Centar Informacijske Sigurnosti

Web tripwires

siječanj 2012.

CIS-DOC-2012-01-037

Upozorenje

Podaci, informacije, tvrdnje i stavovi navedeni u ovom dokumentu nastali su dobrom namjerom i dobrom voljom te profesionalnim radom CIS-ovih stručnjaka, a temelje se na njihovom znanju i petnaestak godina iskustva u radu u informacijskoj sigurnosti. Namjera je da budu točni, precizni, aktualni, potpuni i nepristrani.

Ipak, oni su dani samo kao izvor informacija i CIS ne snosi nikakvu izravnu ili posrednu odgovornost za bilo kakve posljedice nastale korištenjem podataka iz ovog dokumenta.

Ukoliko primijetite bilo kakve netočnosti, krive podatke ili pogreške u ovom dokumentu, ili imate potrebu komentirati sadržaj molimo Vas da to javite elektroničkom poštom na adresu info@CIS.hr.

O CIS-u

CIS izrađuje pregledne dokumente (eng. white paper) na teme iz područja informacijske sigurnosti koji će biti korisni zainteresiranoj javnosti, a u svrhu **podizanje njezine svijesti o informacijskoj sigurnosti** i **sposobnosti za čuvanje i zaštitu informacija i informacijskih sustava**. Pored toga, CIS razvija i održava mrežni portal www.CIS.hr kao referalnu točku za informacijsku sigurnost za cjelokupnu javnost; izrađuje obrazovne materijale namijenjene javnosti; organizira događaje za podizanje svijesti o informacijskoj sigurnosti u javnosti i pojedinim skupinama te djeluje u suradnji sa svim medijima.

CIS **okuplja mlade** zainteresirane za informacijsku sigurnost i radi na njihovom pravilnom odgoju i obrazovanju u području informacijske sigurnosti te pripremu za **profesionalno bavljenje informacijskom sigurnošću**.

Centar informacijske sigurnosti [CIS] nastao je 2010. godine na poticaj Laboratorija za sustave i signale[LSS] Zavoda za elektroničke sustave i obradbu informacija Fakulteta elektrotehnike i računarstva Sveučilišta u Zagrebu, a kao posljedica 15togodišnjeg rada na istraživanju, razvoju i primjeni informacijske sigurnosti. LSS je među ostalim potaknuo osnivanje CARNetovog CERTa i sudjelovao u izradi Nacionalnog programa informacijske sigurnosti RH.

Smisao CISa je da bude **referentno mjesto za informacijsku sigurnost** za javnost, informatičare i posebno za mlade te da sustavno podiže njihovu svijest i sposobnosti u području informacijske sigurnosti.

Rad CISa podržava Ministarstvo znanosti, obrazovanja i sporta Republike Hrvatske, a omogućuju sponzori.

Prava korištenja

Sadržaj

1. Uvod

Većina web stranica poslana je, od poslužitelja do klijenta, koristeći HTTP (eng. *Hyper Text Transfer Protocol*) protokol. Dobro je poznato da pružatelji Internet usluge (eng. *Internet Service Provider* - ISP) ili druge stranke mogu mijenjati sadržaj web stranica pri prijenosu mrežom. Međutim, pretpostavka kako te promjene rade samo neki posrednički alati je netočna. Promjene web stranica koje se događaju su brojne i raznolike, a često rezultiraju i značajnim problemima za korisnike i autore web stranice ili za oboje.

Pružatelji Internet usluga povećavaju prihode dodavanjem reklama u web stranice i time mijenjanju sadržaj stranice. Korisnici žele ukloniti smetnje na stranici kao što su reklame i prozore koji iskaču, dok autori zlonamjernih programa traže način kako bi neovlašteno širili svoje programe. Mnoge od ovih promjena su nepoželjne vlasnicima web stranica ili korisnicima. Umetanje ili uklanjanje reklama koje rade pružatelji Internet usluga i/ili posrednici može utjecati na prihod autora web stranica, ometati krajnjeg korisnika ili, što je još gore, izložiti krajnjeg korisnika kršenju privatnosti.

Nekoliko tipova promjena stranica uvodi pogreške i ranjivosti u mnoge ili sve web stranice, a korisnik posjećuje stranicu koja bi, da nema promjena, bila sigurna i bez pogrešaka. Takve izmjene predstavljaju prijetnje i mogu se napraviti programi za iskorištavanje tih ranjivosti. Budući da puno ovih promjena ima negativne posljedice, vlasnici web stranica mogu imati poticaj za otkrivanje ili čak sprječavanje njihovog događanja. Otkrivanje izmjena može pomoći autorima web stranica kako bi obavijestili korisnike da se stranica koju posjećuju može pojaviti drugačije nego što su je oni napravili. Mogu se poduzeti akcije protiv onih koji rade neželjene promjene, riješiti probleme zbog izmjene stanica te potencijalno spriječiti neke vrste izmjena. Sprječavanje izmjena može biti važno kada se pokušava povećati sigurnost web stranica te bi takve izmjena web stranica trebalo dopustiti. Neki posrednici firmi mijenjanju sadržaj web stranice kako bi povećali sigurnost klijenata, kao što su: *Blue Coat WebFilter* i *BrowserShield*.

HTTPS protokol (eng. *Hyper Text Transfer Protocol Secure*) daje snažno, ali skupo rješenje ovih problema. On kodira web promet kako bi spriječio izmjene između poslužitelja i preglednika, iako posrednici koji djeluju kao krajnje točke HTTPS[1] protokola mogu i dalje promijeniti stranice bez obavijesti poslužitelju. Kodiranje može spriječiti i one korisne izmjene web stranica koje se oslanjaju na pozitivnoj prirodi HTTP protokola. Web tripwire alat pomaže otkriti izmjene web stranica koje se događaju u prijenosu mrežom, odnosno od poslužitelja do korisnika. Zabrinuti autori web stranica zbog izmjena trebali bi postaviti web tripwire alate na svoje stranice kako bi im pomogli shvatiti i reagirati na bilo koje promjene koje se naprave između poslužitelja i klijenta. Web tripwire alati koji se nalaze na strani korisnika, napravljeni kao *JavaScript* agenti koji mogu otkriti izmjene na web stranicama. Web tripwire alati nisu u potpunosti uspješni, tj. ne mogu otkriti sve promjene, ali mogu se napraviti kako bi bili robusniji u praksi. Implementacija web tripwire alata je jeftinija od uvođenja HTTPS protokola (certifikat može biti besplatan ili koštati između 8$ i 1500$ na godinu). Pružaju web poslužiteljima praktični integritet provjera protiv različitih neželjenih ili opasnih izmjena.

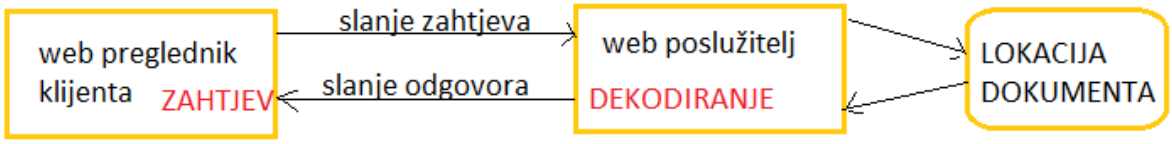
[1] Osnvna ideja HTTPS protokola je stvoriti sigurni kanal u nesigurnoj mreži (Internet). Ovo osigurava prihvatljivu zaštitu ako se koristi odgovarajući šifra i ako je certifikat poslužitelja provjeren i pouzdan.

2. Sigurnosni problemi postojeće web infrastrukture

Želja je napraviti Internet sigurnijim. Danas nije sigurno pretraživati web i zbog toga se razvija puno programa koji pokušavaju povećati sigurnost pretraživanja Interneta. Jedan od tih pokušaja je web tripwire.

2.1. Komunikacija poslužitelja i web preglednika klijenta

Web preglednik, odnosno klijent šalje HTTP zahtjev za određenom stranicom na web poslužitelj. Poslužitelj obrađuje zahtjev te natrag vraća odgovor. On konstantno osluškuje zahtjeve na određenim mrežnim komunikacijskim priključnicama, čekajući da klijent pošalje niz znakova, kojim će zahtijevati uspostavljanje komunikacije. Zahtjev klijenta rezultira slanjem odgovora s poslužitelja, nakon čega će poslužitelj poslati i svoj paket podataka koji najčešće sadrži traženu datoteku ili poruku o grešci. Odmah po ispunjenju zahtjeva klijenta, poslužitelj će prekinuti komunikaciju. Prikaz komunikacije nalazi se na donjoj slici (Slika 1).



Slika 1.Komunikacija između poslužitelja i klijenta

Izvor: CIS

2.2. Opis problema

Tipični scenarij pretraživanja web-a je da u web preglednicima postoje web stranice koje se često posjećuje, web stranice koje su trenutno otvorene i web stranice koje još nisu posjećene. U tim situacijama može se dogoditi puno zlonamjernih stvari. Rezultat pretraživanja može iskoristiti ranjivosti u Internet pregledniku kako bi instalirao zlonamjerni program na računalo. Reklame na web stranicama mogu iskoristiti ranjivosti web preglednika te ako klijent klikne na reklamu može se dogoditi da se instalira zlonamjerni program. Ovo je veliki, ali ne i jedini problem pretraživanja Interneta.

Zlonamjerni XSS (eng. *Cross Site Scripting*) napadi su napadi gdje napadač ima sposobnost postaviti svoju skriptu na nečiju drugu web stranicu. To mu otvara mogućnost za krađu kolačića, mijenjanje sadržaja web stranice ili krađu privatnih informacija. Primjer je *Yahoo*, web stranica za elektroničku poštu, koji je bio ranjiv na XSS napad u 2006. godini. Elektronička pošta koja je dolazila u ulaznu datoteku mogla je čitati skriptu i na taj način slati poruke, odnosno elektroničku poštu, drugim kontaktima koji se nalaze u adresaru ili slati neželjenu poštu (eng. *Spam*).

Postoje i CSRF (eng. *Cross-Site Request Forgery*) napadi gdje web stranice iskorištavaju način na koji preglednici obrađuju uvjerenja. Novi problem koji se javlja je kada klijenti naprave zahtjev za web stranicama. Oni možda neće dobiti stranicu koju je autor stranice namijenio, nego neku modifikaciju takve stranice. Jedan od primjera ovakvih izmjena web stranica je stavljanje ili uklanjanje reklama. Pružatelji Internet usluga postave u web stranice svoje reklame i na taj način primijene stranicu koja se pregledava i može imati ozbiljne posljedice za sigurnost korisnika.

2.3. Web tripwire

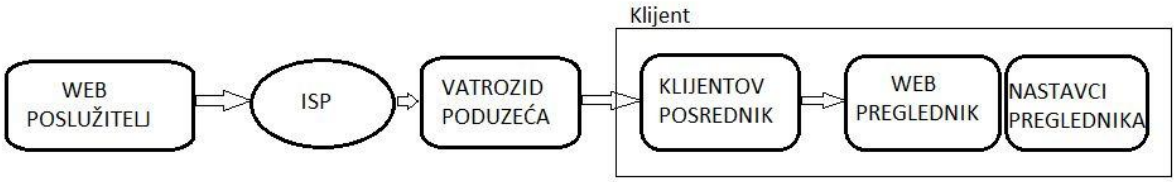
Web tripwire je program koji se umeće u web stranice, a služi za otkrivanje izmjena koje se događaju pri prijenosu mrežom. On pomaže klijentima, jer prikazuje poruku kada dođe do izmjene web stranice koju trenutno gledaju. Vrlo je važno znati kada dođe do izmjene web

stranice, jer se izmjenama mijenja prvobitan sadržaj web stranice te može doći do sigurnosnih propusta koji se nisu nalazili u originalnoj stranici. Izmjena sadržaja može web stranicu učiniti ranjivom pa klijenti njenim pregledavanjem više nisu sigurni. Autori web stranica također žele znati kada dođe do izmjene njihovih stranica, jer to onda više nije stranica koju su oni namijenili za svoje korisnike. U četvrtom poglavlju web tripwire će biti detaljno objašnjen.

3. Razlozi i uzroci izmjena web stranica

Izmjene koje se događaju u prijenosu mrežom, odnosno od poslužitelja do klijenta (tj. njegovog web preglednika) štete mnogima. Prikaz prijenosa mrežom može se vidjeti na slici u nastavku (Slika 2). Jedan od primjera izmjena koje se događaju su kada pružatelji Internet usluga stavljaju reklame u web stranice. Napravljeno je istraživanje na Sveučilište u Washingtonu koje otkriva da li se i kakve izmjene se događaju u web stranicama pri prijenosu mrežom. To bi značilo da pružatelji Internet usluga imaju nekakav uređaj u svojoj mreži koji radi duboko pretraživanje[2] paketa podataka i gleda sadržaj web stranica te rade izmjene kako bi dodali reklame, ili neki drugi sadržaj. Postoje određene tvrtke, kao što je *NebuAd*, koje rade s manjim pružateljem Internet usluga kako bi umetnule dio *JavaScript* koda na svaku web stranicu koju klijent posjeti. Na ovaj način mogu pratiti ponašanje klijenta i imati uvid koje on web stranice posjećuje. U stranice koje klijent posjećuje dodaju se reklame. Postoje besplatne bežične mreže, kao *MetroFi* i *LokBox*, koje na sve stranice stavljaju reklame i na taj način ostvaruju dohodak. Reklame su postavljene tako da mijenjaju prvobitan sadržaj web stranice. Gledano s perspektive pružatelja Internet usluga ovo je dobro, jer na taj način ostvaruju dodatne prihode, dok u isto vrijeme možda smeta korisnicima i može utjecati na ostvarivanje prihoda autorima stranica. Autori web stranica postave reklamu na vlastitu web stranicu i onda kada pružatelji Internet usluga mijenjaju sadržaj stranice i dodaju nove reklame, njihove reklame mogu postati manje uočljive ili potpuno uklonjene sa stranice što je puno važniji problem. Postoji i sve više firmi koje reklamiraju svoje usluge u suradnji s pružateljima Internet usluga. Ove kompanije rade duboko pretraživanje paketa sadržaja web stranice kako bi ih mogli izmijeniti i postaviti svoje reklame.

Međutim, ovo nisu jedine promjene koje na web stranicama rade pružatelji Internet usluga. Neki pružatelji Internet usluga rade i kompresiju na web stranici. Na mrežama male propusnosti radi se kompresija web stranica kako bi one zauzimale manje mjesta. Primjer toga su fotografije koje su velike kvalitete, pa samim time zauzimaju i više dozvoljene propusnosti (jer se treba više podataka prenijeti) te se radi kompresija. Takva fotografija se prikazuje na web stranici. Postoji mogućnost da se fotografija otvori u visokoj kvaliteti, a to se omogućava Java kodom.



Slika 2. Prikaz mreže od poslužitelja do klijenta
Izvor: Detecting In-Flight Page Changes with Web Tripwires

Izmjene na razini vatrozida (eng. *Firewall*) organizacija postavljene su kako bi povećale sigurnost i to je vrsta izmjene web stranica koja je korisna i koja bi se mogla dozvoliti u određenim situacijama. Najčešći primjer je web filtar, *Blue Coat Web Filter*, koji ubacuje dio Java koda u web stranice kako bi mogao tražiti zlonamjerno ponašanje te na taj način izmjenjuje sadržaj web stranice i može ju učiniti ranjivom. Gledano iz perspektive tvrtki, ove izmjene web stranice su jako dobre, jer mogu učiniti Internet sigurnijim za svoje zaposlenike. Također, gledano sa strane autora web stranica može biti dobro i korisno zbog toga što bi ovim izmjenama mogli dodijeliti manji rizik klijentima.

Najviše izmjena događa se kod samog klijenta, alatima za blokiranje reklama i prozora koji iskaču (eng. *Popup*). Jedan od takvih alata je *Zone Alarm* koji služi kao osobni vatrozid. Postoje i alati za posredničke poslužitelje kao što je *Ad Muncher,* koji se mogu instalirati i raditi izmjene web stranice kako bi se spriječile reklame i prozori koji iskaču. Klijentima je ovakva izmjena web stranica dobra, jer uklanja neželjeni sadržaj koji ometa korisnike pri pretraživanju Interneta. No, autorima web stranica ovakva izmjena može utjecati na prihod, jer njihove reklame više nisu vidljive.

[2] Duboko pretraživanje paketa pretražuje sadržaj podataka u paketima te se na temelju toga može raditi filtriranje i izmjenu sadržaja paketa.

Postoje i određene vrste zlonamjernih programa koji rade izmjene web stranca te se mogu otkriti pomoću alata web tripwire. Jedan od primjera izmjene web stranice je zbog alata *Adware*[3]. *Adware* dodaje dodatne poveznice u web stranice koje u originalnom izdanju ne postoje. Poveznice koje dodaje su dvostruko podcrtane i ako se strelicom miša prijeđe preko njih pojavljuje se novi okvir u kojem se nalazi reklama koja je povezana s dvostruko podcrtanom riječi. Prolazeći kroz izvorni kod web stranice *Adware* je radio njegove izmjene i pretvarao neke riječi u poveznice. Ovo je dosta invazivna izmjena u pogledu HTML (eng. *HyperText Markup Language*) izvornog koda.

Izmjene web stranica na strani klijenta mogu raditi i zlonamjerni programi kao što su crvi. Izmjene koje rade zlonamjerni programi narušavaju sigurnost web stranice te na taj način pomažu zločincima.

Neke izmjene koje mijenjaju sadržaj i izgled web stranice nehotice je pokvare, u smislu da web stranica više nije funkcionalna. Izmjene web stranica u prijenosu mrežom uzrokuju pogreške koda kojim je napisana web stranica. U nekim slučajevima zbog izmjene web stranica, dolazi i do izmjene alata web tripwire koji se na njoj nalazi te dolazi do pogreške u radu. Alat više ne radi ono za što je namijenjen te se ne dobiva od njega povratna informacija. Na popularnoj web stranic MySpace i na nekim forumima kada je došlo do izmjene koda stranice došlo je do pogrešaka. Kada su korisnici napisali svoje bilješke, one su imale dodatak koda programa koji blokira prozore koji iskaču. Promjene koje su do ovoga dovele napravio je osobni vatrozid korisnika čija je funkcija blokiranje takvih prozora.

Izmjenama koda web stranica dolazi i do sigurnosnih ranjivosti, poput XSS ranjivosti, koje predstavljaju znatno veći problem, jer su korisnici ugroženi. Sigurnosni propusti događaju se kada web posrednici mijenjaju sadržaj web stranice, jer se na taj način mijenja originalna stranica koja je bila sigurna te sada može postati ranjiva. Napadač zbog XSS ranjivosti koja se događa zbog sigurnosnih propusta na samoj stranici, dobiva mogućnost umetanja vlastitog koda na nečiju web stranicu. Ako klijent pošalje upit za određenom web stranicom i web stranica ga vrati natrag korisniku, otvara se mogućnost da napadač sakrije svoj kod u odgovoru i da se on izvrši u pregledniku klijenta. Razvojni programeri ulažu puno truda da se ti problemi uklone. Neki posrednici svojim radom mijenjaju web stranicu i stavljaju u nju dio koda programa koji je ranjiv na XSS napad. Web preglednik za mobilne uređaje, Opera Mini, cijeli sadržaj dohvaća kroz posrednik koji mijenja format web stranice u format prikladniji za male ekrane. Na ovaj način mijenja se sadržaj web stranice i ona više nije potpuno ista originalu. Ovim izmjenama originalna web stranica koja je bila sigurna može postati ranjiva. Razvojni web programeri ne mogu spriječiti ove izmjene, jer se one događaju nakon što odgovor napusti njihov poslužitelj. Klijentima koji imaju instaliran neki od ovih ranjivih posredničkih alata kao što je *Ad Muncher*, izmijenjen je cijeli Internet promet, te su stoga i sve web stranice postale ranjive na XSS napad. .

[3] Adware je program koji automatski prikazuje ili preuzima oglase na računalo nakon što je instaliran neki program ili nakon korištenja neke aplikacije. Pojedini programi za oglašavanje pripadaju i zlonamjernom programu te se stoga mogu svrstati u programe koji narušavaju privatnost korisnika.

4. Web tripwire

Promjene na web stranici mogu se otkriti koristeći jednostavan *JavaScript* kod koji se postavi na web stranicu koja se promatra i to se zove web tripwire. Web tripwire dio je JavaScript koda koji je postavljen na promatranu web stranicu i koji se šalje na web preglednik te se tamo pokreće. Kada web preglednik pokrene web tripwire kod, on radi provjeru cjelovitosti HTML koda web stranice. Jedan od primjera izmjene web stranice je kada pružatelj Internet usluga izmijeni stranicu dodavanjem reklame i na taj način izmijeni HTML kod. Takvu izmjenu tripwire detektira te prikazuje poruku korisniku kojom ga obavještava da je web stranica koju pregledava promijenjena u prijenosu mrežom. Zapažena izmjena prijavljuje se (u obliku poruke) i poslužitelju kako bi se naknadno mogla analizirati.

Postoji nekoliko tipova web tripwire implementacija koji će biti opisani u daljnjem tekstu.

Mjerenja koja su proveli na Sveučilištu u Washingtonu otkrila su kako izmjena web stranica u prijenosu mrežom može imati negativne posljedice za korisnike i za autore web stranica. Postoji nekoliko razloga zašto otkrivanje izmjena koje se događaju u prijenosu mrežom može biti korisno. One su:

- pretraživači mogu upozoriti korisnike na umetnute skripte koje mogu utjecati na rezultate pretraživanja,
- banke mogu onemogućiti ulazak na web stranicu ako je njihova početna stranica izmijenjena,
- web stranice za elektroničku poštu mogu ispraviti pogreške uzrokovane umetanjem skripte,
- web stranice društvenih mreža mogu upozoriti korisnike ako otkriju ranjivi posrednički alat koji može ugroziti korisničke račune,
- web stranice za oglašavanje mogu se suprotstaviti kompanijama koje dodaju ili izmjenjuju reklame.

Autori web stranica bi možda željeli spriječiti određene vrste izmjena stranice, u svrhu sprečavanja štete za svoje posjetitelje (ili same sebe). HTTPS protokol pruža jedno strogo rješenje za sprječavanje izmjena web stranica tako da se koristi kodiranje (enkripciju) između krajnjih točaka komunikacije. Međutim, korištenje HTTPS (eng. *Hypertext Transfer Protocol Secure*) protokola je skuplje. U slučajevima kada je nemoguće koristiti HTTPS, rješenje može biti korisna i implementacija alata web tripwire. Razlog tome je mogućnost da brzo i efikasno otkriva većinu HTML izmjena uz minimalne troškove. Pružaju i više fleksibilnosti nego HTTPS protokol u smislu kako reagirati u situacijama kada se otkriju promjene. Alat web tripwire može otkriti i prikazati korisnicima kada je došlo do izmjene web stranice. Neki tipovi alata web tripwire mogu samo otkriti kako je došlo do izmjene web stranice, dok neki tipovi mogu odrediti točno mjesto izmjene.

Postoje ciljevi koje bi web tripwire morao zadovoljiti kako bi bio koristan autorima web stranica. Neki tipovi web tripwire alata mogu biti vrlo vrijedni iako ne zadovoljavaju sve ciljeve. Ciljevi su:

- web tripwire alat bi trebao otkriti bilo koju promjenu HTML koda određene web stranice nakon što ona napusti poslužitelj i prije nego stigne na web preglednik klijenta. U ovo se ne ubrajaju izmjene koje se događaju zbog nastavaka koje dodaje preglednik, jer se taj dio smatra važnim za funkcionalnost samog preglednika.
- autori web stranica možda žele web tripwire alat koji bi spriječio samo određene izmjene web stranice, odnosno one izmjene koje bi mogle narušiti sigurnost same stranice. Različiti autori web stranica imaju različite ciljeve u sprječavanju izmjena web stranica.
- Web tripwire bi trebao biti u mogućnosti točno odrediti izmjenu za korisnika i za autora stranice, kako bi im pomogao u razumijevanju njezina uzroka.
- Web tripwire ne bi smjeo utjecati na funkcionalnost izvođenja web stranice u koju je uključen. Trebao bi sačuvati semantiku stranice i podržavati trenutnu inačicu stranice.

Postoji nekoliko strategija za implementaciju web tripwire sustava. Nažalost, ograničenja u web preglednicima otežavaju njihovi izradu. Na Sveučilištu u Washingtonu napravili su pet strategija za izradu JavaScript web tripwire alata te su napravili i usporedbu s HTTPS protokolom. Razlike između strategija prikazane su u tablici (Tablica 1). Svaka od implementacija ima isti osnovni pristup. Web poslužitelji dostavljaju tri elementa do preglednika:

- stranicu koja se zahtjeva,

- tripwire Java skriptu
- dobro poznati prikaz tražene web stranice.

Dobro poznati prikaz može imati jedan od nekoliko oblika, a pri izradi na Sveučilištu u Washingtonu su koristili ili sumu za provjeru stranice ili potpunu kopiju HTML koda web stranice pohranjenog u kodirani niz znakova. Suma za provjeru zahtjeva manje prostora, ali ne može lagano i točno odrediti lokaciju bilo koje otkrivene izmjene. Tehnologija potpune kopije HTML koda web stranice točno određuje gdje se nalazi izmjena koja se dogodila i povećava veličinu stranice koja se dostavlja klijentu. Kada sva tri elementa dođu na korisnikov web preglednik, web tripwire skripta uspoređuje traženu web stranicu s dobro poznatim prikazom i otkriva jesu li se dogodile nekakve izmjene u prijenosu mrežom. Za sve web tripwire implementacije web poslužitelj mora znati koji točno sadržaj web stranice treba provjeriti. Ovaj zahtjev možda zvuči trivijalno, ali mnoge web stranice su jednostavno izlaz programa na poslužitelju i njihov sadržaj ne može biti poznat unaprijed. Za dinamičke web stranice, poslužitelj će možda trebati spremiti sadržaj stranice, ili dovoljno informacija kako bi mogao rekonstruirati sadržaj te kako bi moga napraviti web tripwire koji sadrži dobro poznati prikaz web stranice. Razlike između strategija prikazane su u Tablica 1, a svaka od implementacija ima isti osnovni pristup.

Tablica 1. Usporedba implementacija web triwire rješenja
Izvor: Detecting In-Flight Page Changes with Web Tripwires

Ciljevi	Skripte za brojanje	Pregled DOM-a	XHR zahtjev potom brisanje	XHR zahtjev potom preusmjeravanje	XHR zahtjev na samoj stranici	HTTPS
Otkrivanje svih HTML izmjena	NE	DA	DA	DA	DA	DA
Sprječavanje izmjena	NE	NE	DA	NE	NE	DA
Prikaz razlika	NE	NE	DA	DA	DA	NE
Očuvanje semantike	DA	DA	NE	DA	DA	DA
Postupno učitavanje	DA	DA	NE	NE	DA	DA
Podržavanje povratnoggumba[4]	DA	DA	NE	NE	DA	DA

Web tripwire alat kojeg su napravili studenti na Sveučilištu u Washingtonu i s kojim su radili mjerenja, napravljen je tako da podržava normalne svakodnevne web stranice. Primjer njihove stranice nalazi se na slici (Slika 3). Klijent dohvaća originalne HTML stranice i nema promjene u prvom trenutku. Dodali su jednostavni *JavaScript* kod, koji se sastoji od jedne linije koda, na svoju web stranicu. Jednostavni kod u pozadini dohvaća drugu *JavaScript* datoteku u kojoj se nalazi web tripwire. Ovaj web tripwire uključuje i kodiranu inačicu originalne web stranice koja se može shvatiti kao kontrolna stranica. Ideja je da će web tripwire uspoređivati HTML kod web stranice koju je primio s onom koja se nalazi u web tripwire skripti i provjeriti ima li promjena. U stvarnosti je malo složenije od ovoga, jer se u web pregledniku ne može zatražiti vlastiti HTML kod web stranice kao niz znakova. *JavaScript* kod ima pristup samo unutarnjem prikazu stranice web preglednika i to ovisi od preglednika do preglednika. Stoga su se studenti istraživači poslužiti trikom gdje šalju *XmlHttpRequest* zahtjev za izvornom web stranicom kako bi je na poslužitelju dobili kao originalni niz znakova koje je web preglednik dobio. Nakon ovih radnji može se usporediti dobiveni niz znakova s nizom znakova koji se očekuje te vidjeti ima li bilo kakvih izmjena.

[4] Podržavanje gumba za povratak (eng. *Supports back button*) znači da u web pregledniku postoji mogućnost povratka na stranicu koju se pregledavalo prije trenutne.

Slika 3. Prikaz originalne i izmijenjene stranice
Izvor: Detecting In-Flight Page Changes with Web Tripwires

4.1. Skripte za brojanje (eng. Count Scripts)

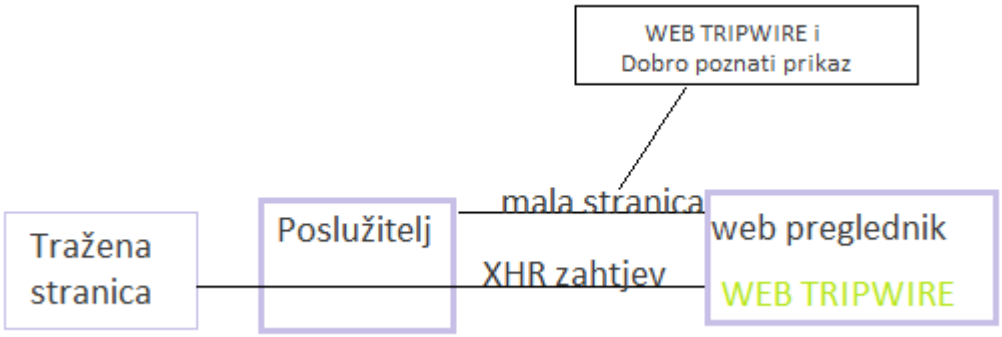
Jednostavan web tripwire koji broji koliko ima oznaka skripte na stranici. Svi veliki web preglednici podržavaju oznake skripte. Oznake skripte se koriste za definiranje skripte na strani klijenta, kao što je JavaScript. JavaScript kod često se koristi za manipulaciju slikama, provjere valjanosti te dinamičke promjene sadržaja. Rezultati mjerenja studenata sa Sveučilišta u Washingtonu pokazuju kako ovaj web tripwire može otkriti 90% izmjena, ali ne otkriva izmjene koje ne utječu na oznake skripte. Crv *W32.Arpiframe* primjer je izmjena koje utječu na oznake skripte. Dobro poznati prikaz web stranice ima očekivani broj oznaka skripte na stranici. Web tripwire skripta uspoređuje taj broj s brojem oznaka skripte koje je prijavio dokumentni objektni model (eng. *Document Object Model*-DOM) kako bi odredio jesu li umetnute nove oznake. Ako je otkrivena izmjena nije lagano otkriti koja od skripti ne pripada originalnoj web stranici, niti je lagano spriječiti pokretanje skripte koja je dodana. Ovaj pristup propušta puno tipova izmjena, ali je jednostavan i ne smeta u izvođenju web stranice.

4.2. Pregled dokumentnog objektnog modela (eng. Check Document Object Model - DOM)

Za opsežnu i cjelovitu provjeru napravljen je web tripwire koji uspoređuje sadržaj cijele web stranice koju klijent dobiva s dobro poznatim prikazom izvorne stranice (koju je poslužitelj poslao). Nažalost, *JavaScript* kod ne može direktno pristupiti HTML nizu znakova koji je web preglednik primio. Skripte imaju pristup unutarnjem prikazu stranice web preglednika preko varijabli. Ovaj unutarnji prikaz ovisi o web pregledniku, te često i o inačicama korištenih web preglednika. Poslužitelj mora unaprijed obraditi stranicu u svim dostupnim preglednicima i inačicama kako bi mogao dati dobro poznati prikaz stranice za bilo kojeg klijenta. Ova tehnika je zbog toga generalno nepraktična. Osim toga, poslužitelj ne može uvijek točno prepoznati korisnički agent klijenta (tj. web preglednik kojeg klijent koristi) pa zato ni ne može znati koji prikaz treba poslati. Umjesto toga poslužitelj mora poslati sve dobro znane prikaze web stranice svakom klijentu. U praksi se šalje lista sa sumom za provjeru kako bi se smanjio opći prostor. Web tripwire skripta provjerava da li se stvarna suma za provjeru web stranice pojavljuje u nizu. Za određenu stranicu koja se promatra izračuna se suma za provjeru te ako dođe do izmjena onda se iznos mijenja. Na taj se način zna kako je došlo do izmjene stranice, ali se ne može odrediti točno mjesto izmjene, jer se suma za provjeru računa za cijelu stranicu.

4.3. XHR zahtjev, a potom pisanje preko stranice (eng. XHR then Overwrite)

Umjesto provjeravanja unutarnjeg prikaza web stranice u pregledniku, ovaj tip web tripwire alata sa poslužitelja dohvaća stranicu koju je zatražio korisnik kao niz znakova. Ovo se može postići korištenjem zahtjeva *XmlHttpRequest* – XHR, koji omogućava skripti da dohvati XML (eng. *Extensible Markup Language*) sadržaj ili druge tekstualne dokumente, sve dok se ti tekstualni dokumenti nalaze na istom poslužitelju kao i trenutna stranica koja se provjerava. Ovo je zanimljiva tehnika za web tripwire iz nekoliko razloga. Prvi razlog je taj što web tripwire skripta prima punu kopiju stranice koju zahtjeva u obliku niza znakova i dopušta izvođenje usporedbe. Drugo, zahtjev se sam po sebi ne razlikuje od tipičnog zahtjeva za web stranicom. I treće, nije vjerojatno kako će odgovor biti izmijenjen s nastavcima koje dodaje web preglednik, zbog toga što se očekuju da odgovor sadrži XML podatke koje ne treba mijenjati. Kao rezultat, web tripwire skripta može imati točan pogled na bilo koju izmjenu web stranice koja se dogodi u prijenosu mrežom. U prvoj XHR web tripwire strategiji, poslužitelj je prvo slao pregledniku malu stranicu (eng. *small boot page*) koja je sadržavala web tripwire skriptu i dobro poznati prikaz stranice koja se zahtjeva kao kodirani niz znakova. Web tripwire skripta tada dohvaća traženu web stranicu s XHR zahtjevom. Uspoređuje odgovor s dobro poznatim prikazom kako bi se otkrile izmjene i nakon toga piše preko sadržaja male stranice koja sadrži web tripwire pomoću funkcije koja se nalazi u web pregledniku. Na slici (Slika 4) nalazi se grafički prikaz. Ova strategija ima veliku prednost jer se može spriječiti većina tipova izmjena koje se događaju tako da uvijek piše preko sadržaja male stranice, koja sadrži web tripwire, sa sadržajem dobro poznate stranice jednostavnim korištenjem XHR zahtjeva za testiranje. Međutim, zlonamjerni korisnici bi lagano mogli zamijeniti sadržaj male stranice s web tripwire skriptom, pa se ovo ne može gledati kao sigurnosni mehanizam. Strategija u kojoj se prepisuje preko stranice ima svojih nedostataka. Prvi je da sprječava stranicu od postepenog učitavanja, jer se cijela stranica mora primiti i provjeriti prije nogo se izvede. Drugi nedostatak je što korištenje unutarnje funkcije preglednika za pisanje smeta gumbu za vraćanje u nekim web preglednicima te sprječava povratak na prethodnu posjećenu stranicu. Treći nedostatak je što postoje pogreške u unutarnjoj funkciji za pisanje u nekim značajnim web preglednicima kao što su *Internet Explorer* i *Safari*. Ta funkcija ima dva načina rada. Ako se funkcija pozove može dodati sadržaj na stranicu ili može zamijeniti cijeli sadržaj web stranice ako se pozove nakon što se stranica učita. Pristup pisanja preko stranice može biti koristan samo u određenim slučajevima.



Slika 4. Prikaz strategije XHR zahtjev, a potom pisanje preko stranice

Izvor: CIS

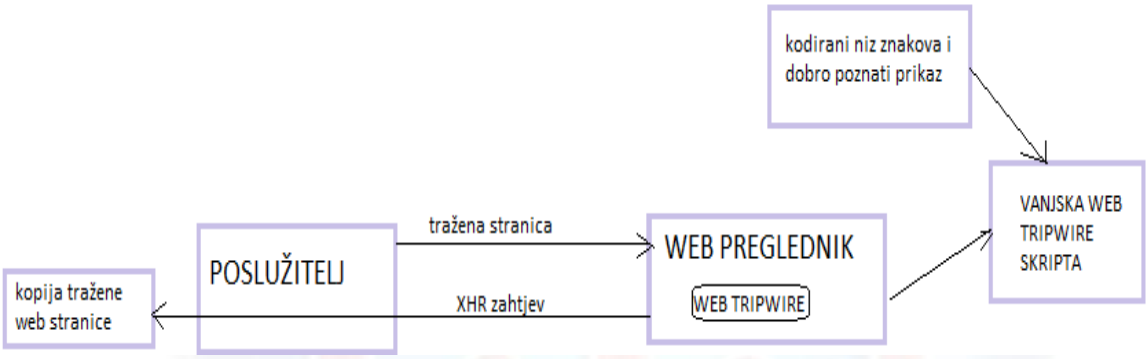
4.4. XHR zahtjev, a potom preusmjeravanje (eng. XHR then Redirect)

Web tripwire skripta dohvaća web stranicu koja se zahtjeva s XHR zahtjevom i potom je provjerava. Umjesto pisanja preko stranice, skripta preusmjerava web preglednik na stranicu koja se zahtjeva. Ovakav pristup sprječava postupno učitavanje i gubi se mogućnost sprječavanja izmjena na stranici, zbog toga što ne može preusmjeriti na dobro poznati prikaz.

Također, konstantno se kvari mogućnost vraćanja na prethodnu stranicu zbog toga što se pokvari gumb za povratak u svim web preglednicima.

4.5. XHR zahtjev na samoj stranici (eng. XHR on Self)

U pristupu XHR zahtjev na samoj stranici, poslužitelj prvo dostavlja zahtijevanu stranicu, a ne malu stranicu s web tripwire skriptom i to omogućava stranici da se učita postupno. Zahtijevana stranica daje upute web pregledniku da dohvati vanjsku web tripwire skriptu, koja sadrži kodirani niz znakova s dobro poznatim prikazom zahtijevane stranice. Nakon toga web tripwire skripta dohvaća drugu kopiju zahtijevane web stranice s XHR zahtjevom, kako bi se mogla obaviti provjera cjelovitosti. Budući da je stranica označena tako da se može pohraniti, barem na kratko vrijeme, preglednik je vraća iz svoje memorije umjesto ponovnog kontaktiranja poslužitelja. Grafička ilustracija strategije nalazi se na donjoj slici (Slika 5). Ova strategija ne može lagano spriječiti promjene, zbog toga što se umetnute skripte mogu pokrenuti prije web tripwire skripte. Međutim, može detektirati većinu promjena na HTML kodu zahtijevane web stranice i prikazati razliku korisniku. Čuva semantiku stranice, sposobnost na postepeno učitavanje stranice i korištenje gumba za vraćanje na prethodnu stranu. Prema navedenim kriterijima, ovo je najbolja implementacija od ovih predstavljenih te ova inačica postiže sve navedene ciljeve osim sprječavanja izmjena u prijenosu web stranice mrežom.



Slika 5. Prikaz strategije XHR zahtjev na samoj stranici

Izvor: CIS

4.6. HTTPS protokol

Ciljevi mehanizama web tripwire tehnologije se malo razlikuju od HTTPS protokola. HTTPS protokol je namijenjen kako bi pružio provjeru pouzdanosti i cjelovitosti za klijente, ali ne daje naznake poslužitelju ako ciljevi nisu postignuti. Web tripwire alati namijenjeni su kako bi pružili provjeru cjelovitosti poslužitelju i po želji mogu obavijestiti klijenta o tome. HTTPS protokol i web tripwire mogu se u nekim slučajevima nadopunjavati, ali HTTPS protokol pruža jače sigurnosno jamstvo. Koristi kriptiranje za određivanje svih izmjena web sadržaja uključujući slike i binarne podatke. Sprječava promjene jednostavnim odbacivanjem bilo koje stranice kod koje je došlo do izmjene u prijenosu mrežom. Također, čuva semantiku stranice i sposobnost da se učita postepeno. Korištenje HTTPS protokola u ovu svrhu stvara veće troškove za autore web stranice nego korištenje web tripwire alata.

5. Procjena i usporedba implementacija web tripwire tehnologije

Da bi se procijenila snaga ili slabost web tripwire alata za autore koji bi ih mogli uvesti u svoje web stranice treba postaviti tri pitanja:

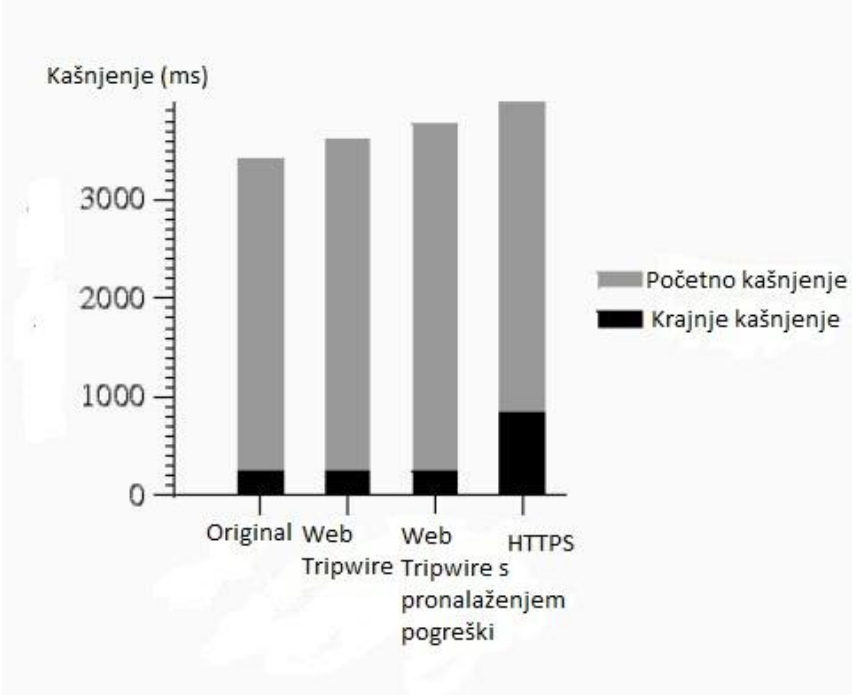
1. Je li web tripwire pristupačan što se tiče cijene u odnosu na HTTP stranice koje nemaju web tripwire alat? Certifikati koje koristi HTTPS protokol za sigurnu komunikaciju u nesigurnoj mreži mogu biti besplatni ili koštati između 8$ i 1500$ godišnje. Ako se koriste besplatni certifikati oni vjerojatno neće obuhvatiti pouzdane certifikate te mogu uzrokovati nepouzdanu komunikaciju.
2. Kako se troškovi web tripwire tehnologije mogu usporediti s troškovima HTTPS protokola?
3. Koliko su web tripwire alati robusni na napade?

Usporedbu troškova korištenja web tripwire alata ili HTTPS protokola za mehanizam cjelovitosti web stranice, izmjerili su kašnjenje studenti na Sveučilištu u Washingtonu na strani klijenta i propusnost poslužitelja za četiri vrste web stranica. Kao osnova uzima se lokalni primjerak početne stranice jedne velike banke. Ovo je realni primjer web stranice koja bi mogla imati web tripwire, zajedno s brojnim ugrađenim slikama, skriptama i stilovima. Napravljena su dva primjerka iste stranice s web tripwire alatima, s time da je jednoj od njih namješteno da prijavljuje izmjene. U oba slučaja koristi se strategija XHR na samoj stranici koja nudi najbolju mogućnost za otkrivanje izmjena, ali ih ne može spriječiti. Treći primjerak web stranice napravljen je pomoću HTTPS koda i nije korišten web tripwire. Eksperiment je napravljen pomoću programa *Emulab*[5] i osobnog računala koje ima 3 GHz Xeon procesor.

Za svaku stranicu posebno je mjereno kašnjenje na korisnikovoj strani pomoću male skripte ugrađene u web stranicu. Mjereno je početno kašnjenje, odnosno vrijeme koje je potrebno da se prva skripta pokrene. Kako bi se pokazao odziv stranice mjereno je i krajnje kašnjenje, odnosno vrijeme koje je potrebno da se stranica potpuno učita. Također, pomoću alata *Wireshark*[6], izmjeren je broj bitova koji su preneseni do klijenta. Testiranje je napravljeno u pregledniku *Firefox* na operacijskom sustavu Windows XP koristeći simuliranu širokopojasnu vezu propusnosti 2 Mbit/s i kašnjenja u jednom smjeru od 50 ms. Svaka prijavljena vrijednost je prosjek od 5 pokušaja, a maksimalna relativna pogreška iznosi 3,25%. Na Slika 6 prikazano je kako stranice s web tripwire sustavom nisu povećale početno kašnjenje više od originalne stranice, koje je bilo oko 240 ms. Za usporedbu, stranica s HTTPS kodom ima puno veće početno kašnjenje koje iznosi oko 840 ms.

[5] Emulab je mrežna ispitna podloga koja pruža istraživačima širok raspon okruženja u kojima mogu razvijati programe, ispravljati pogreške i procjenjivati svoje sustave.

[6] Wireshark je alat koji se koristi za analizu mreže i ispravljanje mrežnih problema. Ranije je bio poznati pod nazivom Etherel, no u novoj inačici došlo je do promjene imena kao i niza dodatnih mogućnosti kao i ispravljenih grešaka.



Slika 6. Prikaz kašnjenja za određene implementacije web tripwire alata
Izvor: Detecting In-Flight Page Changes with Web Tripwires

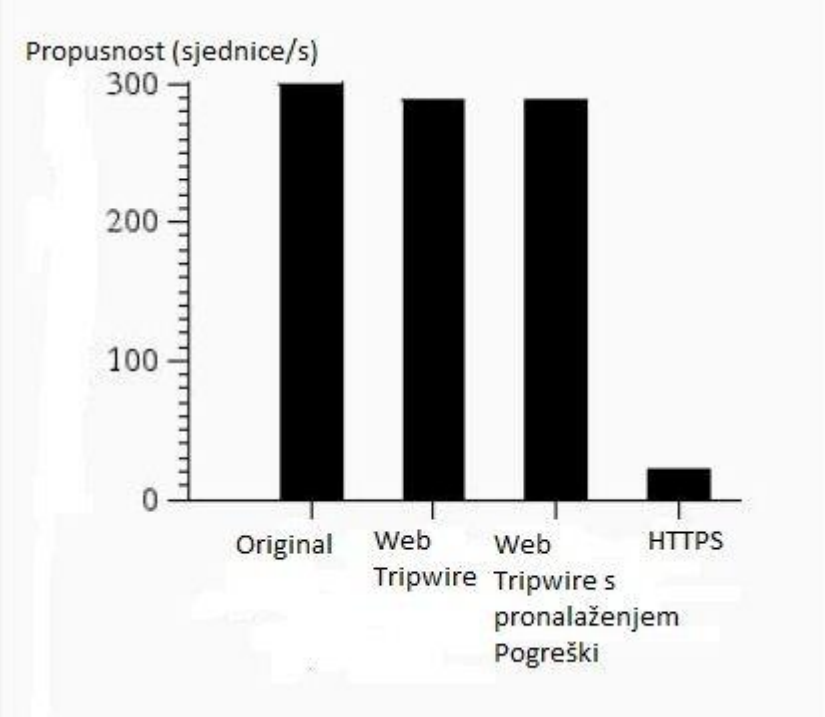
Vrijeme koje je potrebno da se učita cijela stranica, odnosno krajnje kašnjenje, koja ima web tripwire je duže nego vrijeme koje je potrebno da se učita HTTP i HTTPS stranica zbog toga što web tripwire zahtjeva dodatnu obradu skripti u web pregledniku. Stranici s web tripwire alatom koji je prijavio izmjene trebalo je najduže, jer on računa razliku između stvarnog i očekivanog sadržaja stranice. Unatoč tome, krajnje kašnjenje stranica s web tripwire alatom bilo je manje nego stranica s HTTPS kodom. U tablici (Tablica 2) prikazano je kako je prenošenje stranica s web tripwire alatom povećalo ukupni broj bita koji se prenosi za oko 17% u odnosu na originalnu stranicu.

Tablica 2. Broj podataka prenesenih od poslužitelja do klijenta za svaki tip stranice
Izvor: Detecting In-Flight Page Changes with Web Tripwires

Tehnika	**Preneseni podaci**
Original	226,6 KB
Web tripwire	265,8 KB
Web tripwire s ispravljanjem pogrešaka	266,0 KB
HTTPS	230,6 KB

Izmjerena je propusnost poslužitelja pomoću dva Fedora Core 6 klijenta koji pokreću *httperf*[7], na 1 Gbit/s mreži s zanemarivim kašnjenjem. Za svaku stranicu povećano je ponuđeno opterećenje na poslužitelju dok broj održivih sjednica nije došao u maksimum. Svaka sjednica simulira jedan posjet početnoj web stranici banke i uključuje 32 odvojena zahtjeva. Slika 7 prikazuje rezultate mjerenja. Web tripwire je uzrokovao samo 4% degradacija u propusnosti u usporedbi s originalnom stranicom. Za usporedbu, kada se koristi HTTPS propusnost je pala za više od reda veličine. Ovo znači da web tripwire alat treba koristiti s web stranicama na kojima se očekuje veće opterećenje.

[7] Httperf je alat za testiranje i mjerenje učinkovitosti web poslužitelja. Izvorno ga je razvio David Moseberger i suradnici iz Hewlett-Packard Research Laboratories.



Slika 7. Utjecaj web tripwire alata i HTTPS protokola na propusnost poslužitelja
Izvor: Detecting In-Flight Page Changes with Web Tripwires

6. Budućnost

Web tripwire je još uvijek relativno nova tehnologija koja se nije raširila. Trenutno ga koristi tvrtka WordPress koja je napravila dodatak za svoje blogove. Korisnici, ukoliko to žele, mogu preuzeti dodatak za blog s njihove stranice te ga instalirati. Njihov web tripwire alat uspoređuje web stranicu koja se nalazi na poslužitelju s web stranicom koju vidi klijent u svom web pregledniku. Ako ova dva prikaza web stranice nisu ista znači da je negdje u prijenosu mrežom došlo do izmjena.

Tehnologija web tripwire ima veliku mogućnost napredovanja. Postoje neka rješenja koja bi mogla poboljšati i unaprijediti dosadašnja. Buduće web tripwire implementacije mogu se proširiti tako da provjeravaju sve prenesene podatke, umjesto samo HTML koda web stranice. Povećanje u prenesenim bajtovima tada je proporcionalno broju bajtova koji se provjerava s dodatnim web tripwire kodom. Trebalo bi napraviti web tripwire koji može otkriti promjenu koja se događa u web stranici, prijaviti to korisniku, te promjenu spriječiti. Također, web tripwire ne bi smio puno opterećivati web stranicu. Postavljanjem web tripwire alata koji ima mogućnost otkrivanja i sprječavanja izmjena riješili bi svoj problem. Buduća implementacija web tripwire alata može se proširiti tako da provjerava sve podatke koji se prenose, umjesto da provjerava samo HTML kod web stranice. Povećanje prenesenih bita tada je proporcionalno broju podataka koji se provjeravaju, plus veličina dodanog web tripwire koda. Ako je potrebno, ovaj dodatak može se smanjiti tako da se prenosi zbroj za provjeru ili sažeta stranica umjesto punog primjerka. Napretkom web tripwire tehnologije možda neće biti potrebe za HTTPS-om za ovakve slučajeve (situacije u kojima dolazi do izmjene web stranice), jer se trenutno ove dvije tehnologije u mnogo slučajeva nadopunjuju. No, ako web tripwire tehnologija napreduje u smislu da se pokriju svi aspekti u kojima je HTTPS do sada bolji, ova tehnologija bi mogla postati dominantna za otkrivanje i sprječavanje izmjena web stranica u prijenosu mrežom.

7. Zaključak

Izmjene koje se rade u web stranicama veliki su problem za autore i korisnike tih stranica. Autori zbog izmjena web stranice koju su napravili zlonamjerni korisnici gube reklame koje su postavili, a time i prihode od reklama. Izmjene mogu web stranicu učiniti ranjivom na određene vrste napada (XXS i CSRF napadi), a time se ugrožava sigurnost i privatnost korisnika. Korisnici ponekad sami naprave stranicu ranjivom, tako da koriste posredničke alate koji uklanjaju reklame i prozore koji iskaču, jer im je takav sadržaj u web stranici dosadan i ometa ih. Izmjenama stranice čine ih ranjivima. Tu su i promjene koje rade pružatelji Internet usluga, jer oni žele postaviti svoje reklame u web stranice koje prolaze njihovom mrežom. Izmjene se događaju u prijenosu web stranica mrežom, odnosno na putu stranice od poslužitelja do web preglednika na korisnikovoj strani. Web tripwire alati se rade zbog toga što se želi povećati sigurnost u Internetu te spriječiti izmjene web stranica. Mnoge izmjene koje se rade nisu zlonamjerne i opasne, ali se takvim izmjenama može ugroziti sigurnost web stranice i otvoriti put za zlonamjerne napade. Na Sveučilištu u Washingtonu u suradnji s International Computer Science Instituteom napravili su vlastitu web stranicu s web tripwire tehnologijom i promatrali kakve promjene se događaju. Istraživanje je dostupno na sljedećoj adresi:

http://vancouver.cs.washington.edu/

Prikupili su 50 171 IP (eng. *Internet Protocol*) adresu, a kakve sve promjene su zatekli opisano je u tablici (Tablica 3).

Tablica 3. Kategorije uočenih izmjena web stranice
Izvor: Detecting In-Flight Page Changes with Web Tripwires

Kategorija	IP adrese	ISP	Vatrozid poduzeća	Korisnik	Napadač
Blokiranje prozora koji iskaču	277	NE	NE	DA	NE
Blokiranje reklama	188	NE	NE	DA	NE
Problemi u prijenosu	118	DA	NE	NE	NE
Kompresija	30	DA	NE	NE	NE
Narušavanje sigurnosti ili privatnosti	17	NE	DA	DA	NE
Umetanje reklama	16	DA	NE	NE	NE
Promjene oznaka	12	NE	DA	DA	NE
Zlonamjerni program	3	NE	NE	NE	DA
Razni drugi	3	NE	NE	DA	NE

Najviše promjena koje su oni uočili na svojoj stranici događa se na strani klijenta, odnosno korisnika. Često korisnik radi izmjene kojih nije ni svjestan te time omogućava napadačima iskorištavanje ranjivosti koju su napravili.

Web tripwire je vrlo dobro rješenje ako se želi znati kakve izmjene su se dogodile u web stranici i koliko ih je. Za razliku od HTTPS protokola jeftiniji je, a i sama implementacija je jednostavnija. Dobro odrađuje posao za koji je namijenjen i ne smeta normalnom izvođenju web stranice. Može se implementirati u današnje stranice i za sve web preglednike.

8. Leksikon pojmova

XSS napad (Cross-site scripting napad)

Napadačka tehnika koja prisiljava web aplikaciju da korisniku proslijedi zlonamjerni izvršni kod, koji se zatim učitava i izvršava u korisnikovom web pregledniku.

https://www.owasp.org/index.php/Cross-site_Scripting_%28XSS%29

HTTP protokol (HyperText Transfer Protocol)

Osnovna i najčešća metoda prijenosa informacija na Webu. Predstavlja protokol na aplikacijskom sloju OSI modela, a osnovna namjena je prijenos HTML dokumenata (tj. web stranica). HTTP je request/response protokol za komunikaciju između poslužitelja i klijenta. HTTP klijent, kao što je web preglednik najčešće inicira prijenos podataka nakon što uspostavi TCP vezu s udaljenim web poslužiteljem na određenom priključku. Poslužitelj konstantno osluškuje zahtjeve na određenom mrežnom komunikacijskom priključku (tipično priključak 80), čekajući da klijent inicira komunikaciju.

http://hr.wikipedia.org/wiki/HTTP

http://www.w3.org/Protocols/

CSRF (Lažiranje zahtjeva za web stranicom)

Napad' na web stranice koji iskorištava ovjerenje web stranice/aplikacije prema legitimnom autoriziranom korisniku za izvođenje zlonamjernih radnji. Svrha napada je obično krađa povjerljivih informacija o autoriziranom korisniku, a napad se često dostavlja metodama društvenog inženjeringa. Točnije, žrtvi se dostavlja poveznica koja djeluje poznato, te kada korisnik otvori poveznicu pokreće se napad.

https://www.owasp.org/index.php/Cross-Site_Request_Forgery_%28CSRF%29

DOM (Document Object Model)

Platformski i jezično neovisna metoda pristupa objektima u jezicima HTML, XHTML i XML. Objekti DOM modela (kao što su HTML elementi) mogu se adresirati i manipulirati neovisno o jeziku koji se koristi.

http://www.w3.org/DOM/

JavaScript (Programski jezik JavaScript)

JavaScript je skriptni programski jezik, koji se izvodi u web pregledniku na strani korisnika. Napravljen je da bude sličan Javi, zbog lakšega korištenja, ali nije objektno orijentiran kao Java, već se temelji na prototipu i tu prestaje svaka povezanost s programskim jezikom Java. Izvorno ga je razvila tvrtka Netscape (www.netscape.com). JavaScript je izrađen primjenom standarda ECMAScript.

http://javascript.about.com/od/reference/p/javascript.htm

XML (EXtensible Markup Language)

XML je kratica za EXtensible Markup Language, odnosno jezik za označavanje podataka. Ideja je bila stvoriti jedan jezik koji će biti jednostavno čitljiv i ljudima i računalnim programima. U XML-u se sadržaj uokviruje odgovarajućim oznakama koje ga opisuju i imaju poznato, ili lako shvatljivo značenje.

http://webdesign.about.com/od/xml/a/aa091500a.htm

Crv (Računalni crv)

Računalni crv je samo-replicirajući zloćudni program koji koristi mrežu računala kako bi poslao vlastite kopije na druge čvorove mreže bez pomoći korisnika. Ovakvo širenje računalnom mrežom je obično posljedica ranjivosti računala.

http://virusall.com/computer%20worms/worms.php

XMLDsig (XML digitalni potpis)

XMLDsig (također se nazivaju XML Signature, XML-DSig, XML-Sig) definira XML sintaksu za digitalne potpise, a definira ga W3C preporuka XML Signature Syntax and Processing(Sintaksa i obrada XML potpisa).

http://en.wikipedia.org/wiki/XML_Signature

Payload (Koristan teret)

Na području informacijske sigurnosti, koristan teret označava odsječak koda pomoću kojeg se iskorištava određeni propust računala mete. Na primjer, koristan teret računalnog crva može sadržati modul za širenje vlastite kopije putem globalne mreže Internet.

http://searchsecurity.techtarget.com/definition/payload

EFI (Extensible Firmware Interface)

EFI predstavlja programsko sučelje između vlasničkih komponenti i operacijskog sustava. Stvoren je s ciljem mijenjanja starog BIOS sustava koji je zastario.

http://www.pctechguide.com/motherboards/efi-extensible-firmware-interface-explained

IP protokol (Internet Protocol)

IP je jedan od glavnih protokola u Internetu, a koristi se za usmjeravanja paketa kroz Internet. U tu namjenu, dodjeljuje IP adrese izvora paketa i njegovog odredišta na temelju kojih će se paketi usmjeravati kroz nekoliko računalnih mreža.

http://compnetworking.about.com/od/networkprotocolsip/g/ip_protocol.htm

9. Reference

[1] Charles Reis, Steven D. Gribble, Nicholas C. Weaver, Tadayoshi Kohno, Detecting In-Flight Page Changes with Web Tripwires, http://www.cs.washington.edu/research/security/web-tripwire/nsdi-2008.pdf, 2008.

[2] Web Tripwire for WordPress Plugin Demonstration, http://www.youtube.com/watch?v=aF_-qMVH9HQ

[3] Charles Reis, Building a Safer Web: Web Tripwires and a New Browser Architecture, http://www.youtube.com/watch?v=H9_mG_yAoTQ, ožujak, 2008.

[4] Charles Reis, Steven D. Gribble, Nicholas C. Weaver, Tadayoshi Kohno, Detecting In-Flight Page Changes with Web Tripwires, http://www.usenix.org/event/nsdi08/tech/full_papers/reis/reis_html/index.html#note3