko upravljanje podacima dosegnu cijeli web, sve do rubova, a ne samo do centra; do dugačkog repa, a ne samo do glave."

Tim O'Reilly termin definira kao:

> „Web 2.0 je poslovna revolucija u računalnoj industriji uzrokovana tretiranjem mreže kao platforme i nastojanje da se shvate pravila uspjeha na toj novoj platformi."

Web 2.0 tehnologije, kroz svoje brojne definicije, predstavljaju ideju veće međupovezanosti i interaktivnosti web sadržaja. Tim O'Reilly ocjenjuje Web 2.0 kao poslovno prihvaćanje weba kao platforme i korištenje njegovih prednosti, na primjer globalne publike. O'Reilly smatra da Eric Schmidt-ov slogan, "*don't fight the Internet*", obuhvaća bit Web 2.0 tehnologije - izgradnja aplikacija i usluga oko jedinstvene značajke Interneta, a ne očekivanje Interneta da bude prikladan kao platforma.

2.2. Značajke

Na otvaranju prve Web 2.0 konferencije O'Reilly i John Battelle saželi su ono što su oni vidjeli kao teme Web 2.0 tehnologije. Oni tvrde da je web postao platforma, s programskom podrškom iznad razine jednog uređaja i podacima koji čine pokretačku snagu. Web 2.0 tehnologija potiče nizanje poslovnih modela omogućenih udruživanjem sadržaja i usluga.

O'Reilly daje primjere tvrtke ili proizvoda koji utjelovljuju te principe u svom opisu hijerarhije web 2.0 stranica kroz četiri nivoa:

- **Aplikacije 0.** razine jednako funkcioniraju i lokalno na računalu (eng. *offline*) i preko interneta (eng. *online*). O'Reillyjevi primjeri su: MapQuest, Yahoo! Local i Google Maps.
- **Aplikacije 1.** razine funkcioniraju *offline*, ali vlastite značajke poboljšavaju se radom *online*. Primjeri takvih aplikacija su: Writely (sada Google Docs& Spreadsheets) i iTunes (zbog njegovog dijela s glazbenom trgovinom).
- **Aplikacije 2.** razine mogu funkcionirati *offline*, ali najveću korist za korisnika postižu *online* radom – kao primjer se navodi Flickr.
- **Aplikacije 3.** razine su one koje postoje samo na Internetu i svrhu dobivaju proporcionalno kako se ljudi njima služe. Za tu razinu O'Reilly daje primjere poput eBay, Craigslist, Wikipedia, del.icio.us, Skype, dodgeball i AdSense.

Aplikacije koje nisu bazirane na Web-u poput poruka elektroničke pošte (eng. e-mail), *instant-messaging* klijenata (omogućava komuniciranje između korisnika kratkim tekstualnim porukama) i telefona (VoIUP aplikacije) nisu obuhvaćene u hijerarhiji.

Osnovne karakteristike Web 2.0 su: otvorenost, sloboda i kolektivna inteligencija. Korisnici mogu koristiti aplikacije u potpunosti kroz web preglednik – dakle web se definira kao platforma, a korisnici imaju kontrolu nad određenim podacima na nekoj stranici.

2.3. Tehnologije

Dosta složena tehnološka infrastruktura Web 2.0 uključuje: poslužitelj, sadržaj, protokole za izmjenu poruka, preglednike temeljene na standardima sa dodacima i proširenjima te razne klijentske aplikacije.

Web 2.0 stranice obično uključuju neke od sljedećih tehnika:

1. **pretraži** (eng. Search): jednostavnost pronalaženja informacija putem ključnih riječi za pretraživanje,
2. **poveznice** (eng. Links): vodiči od važnih informacija na drugim ili istim web stranicama,
3. **stvaranje** (eng. Authoring): mogućnost nadopunjavanja sadržaja preko platforme koju je moguće stalno ažurirati,
4. **oznake** (eng. Tags): kategorizacija sadržaja po stvaranju jednostavnih oznaka (najčešće jedna riječ) kako bi se olakšalo pretraživanje,
5. **nastavci** (eng. Extensions): automatizacija nekog posla i pogađanje uzorka koristeći algoritme,
6. **signali** (eng. Signals): korištenje RSS (eng. Really Simple Syndication) tehnologija za obavještavanje korisnika o bilo kojoj promjeni sadržaja slanjem poruka elektroničke pošte.

Bogate tehnike za Internet aplikacije kao što su AJAX (eng. Asynchronous JavaScript and XML), Adobe Flash, Flex, Java, Silverlight i Curl su razvijene kako bi poboljšale iskustva korisnika u korištenju aplikacija temeljenim na preglednicima. Tehnologija dopušta web stranici da zatraži nadopunu za neki dio sadržaja i promjeni taj dio u pregledniku bez potrebe za osvježavanjem cijele stranice (podaci se osvježavaju u klijentskom pregledniku bez potrebe za interakcijom s web poslužiteljem).

Web 2.0 aplikacije se grade na postojećoj web arhitekturi, ali se puno više oslanjaju na programe kako bi osigurale bolju funkcionalnost. Dodatne funkcionalnosti koje pruža Web 2.0 ovisi o sposobnosti korisnika za rad s podacima pohranjenim na poslužiteljima. To se može ostvariti kroz modele u HTML stranici, putem skriptnih jezika kao što su Javascript/Ajax, Flash, Curl Applets i Java Applets. Sve metode koriste klijentsko računalo za smanjenje opterećenja poslužitelja i povećanje brzine aplikacija.

Česta zabluda je da se Web 2.0 odnosi na razne vizualne elemente dizajna jer su takvi dizajnerski elementi obično pronađeni na popularnim Web 2.0 stranicama. Još jedna od krivih pretpostavki u vezi Web 2.0. je potreba za korištenjem AJAX programskog jezika prilikom izrade web stranica. Ta pogreška vjerojatno dolazi zbog mnogo web 2.0 stranica koje se oslanjaju na AJAX ili pridružene DHTML efekte (Dynamic HTML je skupina tehnologija za izradu web stranica korištenjem *markup* jezika, skriptnih jezika i DOM). Dakle, dok je AJAX često potreban da bi Web 2.0 stranice dobro funkcionirale, obično nije i neophodan.

2.4. Tipovi Web 2.0 stranica

Osnovni tipovi Web 2.0 stranica su:

- **Društveno umrežavanje** je postalo sinonim za Web 2.0., jer označava aktivno sudjelovanje u virtualnim zajednicama, tj. skupinu korisnika zajedničkih interesa okuplja oko neke zajedničke internetske usluge. Najpopularniji socijalizacijski webovi (društvene stranice) u današnje vrijeme su Facebook (*www.facebook.com*) i MySpace (www.myspace.com).
- **Blog** je termin koji se odnosi na osobni dnevnik pisan na web-u s obrnuto-kronološki poredanim sadržajem (najnoviji sadržaj se nalazi na početku stranice). Na termin blog se nadovezuju termini: blogosfera – zajednica internetskih korisnika koji sudjeluju u stvaranju blogova, bloger – autor bloga te bloganje – učestalo pisanje vlastitog bloga i komentiranje tuđih blogova. Unutar blogova postoje mobilni blogovi, prilagođeni pisanju i čitanju putem mobitela, ručnih računala i slično te Podcast ili audioblogovi pohranjeni kao zvukovna datoteka.

- Bitno mjesto u društvenoj interakciji zauzimaju i **forumi** (javno diskutiranje o određenim temama putem Interneta) te *instant messaging* ili *chat* (razmjena poruka u realnom vremenu).
- **Folksonomija** ili kolaborativno označivanje (eng. tagging) je kolaboracijsko kategoriziranje sadržaja korištenjem oznaka (ključnih riječi u opisivanju bloga, profila, web stranica itd.).

2.5. Protokoli i API

Važno obilježje Web 2.0. stranica je mogućnost web konzorcije (eng. Web syndication) tj. dostupnost pojedinih dijelova web stranica drugim stranicama preko polja. Protokoli koji dozvoljavaju konzorciju uključuju RSS (eng. Really Simple Syndication), RDF i Atom, a svi su temeljeni na XML formatu. Specijalizirani protokoli kao što su FOAF (eng. Friend Of A Friend) i XFN (eng. XHTML Friends Network) proširuju funkcionalnosti sučelja i/ili dozvoljavaju krajnjim korisnicima interakciju bez centraliziranih web stranica.

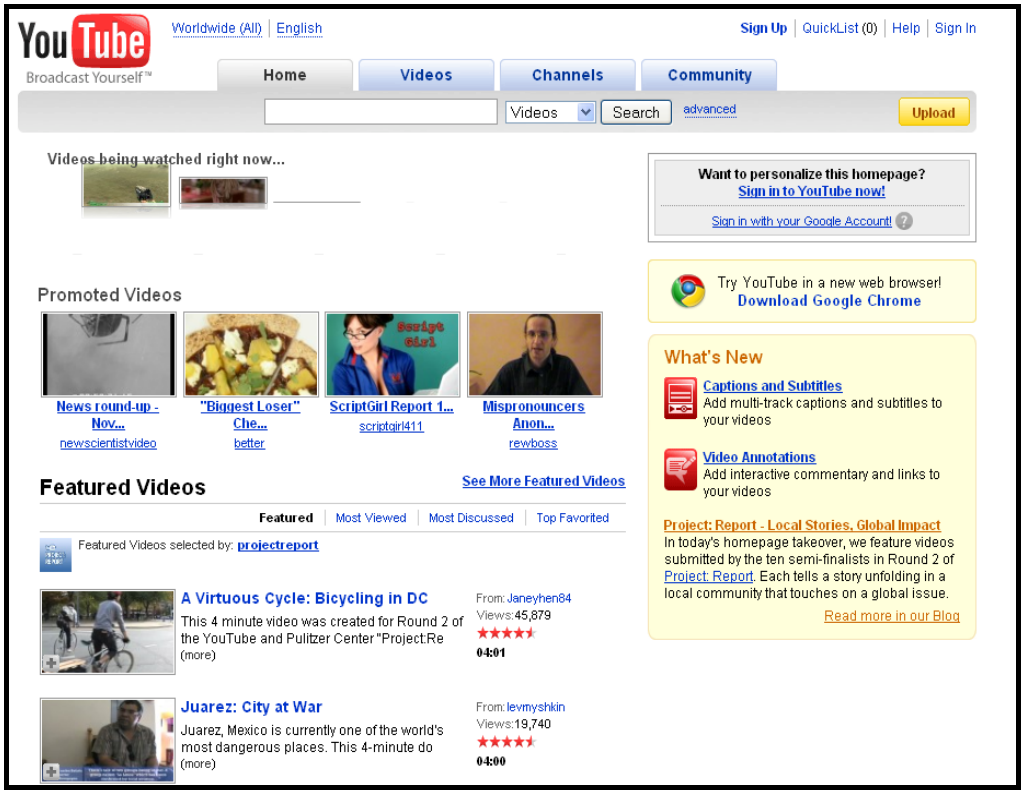
Web API (eng. Application Programming Interface) je skup sučelja, procedura, metoda, klasa ili protokola koji pruža operacijski sustav, knjižnica i/ili usluga kao potporu na zahtjeve računalnih programa. Jedna od osnovnih značajka Web 2.0 stranica je korištenje jednog od dva glavna pristupa Web API-ima:

1. **REST** (eng. Representational State Transfer): Web API koriste samo HTTP za interakciju s XML (eng. eXtensible Markup Language) ili JSON sadržajem. HTTP sadrži uniformno sučelje za pristup sredstvima koje se sastoji od URI adrese, metoda, zaglavlja i sadržaja. Najvažnije HTTP metode su POST, GET, PUT i DELETE. Na primjer, HTTP PUT se koristi za postavljanje vrijednosti resursa i može rezultirati kreiranjem ili zamjenom sadržaja (ovisno o potrebi).
2. **SOAP** (eng. Simple Object Access Protokol): uključuje mnogo složenije XML poruke i zahtjeve poslužitelju koji sadrže složene, ali unaprijed definirane upute koje poslužitelj treba slijediti. Laički primjer korištenja SOAP procedura je slanje SAOP poruka web uslugama na stranicama sa podacima za pretraživanje baze podataka u kojoj su spremljeni podaci o cijenama nekog proizvoda. Ovakav zahtjev kao odgovor prima dokument XML formata s dohvaćenim podacima (cijena, lokacija, raspoloživost i sl.).

2.6. Primjeri Web 2.0 stranica

Primjeri poznatijih Web 2.0 stranica:

- **YouTube** (www.youtube.com) je servis za objavljivanje, pregledavanje i razmjenu te komentiranje videozapisa. Kako prikazuje slika 2 web stranica sadrži mogućnost pretraživanja sadržaja, postavljanje novog sadržaja na web stranicu, a koristi se i označivanje (eng. tagging). Također, implementirane su poveznice koje vode do sadržaja na istim ili novim stranicama.



Slika 2. Web 2.0 stranica-YouTube

- **Yahoo!** (www.yahoo.com), prva velika uspješnica Interneta, rođen je kao katalog, ili kazalo poveznica, skup najboljih radova tisuća, a kasnije i milijuna korisnika web usluga.
- **Google**-ov (www.google.com) proboj u pretraživanju, koji ga je brzo učinio neosporno vodećim pretraživačem na tržištu, bio je PageRank, metoda korištenja strukture poveznica na Internetu, umjesto samih karakteristika dokumenta, da bi se omogućili bolji rezultati pretraživanja.
- **eBay**-ev (*www.ebay.com*) proizvod je kolektivna aktivnost svih njegovih korisnika; kao i sam web, eBay raste organski kao odaziv na aktivnost korisnika, a uloga tvrtke je biti posrednika u kojem se ta aktivnost korisnika može dogoditi.
- **Amazon** (www.amazon.com) prodaje iste proizvode kao i njima konkurentske tvrtke poput Barnesandnoble, a dobivaju i jednake opise proizvoda, slike naslovnica i sažetke od svojih dobavljača. Ali Amazon je napravio znanost korisničkog angažmana. Imaju red veličine više korisničkih recenzija, pozive za sudjelovanje na razne načine na svakoj stranici - i što je još važnije, koriste aktivnost korisnika da bi poboljšali rezultate pretraživanja.
- **Wikipedia** (*www.wikipedia.org*), *online* enciklopedija utemeljena na nevjerojatnoj zamisli da zapis može dodati bilo koji korisnik web-a i da ga može uređivati bilo koji drugi korisnik.
- Stranice kao što su **del.icio.us** (delicious.com) i **Flickr** (*www.flickr.com*), dvije tvrtke koje su u posljednje vrijeme privukle veliku pozornost, predvodile su koncept kolaborativne kategorizacije stranica koristeći slobodno odabrane ključne riječi, često nazivane *tagovi*. Flickr je kombinacija internetskog servisa za objavu digitalnih fotografija i socijalizacijskog web-a.

Slika 3 prikazuje najbolje web stranice u 30 kategorija prema popisu koji je moguće pronaći na web stranici: http://www.seomoz.org/web2.0.

Kategorija	1. Mjesto	2. Mjesto	3. Mjesto
Kazala	Del.icio.us	Stumbleupon	Furl
Knjige	Lulu	Biblio	VuFind
Suradničko pisanje	Google Docs	Writeboard	Thinkfree
Udaljeni pristup	Omnidrive	Fluxiom	Esnips
Edukacija	.Docstoc	Mango	Spanishpod
Zaposlenja	Standoutjobs	CareerBuilder	Monster
Događanja	Upcoming	Going	Confabb
Hrana	ImCooked	Urbanspoon	iFoods
Igre i zabava	Zango	Galaxiki	Doof
Vodiči i izvješća	Yelp	GoogleMaps	Citysearch
Zdravlje	RevolutionHealth	Peertrainer	Imedix
Wiki	PBWiki	Wetpaint	Wikispaces
Kartografske aplikacije	Frappr	Wayfaring	CommunityWalk
Karte	GoogleMaps	maps.live	GoogleEarth
Glazba	Last.fm	Pandora	Mog
Novosti i blog vodiči	GoogleBlogSearch	Bloglines	Technorati
Društveno umrežavanje (SN)	MothersClick	Tudiabetes	Imbee
Organizacija	Backpack	Zoho	Wufoo
Slike/digitalne slike	Flickr	Picnik	Picasa
Profesionalno umrežavanje	LinkedIn	Biznik	ProfessionalOnTheWeb
Pitanja i savjeti	answers.yahoo	Minti	Fixya
Gospodarstvo	Zillow	Rentomatic	HotPads
Maloprodaja	Threadless	Etsy	Stylehive
Pretraživanje	Tweetscan	Rollyo	50Matches
Glavna uporišta SN	Twitter	Facebook	Bebo
Društvene novosti	del.icio.us	Digg	Reddit
Sport	iStats	TeamSnap	Oobgolf
Putovanja	Farecast	Kayak	Boo
Video sadržaj	YouTube	bbc.co.uk	Metacafe
Web razvoj	pipes.yahoo	developer.yahoo	JQuery

Slika 3. Najbolje Web 2.0 stranice

3. Sigurnosni rizici

Iako Web 2.0 tehnologije donose mnoge pogodnosti u korištenju web usluga, također donose i brojne sigurnosne rizike, koji su uočeni kod većine Web 2.0 aplikacija. U nastavku dokumenta opisani su neki od poznatijih i novijih nedostataka uočenih u radu Web 2.0 aplikacija.

3.1. Web 2.0 i client-side napadi

Kako Web 2.0 čini web aplikacije više interaktivnim, omogućeno je lakše izvršavanje zlonamjernog programskog koda na klijentskom pregledniku. AJAX skupina tehnologija omogućava mnoge od tehnoloških napredaka koji se mogu naći u Web 2.0, što prvenstveno podrazumijeva suradnju i interakciju među korisnicima Interneta i davatelja usluga. U izvornom Web mediju korisnici su mogli

jednostavno pregledavati i preuzimati sadržaj web stranica. Web 2.0 omogućuje korisnicima da kreiraju sadržaj na web stranicama, što zahtjeva mnogo izvršavanja programskog koda u korisničkom pregledniku. Prilikom pregledavanja web stranice, korisnikov preglednik stvara zahtjeve i komunicira s Web aplikacijom. Taj scenarij napadaču daje priliku da ugradi zlonamjerni kod na proizvoljnu Web stranicu, kojeg će preglednik drugog korisnika automatski izvršiti. Zahvaljujući ovome, tehnike iskorištavanja poput umetanja HTML koda (eng. HTML code injection), umetanja SQL nizova (eng. SQL injection), XSS (eng. cross-site scripting) napada i krađe sjednica (eng. session hijacking) moguće je učinkovito primijeniti i na Web 2.0 stranice.

Kako zlonamjerni kod pokreće klijent (tj. preglednik na strani klijenta) mnoge prijetnje u ovoj kategoriji spadaju u napade na strani korisnika (eng. client-side attacks).

Prijetnje na korisničkoj strani mogu biti:

1. **Napadi svojstveni okolini socijalnog umrežavanja** - Kako društvene stranice dobivaju popularnost, iste sve više postaju mete napadača. Zlonamjerni napadači mogu instalirati zlonamjerne programe na socijalna web sjedišta kako bi automatizirali napade na korisnike. Zlonamjerni kod na ovim stranicama također može proslijediti program na druge zlonamjerne web stranice kako bi se izvršio *phishing* napad ili na neki drugi način oštetili korisnici (npr. razotkrivanje informacija).
2. ***Mashup* tehnologija** - koristi Web aplikacije za kombiniranje podataka/medija iz više izvora, lokacija i raznih programskih jezika. Uporaba te tehnologije otežava provjeru sigurnosti i integriteta programskog koda web aplikacija. Razlog tome je što *mashup* stranice moraju imati pristup podacima treće strane koristeći API-je te obraditi te podatke kako bi oni imali određeno značenje za korisnike.
3. **Polimorfno izrabljivanje (eng. Polymorphic exploitation)** - Kao odgovor na standardne regularne izraze i zaštitu potpisima koja identificira i detektira zlonamjerne radnje na mreži i poslužitelju, napadači dinamički mijenjaju napade svaki put kad potencijalna žrtva posjeti zlonamjernu web stranicu. Stvarajući jedinstveni način iskorištavanja za svaki korisnički zahtjev, napadači otežavaju zaštitu temeljenu na potpisima. Tradicionalne metode zaštite fokusiraju se na sprječavanju pokretanja zlonamjernih virusa na operacijskim sustavima korisnika. Međutim, mnogo je složenije zaštititi korisnike od pokretanja raznih zlonamjernih kodova u preglednicima.

3.2. Targeted messaging napad

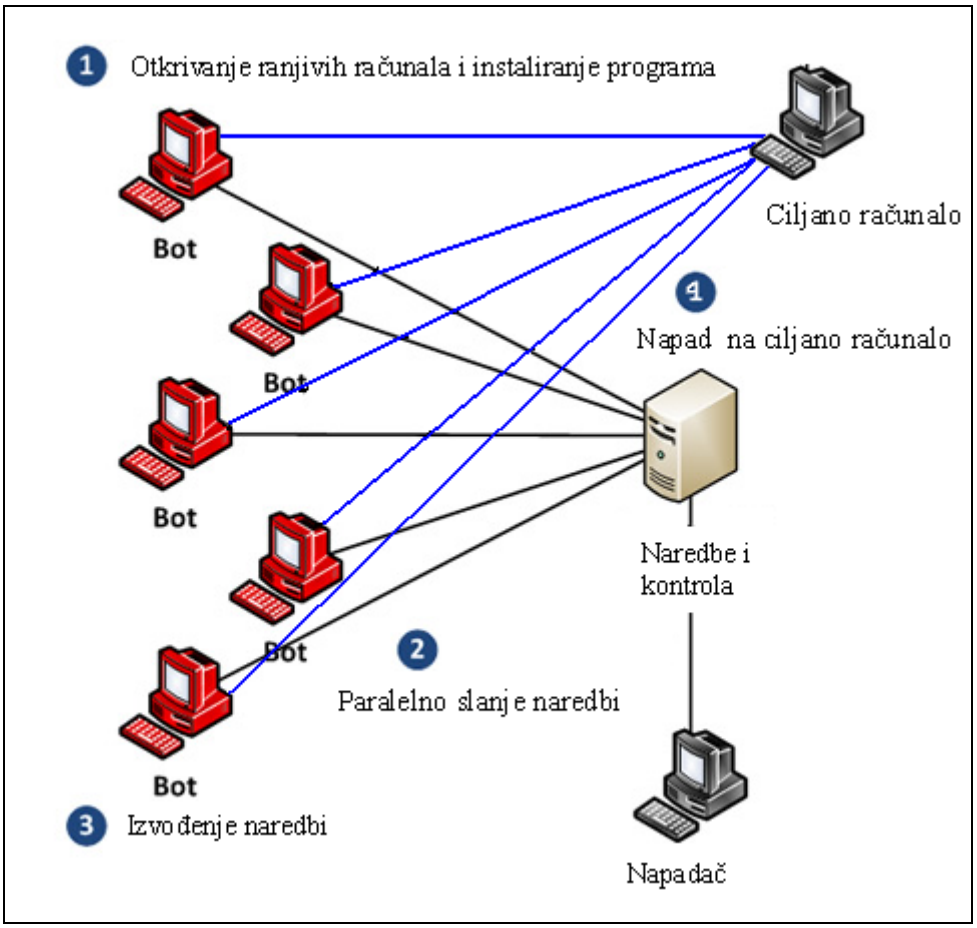
Umjesto fokusiranja na korporativnu infrastrukturu, *targeted messaging* napadi omogućavaju individualnim korisnicima da kradu autentifikacijske parametre (korisnička imena, lozinke), dozvole i privatne podatke krajnjih korisnika. Napad se ostvaruje putem poruka elektroničke pošte i preko trenutnih poruka (eng. instant messaging). Napredak tehnologija za rukovanje neželjenim (eng. *spam)* porukama te bolje obrazovanje korisnika dovelo je do razvoja novih tehnika kako bi se zaobišli filtri, vatrozid zaštite i sl. Također, pojavom Web 2.0 tehnologija i raznih novih tipova web stranica, pojavljuju se i novi oblici ovog napada.

Prijetnje vezane uz *targeted messaging* napad:

- ***Spam* sadržaj prikriven poslovnim sadržajem** – dok anti-spam programi uklanjaju neželjene poruke elektroničke pošte povećava se učestalost kreiranja neželjenih poruka koje izgledaju kao legitiman sadržaj poslovnog karaktera. Ovo uključuje PDF, XLS neželjene poruke, kao i eksperimentiranje s drugim formatima datoteka kako bi se neželjenim porukama dao izgled legitimnih (npr. legitimnih PDF datoteka).
- **Napadi ugrađen u trenutne poruke** (eng. instant messaging) **-** Napadač ugrađuje poveznice na zlonamjerna mjesta unutar inače legitimnih trenutačnih poruka. Na kraju IM konverzacije napadač uključuje poruku kako bi požurio posjetitelja da posjeti zlonamjernu poveznicu. Tada se provodi neki od *phishing* napada.
- **Neželjeni sadržaj u video sadržaju -** Sve veća popularnost mjesta za razmjenu videa dovodi do razvoja tehnika za napad na njih i kroz njih. Napadači mogu instalirati zlonamjerni kod unutar video sadržaja, koji će zatim utjecati na korisnike koji pristupaju video isječku.

3.3. Botnet

Botnet je skupina određenog broja računala kontroliranih od strane zlonamjernog poslužitelja ili gospodara. Scenarij stvaranja takve skupine računala prikazuje slika 4.



Slika 4. *Botnet*

Kontrola nad računalima postiže se isporukom i instaliranjem zlonamjernih programa. Zlonamjerni program se može isporučiti putem Trojana (eng. Trojans), poruka elektroničke pošte ili neovlaštenim trenutačnim porukama klijenta ili zaraženih web stranica. Jednom kada je instaliran program, on ostaje skriven kako bi se izbjegla detekcija antivirus i anti-spyware tehnologijama, a napadač ga aktivira prema želji (najčešće se koristi scenarij pokretanja napada s velikog broja bot računala odjednom kako bi se otežala obrana žrtve).

Kako su mnogi programeri užurbano usvojili Web 2.0 tehnologije, mnoge web stranice sadrže ranjivosti koje zlonamjerni korisnici mogu iskoristiti za stvaranje *Botnet* mreža. GTISC (eng. Georgia Tech Information Security Center) centar za sigurnost procjenjuje da je trenutno oko 15% računala stalno spojenih na Internet pod kontrolom nekog zlonamjernog poslužitelja.

Razlozi brzog rasta broja *botnet* mreža:

- mogućnost širenja kroz legitimne web stranice pomoću skrivenog zlonamjernog koda,
- poboljšanje specifikacija zlonamjernih programa za iskorištavanje i mehanizama za isporuku tih programa pa ih je teže detektirati,
- jednostavne akcije pregledavanja web sadržaja pokreću napad.

Korištenjem *botnet* mreža moguće je izvesti razne zlonamjerne aktivnosti:

- krađe podataka (brojeve kreditnih kartica, povjerljive informacije itd.),
- napad uskraćivanja usluga,
- isporuka neželjenog sadržaja (eng. spam),

- zavaravanje DNS poslužitelja (eng. DNS spoofing).

Sve više *botnet*-a formira se putem P2P (eng. Peer to Peer) mreže tradicionalni sustavi za detekciju i sprječavanje upada (IDS/IPS) ne bi mogli detektirati napad. Kada se otkrije ovakva zlonamjerna aktivnost i sazna izvor, tj. kontrolirajući poslužitelj, IDS/ISP sustavi blokiraju vezu s njim. Decentralizirano P2P okružje omogućuje napadaču da upravlja računalima *botnet mreže* s više računala, što mu omogućuje izbjegavanje postojećih sigurnosnih rješenja.

3.4. Zlonamjerni programi

Iskorištavanjem loše konfiguracije web poslužitelja, stranica društvenog umrežavanja i lažnih domena olakšava se isporuka zlonamjernih programa do korisničkih računala. Napadači koriste sve bolje tehnike socijalnog inženjeringa kako bi skrili zlonamjerni sadržaj te ga učinili što sličnijim legitimnom sadržaju. Također, napadači uče kako lokalizirati napad te postići što lakši upad u sustav.

Web 2.0 stranice društvenog umrežavanja poput MySpace, Facebook i dr. mogu lako navesti neoprezne korisnike na posjećivanje poveznica koje skrivaju zlonamjerne programe. Jedan takav primjer dogodio se 2006.g. kada se crv Spaceflash proširio preko „About me" dijela MySpace web stranice. Detaljniji opis samog napada dan je u odjeljku „Poznati napadi na Web 2.0 sustave u svijetu".

U kolovozu 2008. god. otkriveno je 28.940 različitih zlonamjernih i neželjenih programa na korisničkim računalima (http://www.kaspersky.com/news?id=207575678). Usporedbom rezultata s izvješćem za srpanj, primjećuje se rast za od 8.000 zlonamjernih programa (oko 40%).

3.5. Prijetnje temeljene na VoIP protokolu

Mobilni uređaji postaju potpuno novi instrumenti koji imaju mogućnost pristupa Internetu. Mnoge Web 2.0 aplikacije usmjeravaju svoj razvoj prema toj novoj domeni korisnika. VoIP (eng. Voice over Internet Protocol) tehnologije se također poboljšavaju te postaju rivali fiksnoj i mobilnoj komunikaciji u smislu pouzdanosti i kvalitete. Kako raste količina podataka s kojom rukuje Internet telefonija i mobilno računarstvo, korisnici ove tehnologije sve više postaju cilj zlonamjernih napadača.

Prijetnje VoIP tehnologijama:

- krađa podataka,
- glasovne prijevare,
- pokretanje proizvoljnog programskog koda,
- odbijanje usluga.

Veliki problem predstavlja mogućnost instaliranja zlonamjernih programa na mobitele, što bi ih moglo pretvoriti u *botnet* te tako ugroziti sigurnost jezgre same mreže.

4. Iskorištavanje propusta

Zlonamjerni korisnici mogu iskoristiti sigurnosne rizike kako bi izveli napad na korisnička računala ili poslužitelje. U nastavku dokumenta opisani su neki osnovni primjeri iskorištavanja propusta. Također, dan je popis nekoliko primjera iskorištavanja propusta kod poznatijih web stranica.

4.1. Teorijski primjeri

U nastavku odlomka dani su opisi osnovnih Web 2.0 napada.

4.1.1. XSS napad

XSS (eng. cross-site scripting) napad je jedan od mnogih načina iskorištavanja sigurnosnih rizika web stranica. Podrazumijeva umetanje posebno oblikovanog programskog koda unutar web stranica koji se pokreće u korisnikovom pregledniku prilikom prikaza te stranice.

Znači, prilikom pregleda stranice, AJAX se izvršava na korisničkoj strani što napadaču omogućava pokretanje zlonamjerno oblikovane skripte. Uloga napadača postaje zapravo samo "nagovaranje" korisnika da posjete određene web stranice iz svojih preglednika. Ako krajnji korisnik vjeruje aplikaciji, napadač iskorištava to da čini stvari koje mu inače nisu dozvoljene. Iako je XSS napad bio moguć i na ranijim tehnologijama, primjena AJAX-a daje mu novu dimenziju.

Upotreba XSS napada može ugroziti privatne informacije, manipulirati ili ukrasti kolačiće (eng. cookies), stvoriti pogrešne zahtjeve za korisnika ili omogućiti izvršavanje zlonamjernog programskog koda na sustavu krajnjeg korisnika. Ovaj se napad općenito smatra kao jedna od najčešćih ranjivosti aplikacijskog sloja, jer svaka stranica koja omogućuje korisniku unos nekih podataka može biti ranjiva na XSS napad.

4.1.2. XML poisoning

U mnogim Web 2.0 aplikacijama XML (eng. EXtensible Markup Language) paketi se razmjenjuju između poslužitelja i preglednika. XML blokovi koje koristi aplikacija dolaze iz AJAX klijenta. Prijenos XML blokova podataka ostavlja napadaču mogućnost pronalaženja načina na koji će "zatrovati" (eng. poison) te podatke. Često se primjenjuju tehnike koje omogućuju višestruku proizvodnju sličnih XML čvorova pomoću posebno oblikovanih rekurzivnih funkcija. Ovisno o računalu, ovi postupci mogu dovesti do stvaranja DoS (eng. Denial of Service) stanja na računalu, što naravno čini računalo neuporabljivo za rad krajnjem korisniku.

Ovaj napad je moguće izvesti kada web usluge koriste SOAP protokol za razmjenu XML poruka, što i je najčešći slučaj kod Web 2.0 tehnologija. Napadači mogu modificirati analizu SOAP poruke kako bi izložili integritet originalne poruke opasnosti.

Također, kod Web 2.0 tehnologija, velike razlike XML-a na aplikacijskom sloju otvaraju nove mogućnosti za izvođenje ovog napada.

4.1.3. Izvršavanje zlonamjernog AJAX koda

Krajnji korisnici ne mogu odrediti kada preglednik stvara XHR (eng. XMLHTTPRequest) objekt kako bi ostvario AJAX poziv. XHR je zapravo API kojeg koriste programski jezici preglednika za prijenos XML podataka između korisničke i poslužiteljske strane putem HTTP protokola. XHR koristi većina Web 2.0 aplikacija kako bi poboljšali svoje specifikacije.

Kada preglednik šalje AJAX poziv bilo kojoj web stranici, on šalje kolačiće za svaki zahtjev. Napadač može iskoristit AJAX programski kod na zlonamjernim stranicama kako bi dohvatio kolačiće koji mogu nositi važne informacije (lozinke, identifikacijske parametre i sl.). Ovo dovodi do proboja sigurnosti i curenja važnih informacija.

4.1.4. RSS / Atom injection

RSS polja su uobičajena mjesta za dijeljenje informacija na portalima i web aplikacijama. Polja koriste web aplikacije, a podatke šalju RSS klijentu. Zahvaljujući takvoj arhitekturi, napadač može umetnuti JavaScripts u RSS polja kako bi generirao napad na preglednik. Kada korisnik posjeti određenu web stranicu, učita RSS polja zajedno sa zlonamjernom skriptom koja se pokreće. Uloga zlonamjerne skripte može biti ili instaliranje zlonamjernog programa ili krađa kolačića.

Kako RSS i Atom skupovi podataka postaju sastavni dio web aplikacije, vrlo je važno filtrirati određene znakove na strani poslužitelja prije slanja podataka do krajnjeg korisnika.

RSS može sadržavati čisti tekst ili HTML kao sadržaj, ali ne postoji način na koji se može odrediti koji se koristi. Atom, s druge strane, pruža mehanizam za nedvosmisleno označavanje tipa sadržaja i omogućava raznolike vrste sadržaja (uključujući polje običnog teksta HTML, XHTML, XML, Base64-encoded binary) i reference na sadržaj (kao što su dokumenti, video sadržaj, audio zapisi i sl.).

Također, RSS elementi ne mogu se koristiti u drugim dokumentima, dok je Atom sintaksa specifično dizajnirana kako bi se omogućilo da se elementi ponovno koriste izvan konteksta nekog dokumenta.

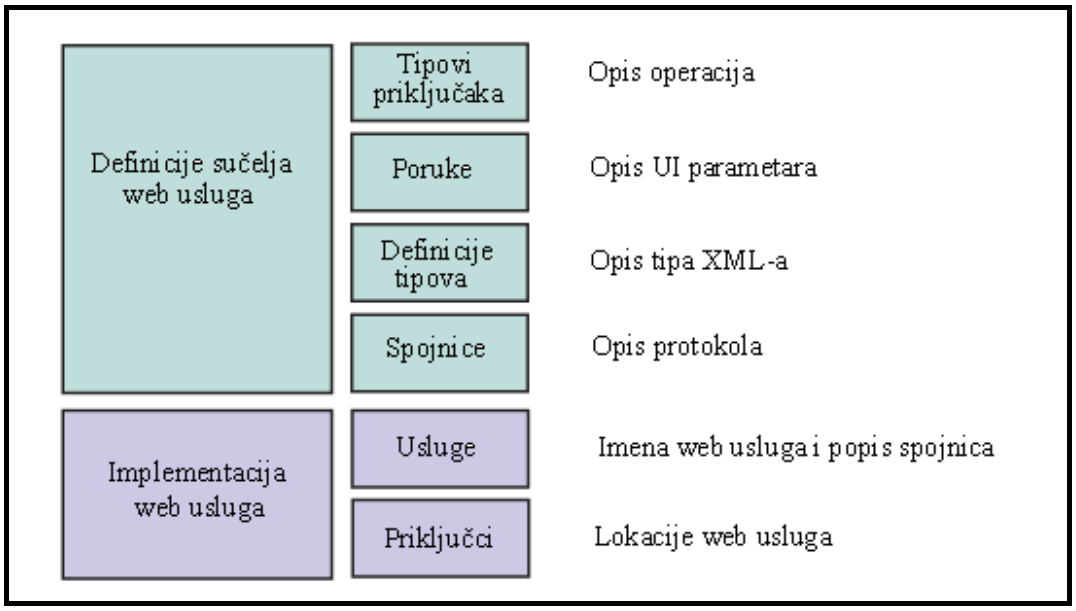
4.1.5. WSDL skeniranje

WSDL (Web Services Definition Language) je datoteka koja daje ključne informacije o tehnologijama, metodama, dodacima, itd. I pošiljatelj i primatelj poruka moraju imati isti opis usluga (WSDL datoteku). Pošiljatelj ga treba kako bi formirao poruku na ispravan način, a primatelj kako bi ispravno primio poruku.

WSDL datoteka sadrži:

1. definicije sučelja web usluga – sadrži prostor za imena (eng. namespaces) i elemente
2. implementacija web usluga – sadrži definiciju usluga i priključaka

Svaki dio dijeli se na manje cjeline, a strukturu datoteke moguće je vidjeti na slici 5.



Slika 5. Struktura WSDL datoteke

Budući da Web 2.0 tehnologije sadrže opis usluga pohranjen unutar WSDL datoteka, one predstavljaju vrlo osjetljivu informaciju koja napadaču može pomoći u definiranju metoda iskorištavanja propusta. Zbog toga, vrlo je važno ograničiti pristup WSDL datoteci ili ju držati potpuno skrivenu. Ako napadači dođu u posjed navedenih datoteka, njihovim skeniranjem oni mogu otkriti sigurnosne propuste određene aplikacije te odrediti najpogodniju metodu napada.

4.1.6. Validacija na strani klijenta u AJAX rutinama

Aplikacije temeljene na Web 2.0 tehnologijama koriste AJAX rutine kako bi obavile potrebne radnje na korisničkoj strani, kao što je provjera tipova podataka, sadržaja, polja i sl. Obično, uz provjere na strani korisnika potrebno je implementirati i provjere na strani poslužitelja. Većina programera se toga ne pridržava što dovodi do krivog mišljenja da se briga o provjeri parametara provodi u AJAX rutinama.

Napadač može zaobići provjeru AJAX parametara na strani korisnika slanjem HTTP POST i GET zahtjeva izravno samoj aplikaciji (SQL injection, LDAP injection i sl.) što može ugroziti sigurnost ključnih sredstava web aplikacije. Na primjer, pomoću osnovnih napada kao što je *SQL injection* moguće je razotkriti povjerljive podatke baze podataka.

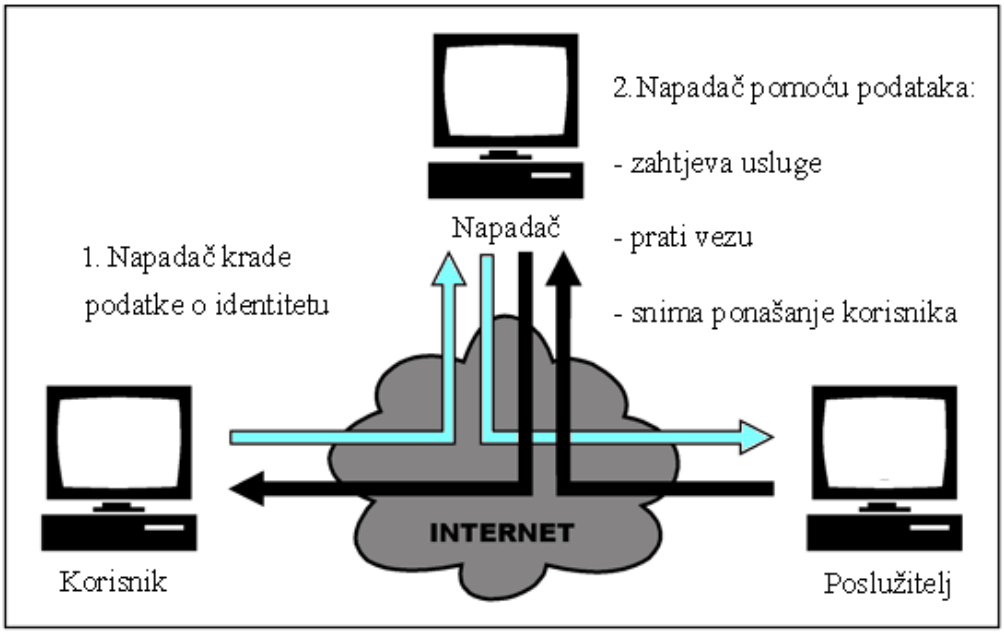
4.1.7. Web usmjeravanje

Sigurnosni protokoli web usluga sadrže *WS-Routing* (eng. Web Service routing) usluge koje provode provjeru sintakse za definiranje puta SOAP poruke. One predstavljaju ključnu ulogu u web usmjeravanju, a struktura poruke prikazana je na slici 6.

Usluge usmjeravanja omogućavaju putovanje SOAP poruka u posebnim nizovima iz raznih čvorova na Internetu. Prijenos tih poruka se odvija asinkrono preko različitih protokola (uključujući TCP i HTTP). Često tim čvorovima putuju kriptirane poruke.

Slika 6. Struktura SOAP poruke

Ugrožavanje sigurnosti bilo kojeg čvora u nizu rezultira mogućnošću pristupa SOAP porukama koje putuju između dvije krajnje točke. Napadač može izvesti MITM (eng. man-in-the-middle) napad da presretne SOAP poruke u prijenosu. Ovaj scenarij prikazan je na slici 7.



Slika 7. Man-in-the-middle napad

MITM može biti ozbiljan sigurnosni proboj SOAP poruka. Opisani napadi dobivaju sve više pažnje kako razvoj web aplikacija ide prema prihvaćanju novog okruženja Web usluga.

4.1.8. Manipulacija parametrima SOAP poruka

Web 2.0 usluge koriste informacije i varijable iz SOAP poruka, a tim parametrima moguće je manipulirati. Kako je već prikazano, SOAP poruke nose XML sadržaj te neki MIME sadržaj u dodacima, a služe za prijenos podataka između izvora i odredišta.

Otkrivajući informacije unutar SOAP poruka, napadač može započeti manipulaciju nekim čvorom te pokušati umetnuti zlonamjerni niz (SQL, LDAP, XPATH). Zahvaljujući tome, napadač može isprobati razne vrste napada kako bi uzrokovao zastoj korisničkog računala.

Nepravilna ili nedovoljna provjera ulaznih parametara u kodu web usluga ostavlja web usluge otvorenim za ugrožavanje sigurnosti.

4.1.9. XPATH injection u SOAP poruke

XPATH je jezik za ispitivanje i upravljanje XML dokumentima, a sličan je SQL upitima gdje je moguće dohvatiti određenu informaciju (parametre) ili cijele redove/stupce baze podataka. Ti XML dokumenti pružaju web aplikacijama podatke koji su im potrebni da bi ispravno funkcionirale. Mogućnosti provjere XPATH sintakse podržavaju mnogi jezici. Web aplikacije koriste velike XML dokumente i često te aplikacije primaju ulazne parametre od krajnjih korisnika i kreiraju XPATH niz. Napadač može iskoristiti ranjivost tih dijelova koda kako bi umetnuo XML dokument sa zlonamjernim XPATH nizom (eng. XPATH injection). Ako se umetnuti kod uspješno izvrši napadač može zaobići mehanizme autentifikacije ili prouzročiti gubitak određenih podataka. Jedini način sprječavanja ovog napada je osiguravanje pravilne provjere ulaznih parametara prije sintaksne analize XPATH nizova.

4.1.10. Manipulacija RIA thick client komponentama

RIA (eng. Rich Internet Applications) je korisničko sučelje koje koristi razvijene UI značajke poput Flash, ActiveX Controls ili Applets kao primarno sučelje za web aplikacije. Jedan od najvećih problema RIA sučelja predstavlja upravljanje sjednicama (eng. session management) jer se provodi u pregledniku.

U isto vrijeme, budući da su cjelovite binarne komponente preuzete na korisničku lokaciju, napadač može preokrenuti izgradnju binarnih datoteka i rastaviti kod. Kasnije ih je moguće sastaviti te zaobići autentifikacijsku logiku sadržanu u kodu, što predstavlja još jedan način napada na Web 2.0 okruženje.

4.2. Poznati napadi na Web 2.0 sustave u svijetu

Potkraj 2005. god. Samy Kamkar je kreirao crva koji se danas smatra prvim Web 2.0 crvom. Neuhvatljivi crv, kojeg je po sebi nazvao „**Samy worm**", nije mogao biti blokiran vatrozidom (eng. firewall), a proširio se tako brzo da su vlasnici web stranice MySpace privremeno korisnicima uskratili usluge stranice. Ovaj crv je predstavljao novu generaciju web napada sposobnih ugroziti Web 2.0 infrastrukturu. Također, ukazao je na potrebu poboljšanja sigurnosnih mehanizma u Web 2.0 svijetu. Kamkar je pregledom ograničenja postavljanja sadržaja na web stranicu MySpace otkrio sigurnosni nedostatak u kodu. Uočeni propust omogućio mu je umetanje većih naslova nego što je korisnicima bilo dopušteno. Daljnjim pregledom ograničenja postavljenih korisnicima otkrio je način na koji može dodati razne efekte na stranicu. Također, pronađeni nedostaci omogućili su mu uspostavu kontrole nad preglednikom svakog korisnika koji je posjetio njego profil. Takav korisnik je nesvjesno širio crva na sve korisnike koji bi pregledali njegov profil. Unutar samo nekoliko sati crv se proširio na skoro milijun korisnika web stranice MySpace. Razlog tako brzog širenja crva leži u dobroj međusobnoj povezanosti korisnika. Ovo je osnovni primjer neželjenih posljedica do kojih može doći ako se korisnicima dodijeli mogućnost podešavanja web postavki.

Godine 2006. otkriven je crv **Yamanner** pisan u JavaScript programskom jeziku, koji je iskorištavao sigurnosne nedostatke u Yahoo! uslugama za izmjenu elektroničke pošte. Zahvaljujući postojećim nedostacima crv je imao mogućnost prikupljanja adresa elektroničke pošte sa korisničkih računa i automatskog daljnjeg širenja. Skeniranjem poruka elektroničke pošte prikupljao je adrese poštanskih

sandučića koji su pripadali domenama @yahoo.com i @yahoogroups.com. Nakon skupljanja adresa crv bi proslijedio svoju kopiju putem poruka elektroničke pošte sa slijedećim karakteristikama:

```
From: Varies
Subject: New Graphic Site
Message Body: Note: forwarded message attached.
```

Slanje prikupljenih adresa na web stranicu www.av3.net/index.htm ostvareno je preusmjeravanjem korisničkog preglednika na navedeni URL.

Iste godine otkriven je još jedan crv nazvan **Spaceflash**, koji je također ugrozio sigurnost web stranice MySpace. Priliko pregleda „About me" stranice drugog korisnika koja sadrži Spaceflash crva, korisnik bi bio preusmjeren na posebno oblikovani URL :

```
editprofile.myspace.com/index.cfm?fuseaction=blog.view[REMOVED
```

koji sadrži posebno oblikovanu Macromedia Flash datoteku (nazvanu retrievecookie.swf). Zatim, posjećivanjem MySpace.com blog URL-a:

```
editprofile.myspace.com/index.cfm?fuseaction=user.HomeComments[REMOVED
```

dolazi do učitavanja JavaScript koda te njegova pokretanja. Opisani scenarij dovodi do brisanja korisnikove „About Me“ stranice te umetanja linije:

```
<EMBEDsrc="http://i105. photobucket.com/albums/m225/[REMOVED]">BY [REMOVED]
```

Napadač bi tu situaciju mogao iskoristiti za izvođenje zlonamjernih radnji i krađu osjetljivih podataka.

2007. g. **XSS** crv proširio se preko Google-ove Orkut socijalne web stranice preko poruka elektroničke pošte i Orkut poruka. Zahvaljujući postojanju XSS ranjivosti crv je zahvatio stotine tisuća korisničkih računala.

5. Statistički podaci o sigurnosti Web 2.0

Kako bi sa ukazalo na velike sigurnosne probleme Web 2.0 aplikacija, u nastavku je dan prikaz statističkih podataka o napadima na web stranice. Statistički podaci su preuzeti iz izvješća dviju agencija koje prate sigurnost web aplikacija: Sophos i Websense. Istraživanja ostalih renomiranih ustanova i/ili organizacija uglavnom se poklapaju sa Sophos-ovim i Websense-ovim istraživanjima.

5.1. Sophos izvješće

Web stranice:

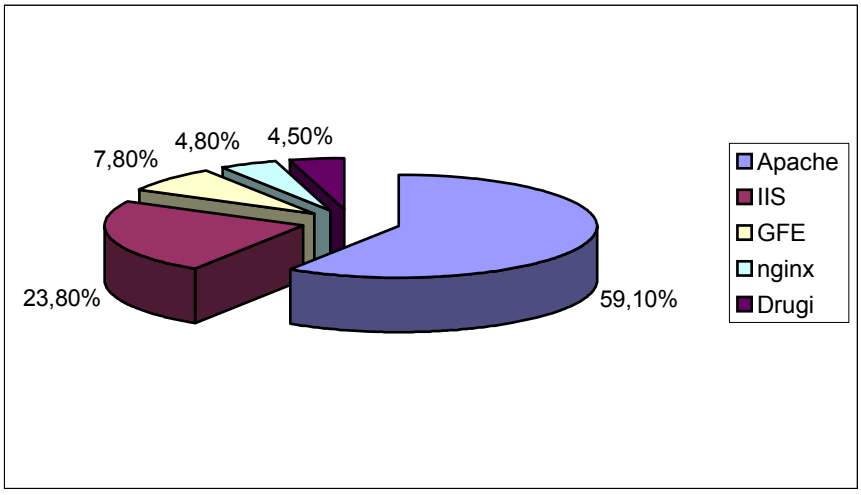
- 90% web stranica sa zlonamjernim kodom su legitimne web stranice,
- Broj različitih zlonamjernih prijetnji - preko 11 milijuna,
- Najveća zlonamjerna prijetnja - SQL injection,
- Učestalost novih prijetnji - otkrivanje 1 nove prijetnje svakih 5 sekundi,
- Poslužitelj sa najviše ranjivosti - blogger (Blogspot.com).

Poruke elektroničke pošte:

- Učestalost poruka koje sadrže zlonamjerni kod – 1 od 2.500,
- Neželjeni sadržaj unutar poslovnih poruka elektroničke pošte – 97%,
- Virusi koje se najčešće šire putem poruka elektroničke pošte: *Pushdo* (31%) i *Netsky* (20%).

Web poslužitelji:

Većina ranjivosti (skoro 60%) zahvaća Apache poslužitelje što pokazuje da sigurnost web stranica nije samo problem vezan uz Microsoft Windows nego i uz Linux operacijske sustave. Na slici 8 dan je postotak pojave ranjivosti na poslužiteljima.



Slika 8. Ranjivost web poslužitelja

5.2. Websense izvješće

Izvješće predstavlja rezultat istraživanja Wersense laboratorija korištenjem ThreatSeeker Network tehnologije tijekom prve polovice 2008.god.

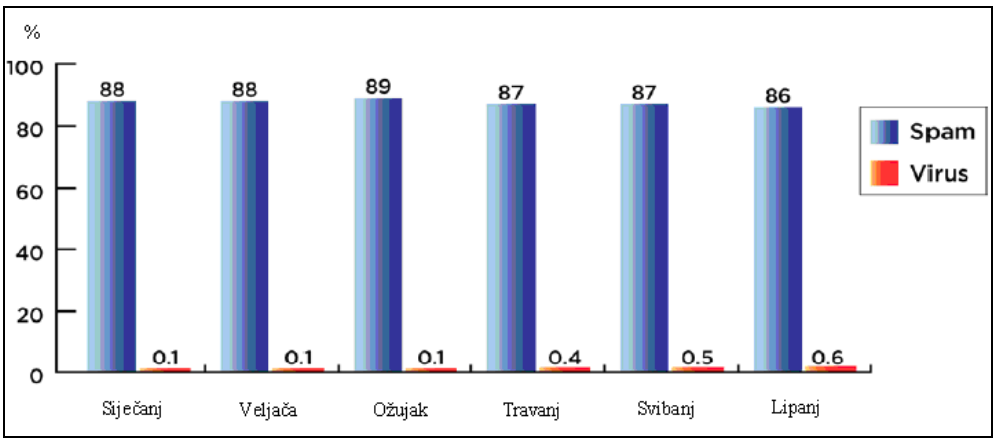
Sigurnost web stranica:

- 75 % web stranica koje sadrže zlonamjerni kod su legitimne stranice čija je sigurnost ugrožena,
- 60 % 100 najpopularnijih web stranica bile su uključene u zlonamjerne aktivnosti u prvoj polovici 2008.

Sigurnost vezana uz poruke elektroničke pošte:

- 87 % poruka elektroničke pošte spada u neželjeni sadržaj (*spam*),
- 76.5 % svih poruka elektroničke pošte sadrže poveznice na stranice sa zlonamjernim kodom,
- 9% poruka elektroničke pošte - *phishing* napad.

Slika 9 prikazuje postotak poruka koje spadaju u neželjene poruke te postotak poruka koje skrivaju viruse od ukupnog broja poruka za prvih 6 mjeseci 2008.god.



Slika 9. Postotak *spam* poruka i virusa

Sigurnost podataka:

- 29 % web napada uključuje krađu podataka,
- 46 % napada krađe podataka počini se preko interneta.

6. Savjeti za programiranje Web 2.0 aplikacija

U nastavku dokumenta dani su neki osnovni principi programiranja i testiranja Web 2.0 aplikacija kojih se treba pridržavati prilikom njihove izgradnje.

6.1. Programiranje

Programiranje Web 2.0 aplikacija zahtjeva poznavanje različitih tehnologija kao i njihovu primjenu da bi se osigurala potrebna razina sigurnosti. Neki osnovni principi kojih se treba pridržavati, kao i osnovne tehnologije, kratko su opisani u nastavku.

6.1.1. OOP

OOP (eng. Object-oriented Programming) su programske paradigme koje koriste objekte i njihovo međudjelovanje u svrhu dizajniranja aplikacija i računalskih programa. Programske tehnike mogu uključiti značajke poput:

- enkapsulacije (eng. encapsulation) – tehnika kojoj je svrha skrivanje podataka,
- modularnog programiranja (eng. modular programming) – dijeljenje računalnog programa u zasebne module koji se u funkcionalnosti preklapaju što manje,
- polimorfizma (eng. polymorphism) – sposobnost obrade objekta na razne načine ovisno o tipu podataka ili klasi,
- nasljeđivanja (eng. inheritance) – način formiranja novih klasa na temelju ranije formiranih klasa pri čemu nove klase preuzimaju atribute i ponašanje ranije definiranih klasa.

Razvoj OOP označava veliku promjenu u računarstvu jer podrazumijeva oblikovanje programskog koda koje je moguće ponovo upotrijebiti u raznim programima. Iako se OOP nije široko koristio do 90-ih godina, danas većina modernih programskih jezika ima podršku za OOP. Prilikom programiranja Web 2.0 aplikacija preporuča se korištenje OOP upravo zbog navedenih značajki.

6.1.2. AJAX

AJAX (Asynchronous Javascript and XML) je skupina tehnika koje služe za kreiranje web aplikacija. Zahvaljujući AJAX tehnologijama web aplikacije mogu dohvatiti podatke s poslužitelja asinkrono u pozadini (bez potrebe za promjenom u ponašanju stranice). Preuzimanje podataka se obavlja preko *XMLHttpRequest* objekata.

Također, prednost korištenja AJAX-a je smanjenje komunikacije klijenata i poslužitelja jer se zahtjev za skriptama šalje samo jednom. Zahvaljujući AJAX-u web aplikacije mogu zahtijevati ažuriranje samo onog sadržaja koji im je potreban što znatno smanjuje vrijeme potrebno za prikaz stranica.

Korištenje opisanih tehnologija pri programiranju Web 2.0 aplikacija smanjuje rad na poslužitelju te povećava brzinu samih aplikacija.

6.1.3. Relacijske baze podataka

Relacijska baza podataka jest skup relacija definiranih relacijskom shemom baze podataka. Relacijska shema baze podataka jest skup različitih relacijskih shema koje opisuje građu relacije, a sastoje se od naziva relacije i konačnog skupa atributa.

Ciljevi koji se postavljaju nad bazom podataka:

- nezavisnost podataka,
- konzistentno semantičko postupanje s podacima,
- eliminacija redundancije (zalihosti) podataka,
- skupovno orijentiran jezik za obradu podataka,
- razni načini opisa i obrade jednostavnih i kompleksnih podataka.

Relacijske baze podataka su jednostavne za korištenje, a pri tome nije potrebno preveliko znanje programiranja što može biti dodatni poticaj za njihovu uporabu prilikom izgradnje Web 2.0 aplikacija.

6.1.4. Skriptni jezici

Postoji iznimno velik broj skriptnih jezika, a neki poznatiji među njima su: Javascript, Python, Ruby, PHP itd. Obično se koriste za povezivanje više samostalnih programa u cjelinu, intenzivnu obradu teksta, manipulaciju datotečnim sustavom i sl.

Prednosti korištenja skriptnih jezika su:

- kraći programi,
- brži razvoj – korištenje biblioteke s gotovim komponentama,
- ne postoje deklaracije varijabli,
- mnoštvo alata za povezivanje programa,
- mnoštvo alata za obradu teksta,
- jednostavna izgradnja korisničkog sučelja (GUI),

Skriptni jezici se u većini slučajeva interpretiraju (iako se mogu i prevoditi) te time omogućuju naizmjenično pisanje i testiranje linije po linije koda, čime se dobiva na dinamičnosti programiranja. Opisane pogodnosti uvelike pridonose funkcionalnosti Web 2.0 aplikacija pa se preporuča uporaba skriptnih jezika.

6.1.5. HTTP protokol

HTTP (engl. HyperText Transfer Protocol) protokol je skup pravila za prijenos informacija na Web-u. Osnovna namjena ovog protokola je omogućavanje objavljivanja i prezentacije HTML dokumenata, tj. web stranica. Zasniva se na zahtjevima i odgovorima, a služi za uspostavu komunikacije između korisnika i poslužitelja.

HTTP protokol se razlikuje od ostalih TCP protokola kao što je npr. FTP (eng. File Transfer Protocol) po tome što se konekcija i komunikacija sa poslužiteljem prekida odmah nakon izvršenja zahtjeva klijenta (isporučenog paketa traženih podataka). Ova karakteristika HTTP protokola povremeno stvara probleme web dizajnerima, s obzirom da nedostatak "konstante konekcije" s poslužiteljem moraju nadoknaditi uporabom drugih metoda za očuvanjem "korisničkog stanja". Jedna od tih metoda uključuje uporabu HTTP kolačića.

Prilikom programiranja Web 2.0 aplikacija vrlo je važno upoznati funkcionalnosti i način rada HTTP protokola.

6.1.6. OSS

OSS (eng. Open Source Software) podrazumijeva programe koji su izdani bez zabrane pregleda izvornog koda (eng. source code) samog proizvoda te imaju licencu neke organizacije. Programi otvorenog koda potiču korisnike da distribuiraju besplatno proizvod i naprave izmjene prema vlastitim potrebama. Također, mogućnost mijenjanja, prepravljanja koda osigurava poboljšavanje karakteristika proizvoda.

Međutim, kako je izvorni programski kod dostupan svim korisnicima (dakle i zlonamjernima) uvid u iste omogućava lakšu detekciju nedostataka u dizajnu pa je potrebno voditi i o tome računa. Redovito osvježavanje OOS programskih rješenja izdanim zakrpama od iznimne je važnosti za sigurnost cijelog sustava.

6.1.7. DOM

DOM (eng. Document Object Model) je skupina pravila za dohvat podataka iz XML ili HTML dokumenta. Ti dokumenti mogu imati sadržaj i podatke skrivene unutar objekata kako bi se osigurala kontrola mogućnosti manipulacije dokumentom pojedinim korisnicima. DOM model specificira provođenje kontrole nad dokumentima.

Prednosti korištenja DOM modela:

- Svaki HTML ili XML element je moguće individualno adresirati.
- Specifikacija je neovisna o programskom jeziku (eng. language-independent) i ako je to moguće, opisana pomoću IDL jezika (eng. Interface Definition Language).
- Sučelje je implementirano u Java programskom jeziku ili ECMAScript, jeziku temeljenom na JavaScript i JScript.

Korištenje DOM modela i razumijevanja rada s njim vrlo je važno prilikom izgradnje Web 2.0 aplikacija.

6.1.8. Kriptiranje i upravljanje digitalnim pravima

Kriptiranje je proces transformacije informacija korištenjem nekog algoritma kako bi bile razumljive samo korisnicima kojima su namijenjene, tj. onima koji poznaju ključ kriptiranja. Postoje mnogi algoritmi kriptiranja, a neki od poznatijih su: RSA, DES/3DES i dr.

Upravljanje digitalnim pravima (eng. Digital Rights Management - DRM) je izraz koji se odnosi na tehnologiju kontrole pristupa digitalnom mediju kako bi se ograničilo korištenje i kopiranje informacija te pretvorba u drugi format.

Korištenje ovih tehnologija sprječava krađu intelektualnog vlasništva, što igra vrlo važnu ulogu u Web 2.0 aplikacijama.

6.1.9. Platforme

Platforma opisuje neku vrstu sklopovske arhitekture ili programskog okruženja koje omogućuje pokretanje programa. Obično uključuje: arhitekturu računala, programski jezik i biblioteke te grafičko sučelje.

Odabir platforme je ključni element u razvoju programa, jer o tome ovisi rad programa (aplikacije). Sam Web 2.0 je dizajniran kako bi potaknuo trend koji podrazumijeva cijeli web kao platformu.

6.1.10. Stylesheets (CSS i XSLT)

CSS (eng. Cascading Style Sheets) je jezik za opis prikazivanja dokumenta (eng. stylesheet) pisanog u jeziku koji koristi skupine uputa o prikazu sadržaja (eng. markup). Korisnici web stranice mogu ga koristiti lokalno da bi definirali boju, veličinu slova i druge aspekte prikazanog dokumenta.

Prednosti korištenja CSS su:

- poboljšanje pristupa sadržaju,
- pružanje bolje fleksibilnosti i kontrole u specificiranju karakteristika,
- smanjenje kompleksnosti i ponavljanja strukovnih elemenata.

XSLT (eng. Extensible Stylesheet Language Transformations) je jezik koji se koristi za transformiranje XML dokumenata u druge XML dokumente. Zahvaljujući tome, izmjene se obavljaju na konačnom dokumentu (koji nastaje kreiranjem na temelju sadržaja dokumenta koji se želi mijenjati).

Uporaba opisanih jezika omogućava korisnicima veću fleksibilnost i upravljanje dijelovima sadržaja na web stranicama, što je jedna od važnih značajki Web 2.0 tehnologija.

6.2. Testiranje

Budući da su Web 2.0 tehnologije donose potpuno novu paradigmu razvoja i isporuke, treba ažurirati tehnologiju razvoja i strategiju testiranja Web 2.0 aplikacija.

Slijede neki prijedlozi kojih se treba pridržavati:

1. Kako bi se izgradio kvalitetan proizvod nije dovoljno samo raditi izmjene tijekom razvoja, potrebno je ugraditi posebne mjere koje je moguće pratiti tijekom i nakon razvoja.
2. Nije dovoljno samo razumjeti koncept *third-party* web usluge nego testirati njihove usluge i podatke.
3. Potrebno je upoznati zahtjeve korisnika kao i tehnologije koje oni koriste.
4. Potrebno je ispitati rad na svakom operacijskom sustavu koji korisnik može koristiti.
5. Treba sačuvati zapise svih testiranja na platformama kako bi uvijek bilo poznato koji su postojeći problemi te kako ih ispraviti.
6. Potrebno je zapisivati sve aktivnosti u pregledniku tijekom testiranja i rada.
7. Treba razumjeti vezu između izvedbe (eng. performance) i percepcije (eng. perception).
8. Također, potrebno je uključiti preglednik u svoje buduće integracijske procese.
9. Treba provoditi "on-demand" testiranje (program kao usluga).
10. I na kraju, treba ponoviti testiranje kako se web stranica razvija.

6.2.1. OWASP vodič za testiranje

OWASP (eng. Open Web Application Security Project) zaklada 2007.g. izdala je inačicu 2.0 vodiča za testiranje poz nazivom „OWASP TESTING GUIDE“. Vodič je namijenjen:

- Programerima – kako bi potvrdili izgradnju sigurnog koda,
- Osobama zaduženim za testiranje programa – kako bi proširili skupinu testova koje koriste za aplikacije,
- Stručnjacima za sigurnost – kako bi ga kombinirali s drugim tehnikama u provjeri sigurnosnih nedostataka aplikacija.

Vodič opisuje principe testiranja kao i same tehnike testiranja. Također, prikazane su mjere koje je potrebno napraviti u svakoj fazi razvoja aplikacije. Faze razvoja aplikacije su:

1. prije početka razvoja,
2. tijekom definiranja i dizajna,
3. tijekom razvoja (eng. Development),
4. tijekom implementacije (eng.Deployment),
5. tijekom održavanja.

Zatim slijedi opis testiranja raznih načina na koji napadači mogu ugroziti ranjive sustave, otkriti osjetljive podatke ili izvesti neki od brojnih napada. Tu se nalazi detaljan opis testiranja pogrešaka u kodu, autentifikacije, upravljanja sjednicama, SSL/TLS i „Brute Force" testiranja te testiranja napada umetanjem nizova i raznih drugih metoda napada.

Na kraju dokumenta nalazi se detaljan opis načina pisanja izvještaja o ranjivostima, tj. pravilne procjene značaja pojedinog rizika na sigurnost sustava.

U dodatku dokumenta mogu se pronaći preporučeni alati za testiranje, poveznice na neke dodatne sadržaje o testiranju i ranjivostima aplikacija te „fuzz vectors" koji se mogu koristiti uz neke *fuzzer* alate (WebScarab, JBroFuzz, WSFuzzer).

7. Zaštita

7.1. Poslužitelj i vatrozid

Kako bi se osigurala sigurnost podataka na strani poslužitelja potrebno je provoditi provjeru podataka (eng. server-side data validation). Taj pojam predstavlja jedan od osnovnih i najvećih problema u sigurnosti web aplikacija. Osim potrebe za provjerom podataka na strani poslužitelja, postoji potreba za provjerom podataka na korisničkoj strani.

Vatrozid (eng. firewall) je integrirana skupina sigurnosnih mjera dizajniranih kako bi se spriječio neovlašteni pristup računalnom mrežnom sustavu. Također, predstavlja mrežni uređaj ili skupinu uređaja konfiguriranih da spriječe, odbiju, šifriraju, dešifriraju ili proslijede sav promet između različitih domena. Rad se temelji na skupini pravila i kriterija. Osnovna funkcija vatrozida je filtriranje podataka te odbacivanje nepotrebnih paketa.

7.2. WA skeneri

WA skeneri (eng. web application security scanners) su programi koji komuniciraju sa web aplikacijama kako bi identificirali potencijalne sigurnosne nedostatke u web aplikacijama. Oni ne pristupaju izvornom koda nego detektiraju ranjivosti izvođenjem napada.

Imaju mogućnost otkrivanja sljedećih ranjivosti:

- provjeru ulaznih/izlaznih parametara (XSS napad, SQL injection),
- probleme specifične za neku aplikaciju,
- pogreške u konfiguraciji poslužitelja.

Prednosti:

- Alati mogu detektirati ranjivosti krajnje inačice proizvoda prije puštanja u korištenje.
- Alat simulira napadača izvodeći napad te uspoređuje koji se rezultati ne slažu s očekivanim rezultatima.
- Alat nije ovisan o jeziku programiranja pa može skenirati JSP, PHP i sl.

Nedostaci i ograničenja:

- Alati ne pokrivaju u potpunosti kod aplikacije i samu aplikaciju.
- Vrlo je teško otkriti lokalnu ranjivost poput korištenja slabih kriptografskih algoritama, curenja informacija i sl.
- Alati ne mogu implementirati sve vrste napada tipične za neku ranjivost (obično obavljaju samo klasične napade).
- Korištenje alata zahtjeva razumijevanje ponašanja aplikacija s JavaScript, Flash i sličnim sadržajem.

7.3. Ostali elementi zaštite

Stručnjaci savjetuju bolje praćenje akcija (eng. logging) na web stranicama. Zahvaljujući tim zapisima moguće je primijetiti zlonamjernu radnju nekog korisnika. Postoje razni besplatni alati koji omogućavaju praćenje akcija napisani u programskom jeziku C#:

- log4net:
  - pruža visoke performanse te omogućuje veliku fleksibilnost i proširivost,
  - ima mogućnost praćenja višestrukih prijava,
  - sadrži hijerarhijsku arhitekturu.
- Nlog:
  - biblioteka za praćenje akcija,
  - jednostavna za konfiguriranje i korištenje.
- Common.Logging:
  - biblioteka koja omogućava odabir specifičnih implementacija za praćenja akcija.

Također, potrebno je primijeniti postojeće tehnologije za podizanje zaštite na viši nivo (poput enkripcije podataka – osigurava se tajnost podataka čak i ako oni dođu u posjed korisnika kojima nisu namijenjeni).

Još jedan važan element zaštite je rukovanje pravima pristupa (jer omogućava zaštitu podataka). Korisnicama web usluga treba ograničiti pristup podacima koji nisu njima namijenjeni.

Osim toga korisnik može isključiti pokretanje skripti koje dolaze iz nepovjerljivih izvora u svom pregledniku jer dosta web aplikacija može raditi bez potrebe za tim. Na taj način, čak i ako je zlonamjerno oblikovana skripta umetnuta na web stranicu korisnik neće biti ranjiv na XSS napad.

8. Zaključak

Od same pojave, Web 2.0 tehnologije vrlo se brzo šire, što dokazuje ogroman broj Web 2.0 stranica koje se dnevno pokrenu. Razlog tome leži u poticanju korisnika da aktivno sudjeluju u kreiranju sadržaja samih web stranica te tijekom korištenja daju svoj prilog nekom Web sadržaju ili aplikaciji. Također, vrlo bogat novi dizajn, koji omogućuju razvoj i primjena novih tehnologija, postaje privlačan velikom broju korisnika. Nadalje, tu su neki aspekti društvenog umrežavanja te kvalitetnije grafičko uređenje nego na Web 1.0 tehnologijama.

Kako je prethodno pokazano, uz brojne pogodnosti ova tehnologija korisnicima donosi i mnoge sigurnosne rizike. Pravilnim programiranjem moguće je spriječiti brojne ranjivosti koje su svakodnevna pojava većine stranica. Također, vrlo ja važno definirati prava pristupa te ograničenja korisnicima kako oni ne bi mogli zlouporabiti svoje ovlasti.

Što se tiče zaštite na strani korisnika, preporuča se provjera svih ulaznih podataka, kao i implementacija određenih zaštitnih mehanizama za filtriranje prometa (npr. vatrozida). Osim toga, isključivanjem pokretanja skripti koje dolaze iz nepovjerljivih izvora u vlastitom pregledniku, korisnik može spriječiti izvođenje XSS napada (što se pokazalo da predstavlja veliku većinu zabilježenih sigurnosnih incidenata).

9. Literatura

[1] Web 2.0, http://hr.wikipedia.org/wiki/Web_2.0, studeni 2008.

[2] Web 2.0, http://en.wikipedia.org/wiki/Web_2.0, studeni 2008.

[3] Web 2.0, http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html, studeni 2008.

[4] 174 Web 2.0 Sites in 41 Categories, http://www.seomoz.org/web2.0, studeni 2008.

[5] Emerging Cyber Threats Report for 2009, http://www.gtiscsecuritysummit.com/pdf/CyberThreatsReport2009.pdf

[6] Emerging Cyber Threats Report for 2008, http://www.gtisc.gatech.edu/pdf/GTISC%20Cyber%20Threats%20Report.pdf

[7] Security, Privacy and Policy in a Web 2.0 World, http://robotbrother.blogspot.com/2007/08/security-privacy-and-policy-in-web-20.html, studeni 2008.

[8] Top 10 Web 2,0 Attack Vectors, http://www.net-security.org/article.php?id=949, studeni 2008.

[9] Samy worm, http://en.wikipedia.org/wiki/Samy_(XSS), studeni 2008.

[10] Websense izvješće, http://www.websense.com/securitylabs/docs/WSL_Report_1H08_FINAL.pdf

[11] Koncepti za programiranje Web 2.0 aplikacija, http://otherlibrarian.wordpress.com/2007/09/06/under-the-hood-of-web-20-the-top-ten-programming-concepts-for-librarians-to-understand/, studeni 2008.

[12] Testiranje i izvedba Web 2.0 aplikacija, http://searchsoftwarequality.techtarget.com/news/article/0,289142,sid92_gci1260130,00.html, studeni 2008.

[13] OWASP Foundation, OWASP TESTING GUIDE v2.0, http://www.lulu.com/items/volume_63/4037000/4037522/1/print/OWASP_Testing_Guide_v2_for_print.doc.pdf, 2007.

[14] WA skeneri, http://en.wikipedia.org/wiki/Web_Application_Security_Scanner, studeni 2008.

[15] Vatrozid, http://en.wikipedia.org/wiki/Firewall, studeni 2008.

[16] Open Source Logging Tools in C#, http://csharp-source.net/open-source/logging, studeni 2008.