CARNet
HRVATSKA AKADEMSKA I ISTRAŽIVAČKA MREŽA
CROATIAN ACADEMIC AND RESEARCH NETWORK

Usporedba besplatnih alata za ispitivanje sigurnosti web aplikacija

CCERT-PUBDOC-2008-06-232

+CERT.hr u suradnji s LSS

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

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

CARNet CERT, www.cert.hr

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

LS&S, www.LSS.hr

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

Sadržaj

1. Uvod

Web aplikacija je aplikacija kojoj se pristupa preko web preglednika kroz mrežu poput Interneta i Intraneta. Mogućnost ažuriranja i održavanje Web aplikacija bez distribuiranja i instaliranja poslužitelja na tisuće klijentskih računala je ključni razlog za njihovu popularnost.

Većina web aplikacija sadrži neku od brojnih sigurnosnih ranjivosti koje mogu ugroziti integritet sadržaja web stranice, otkriti povjerljive podatke te dovesti i do rušenja same aplikacije. Iako postoje mnogi mehanizmi zaštite, svakim se danom javljaju kritičnije ranjivosti koje napadači mogu iskoristiti za određene napade.

Postavljajući jednostavne upite Google tražilici moguće je pronaći ranjive aplikacije, a iskorištavanje ranjivosti ovisi samo o znanju napadača. Jedan od primjera dogodio se nedavno i u Hrvatskoj, kada su turski hakeri napali službene stranice MVPEI (Ministarstva vanjskih poslova i europskih integracija) uoči utakmice Europskog prvenstva.

Ovaj dokument daje kratak pregled osnovnih ranjivosti koje se najčešće javljaju kod većine web aplikacija. Uz kratak opis navedeni su i neki od načina zaštite podataka. Nakon upoznavanja s osnovnim ranjivostima predstavljeni su neki od poznatijih besplatnih alata za skeniranje web aplikacija. Za svaki alat dan je opis glavnih obilježja, dostupnosti proizvoda, popis ranjivosti koje može detektirati i sl. Pri kraju nalazi se kratka usporedba funkcionalnosti besplatnih alata te usporedba s komercijalnim inačicama.

2. Sigurnosni problemi web aplikacija

Web aplikacije sadrže razne sigurnosne nedostatke koje je lako pronaći i iskoristiti. U nastavku je dan popis osnovnih sigurnosnih problema koji su najviše zastupljeni kod većine web aplikacija.

2.1. Vrste sigurnosnih propusta

Jedan od osnovnih problema web aplikacija je postavljanje nepravilnih parametara (informacija u HTTP zahtjevima) što znači da informacije iz zahtjeva nisu provjerene prije korištenja u aplikaciji. Napadač može iskoristiti ove nedostatke da napadne pozadinske komponente kroz web aplikaciju.

Drugi problem predstavlja neispravna kontrola pristupa što znači da ograničenja autoriziranih korisnika nisu ispravno postavljena. Napadač to može iskoristiti za pristup drugim korisničkim računima te za pregled osjetljivih podataka.

Neispravni korisnički računi i upravljačke sesije također predstavljaju jedan od problema kojim napadač može ugroziti lozinke, ključeve, kolačiće sjednica i sl.

Jedan od najraširenijih nedostatak je XSS (eng. Cross Site Scripting) ranjivost pri kojoj se aplikacije koriste kao mehanizmi za prijenos napada do korisničkog preglednika. Uspješan napad može otkriti korisničke oznake sjednica, kao i napasti lokalno računalo, ili iskvariti sadržaj da bi zavarao korisnika.

Problem prepisivanja spremnika javlja se kada pojedine komponente (uključene u CGI biblioteke i komponente aplikacija web poslužitelja) provode neispravnu provjeru ulaznih parametara što može dovesti do potpunog preuzimanja ovlasti nad poslužiteljem. CGI (eng. Common Gateway Interface) je standard za komunikaciju vanjskih programa i informacijskih poslužitelja kao što su HTTP (eng. Hypertxet transport protocol) poslužitelji (web poslužitelja).

Veliki problem uzrokuju i nedostaci nastali umetanjem programskog koda, pri kojima web aplikacije prenose parametre kada pristupaju vanjskim sustavima ili lokalnim operacijskim sustavima. Ako napadač može umetnuti posebno oblikovanu naredbu u ove parametre, vanjski sustav može pokrenuti te naredbe.

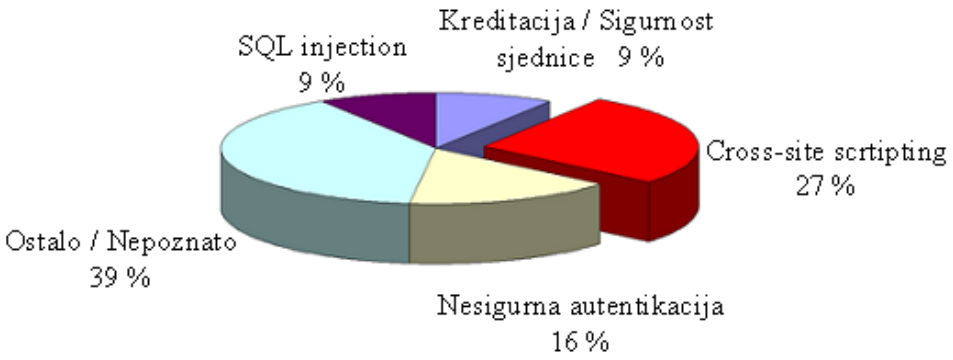
Pogrešno rukovanje pogreškama je nepravilno rukovanje stanjima pogrešaka koja se javljaju tijekom normalnih operacija. Ako napadač može prouzročiti pojavu pogreške koju web aplikacija nepravilno obradi, on može otkriti detaljne informacije o sustavu i prouzročiti DoS (eng. Denial of Service) uvjete.

Nesigurno korištenje kriptografije predstavlja veliki problem jer web aplikacije obično koriste kriptografske funkcije za zaštitu informacija. Te funkcije i kôd za njihovu integraciju često se pokazuju kao slaba zaštita.

Administracijski nedostaci se javljaju kada web aplikacije dopuštaju administratorima da pristupaju stranicama koristeći web sučelja. Ako te administratorske funkcije nisu pažljivo zaštićene, napadač može dobiti pristup svim aspektima stranice.

Važan problem su također i pogreške u konfiguraciji web i aplikacijskih poslužitelja jer je potrebno postaviti odgovarajuću specifikaciju poslužitelja da bi se omogućila sigurnost web aplikacije.

U nastavku je dan kratak opis svih spomenutih problema, kao i opis moguće zaštite. Slika 1 daje grafički prikaz najčešćih ranjivosti web aplikacije prema izvješću WASC (eng. Web Application Security Consortium) organizacije.



Slika 1. Postotci zastupljenosti nekih ranjivosti web aplikacija prema WASC izvješću

2.2. Neispravni parametri

Web aplikacije koriste informacije iz HTTP zahtjeva da bi odlučile kako odgovoriti. Napadači mogu izmijeniti neki dio HTTP zahtjeva, uključujući URL, zaglavlje, skrivene datoteke, polja, kolačiće i nizove upita da bi zaobišli sigurnosne mehanizme stranica. Osnovni napadi koje je moguće izvesti su: umetanje programskih naredbi (PHP, ASP i sl.), umetanje SQL naredbi (eng. SQL injection), preuzimanje kolačića te manipulacija poljima. Neispravna provjera parametara javlja se kad god aplikacija nema jake mehanizme provjere svih informacija iz HTTP zahtjeva.

Većina web okruženja podržava brojne različite načine kodiranja informacija. Parametri se moraju pretvoriti u najjednostavniji oblik prije provjere jer inače posebno oblikovani ulazni parametri mogu biti skriveni.

Iznenađujuće velik broj web aplikacija koristi samo mehanizme provjere ulaznih parametara samo na strani korisnika. Ti mehanizmi se lako zaobilaze, što ostavlja web stranice nezaštićene. Zbog toga su potrebni mehanizmi provjere parametara na strani poslužitelja. Velik broj napada postaje nemoguć ili teško izvodljiv ako programeri obave provjeru informacija prije njihove uporabe.

Svaki dio HTTP zahtjeva koji koriste web aplikacije bez pažljive provjere zove se zaraženi (eng. tainted) parametar. Najlakši način pronalaska takvih parametara je detaljan pregled programskog koda tražeći klase gdje se informacije izvlače iz HTTP zahtjeva.

Najbolji način zaštite je osiguravanje provjere svih parametra prije njihova korištenja.

Parametre treba provjeriti za specifikacije koje definiraju:

- Tipove podataka (string, integer, real...)
- Dopušteni niz znakova
- Minimalnu i maksimalnu duljinu
- Dopuštanje *null* vrijednosti
- Mjesta gdje su parametri dopušteni ili zabranjeni
- Umnožavanja
- Raspon vrijednosti
- Posebne dopuštene vrijednosti (enumeration)
- Posebni modeli (regularni izrazi)

2.3. Neispravna kontrola pristupa

Kontrola pristupa, ponekad zvana i autorizacija, je način na koji web aplikacija dopušta pristup sadržaju i funkcijama pojedinim korisnicima, dok je drugima taj pristup uskraćen. Provodi se nakon autentikacije i vrlo je teška za implementaciju. U osnovi, sastoji se od podjele korisnika u grupe (uloge) sa različitim pravima i mogućnostima.

Mnoge nedostatke u implementaciji kontrole pristupa je lako otkriti i iskoristiti podmetanjem posebno oblikovanih zahtjeva. Neautorizirani korisnik tada može promijeniti sadržaj, pokrenuti funkcije s ovlastima drugog korisnika ili čak postići administratorske ovlasti (potpunu kontrolu nad sustavom).

Način kontrole pristupa treba biti dokumentiran, a ako dokumentacija ne postoji, moguće ja da je stranica ranjiva.

Učinkovita zaštita sastoji se u prvom redu od dokumentiranja kontrole pristupa, tj. potrebno je definirati koji tip korisnika ima pravo pristupa kojem sadržaju i kojim funkcijama. Mehanizam provjere pristupa potrebno je višestruko testirati da bi se osigurala sigurnost.

Neki posebni problemi kontrole pristupa uključuju:

- Nesigurni id (identifikacijski broj) - mnoge web stranice koriste posebne oblike id-a za određivanje korisnika, sadržaja i funkcija. Ako ih napadač pogodi, može povećati prava pristupa te otkriti osjetljive informacije.
- Nasilna provjera kontrole pristupa u pregledniku - provjere prije pristupa samoj stranici
- Napad Path Transversal - pokušaj pristupa datotekama koje nisi dostupne

- Ograničenja datoteka - većina datoteka pohranjenih na poslužiteljima su konfiguracijske datoteke koje ne bi trebale biti dostupne javnosti. Datoteke namijenjene javnosti potrebno je posebno označiti koristeći mehanizam ograničenja.
- Predmemorija na strani korisnika - mnogi korisnici pristupaju web aplikacijama s udaljenih računala preko pristupnih točaka. Preglednici ponekad spremaju stranice kojima zatim napadači mogu pristupati te na taj način dobiti pristup inače nedostupnim dijelovima stranica.

2.4. Neispravni korisnički računi i upravljačke sjednice

Upravljanje korisničkim računima i sjednicama uključuje sve aspekte rukovanja korisničkim računima i aktivnim sjednicama. Autentikacija je jedan dio ovog procesa, a sadrži funkcije vezane za promjenu lozinke, zaboravljene lozinke, pamćenja lozinke, obnove korisničkih računa i sl. Upravljanje sjednicom zahtjeva snažne oznake sjednica koje nije moguće pogoditi, ukrasti ili zauzeti. Sjednica je interaktivna razmjena informacija u komunikaciji između uređaja koja je osnovana u određenom trenutku u vremenu i prekinuta u nekom kasnijem trenutku.

Još jedan veliki problem predstavlja «backend» autentikacija, tj. način na koji web aplikacija provodi svoju autentikaciju prilikom pristupa bazi podataka, direktorijima i web poslužiteljima. Često se lozinke uključuju u izvorni kod ili konfiguracijske datoteke, pa se nedostaci mogu iskoristit za pregled takvih datoteka.

Pregled kôda i testiranje probojnosti mogu se koristiti za otkrivanje problema upravljanja korisničkim računima i sjednicom.

Posebno ugrožena područja su:

- Kontrola promjene lozinke
- Jačina lozinke
- Pohrana lozinki
- Zaštita u prijenosu
- Zaštita id sjednica
- Popis korisničkih računa
- Spremljene verzije stranica
- Povjerljive veze
- «Backend» autentikacija

2.5. XSS nedostaci

Cross-site scripting (XSS ili CSS) nedostaci se pojavljuju kada napadač koristi web aplikaciju da pošalje posebno oblikovan kôd, obično JavaScript, do krajnjeg korisnika. Kada web aplikacija koristi ulazne parametre od korisnika bez njihova filtriranja, napadač može umetnuti posebno oblikovani znakovni niz kao ulazni parametar. Krajnji korisnik vjeruje aplikaciji pa napadač iskorištava to da čini stvari koje mu inače nisu dozvoljene. Napadači također često kodiraju zahtjeve (koristeći Unicode i sl.) pokušavajući sakriti namjere jer kodirani zahtjevi izgledaju manje sumnjivo.

XSS napad može općenito biti podijeljen u dvije kategorije: pohranjeni (eng. stored) i reflektirani (eng. reflected). Pohranjeni su oni napadi kod kojih umetnuti kôd je privremeno pohranjen na ciljni server, u bazu podataka i sl. Reflektirani su oni gdje umetnuti kôd ima drugačiji put do žrtve, kao npr. poruku elektroničke pošte ili neki drugi poslužitelj. Kada se korisnika navede na posjećivanje poveznice (eng. link), umetnuti kôd „putuje“ do ranjivog web poslužitelja koji reflektira napad nazad do korisnikova preglednika pa napadač može pokrenuti kôd jer je došao od povjerljivog poslužitelja. Vrlo je velika mogućnost da neka stranica sadrži XSS ranjivost, a postoje mnogi alati koji pomažu napadačima da pronađu takve stranice (kao što je mogućnost korištenje Google tražilice).

Najbolji način zaštite je pregledavanje kôda s namjerom da se pronađu mjesta na kojima HTTP zahtjevi formiraju HTML izlaze. Filtriranjem izlaznih skripta može se spriječiti prikaz izlaznog sadržaja zlonamjernom korisniku.

2.6. Prepisivanje spremnika

Prepisivanje spremnika je nepravilno stanje koje se javlja kada proces pokuša pohraniti podatke izvan granica međuspremnika fiksne duljine. Napadači koriste prepisivanje spremnika slanjem posebno oblikovanog ulaznog niza u web aplikaciju kako bi pokrenuo proizvoljni programski kôd. Pojavu prepisivanja spremnika nije jednostavno otkriti kao ni iskoristiti. Pojavljuju se kod aplikacija koje koriste biblioteke (npr. grafičke datoteke pri obradi slika), a također se nalaze u korisničkom kôdu web aplikacija pa ih je teže otkriti. Čak i nakon otkrivanja nedostatka teže ga je iskoristiti jer napadaču nisu dostupni izvorni kôd i poruke o pogreškama.

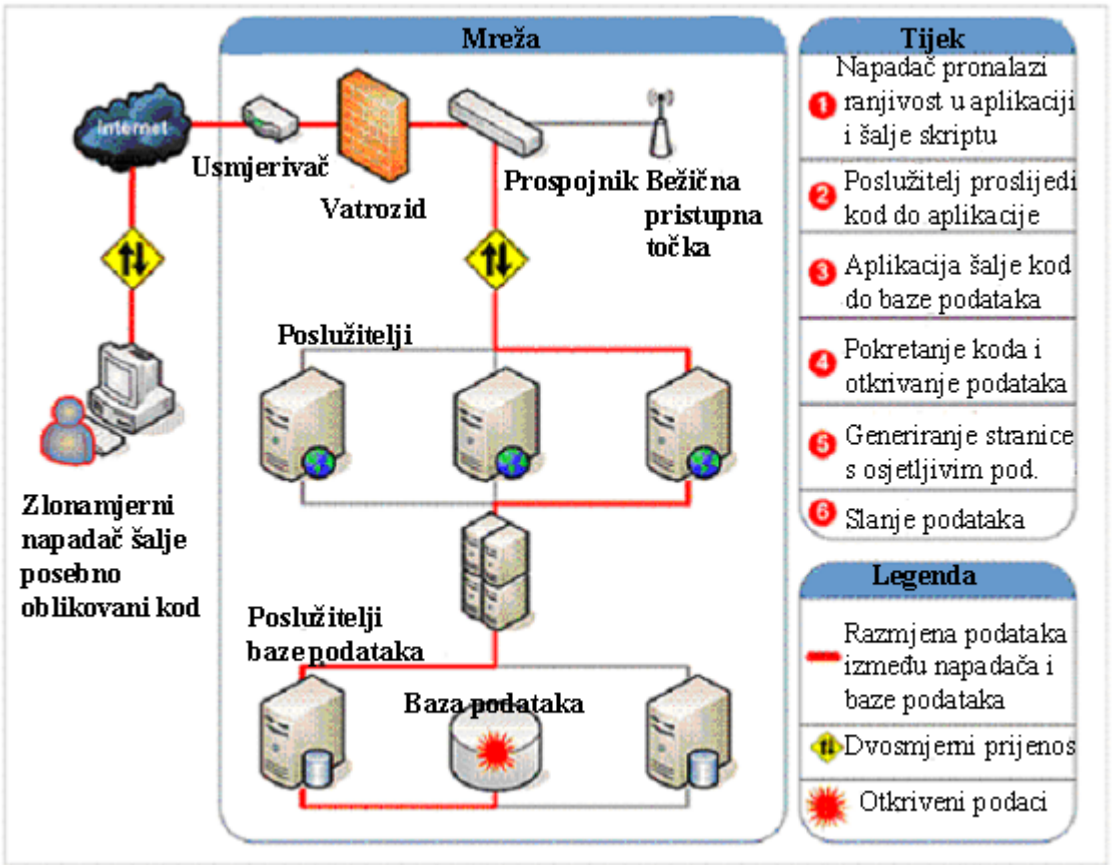
Učinkovita zaštita je praćenje izvješća o proizvodima koje korisnik upotrebljava.

2.7. Osnovni nedostaci nastali podmetanjem naredbi

Osnovni nedostaci nastali podmetanjem naredbi omogućuju napadaču prenošenje posebno oblikovanog programskog koda kroz web aplikaciju do drugog sustava. Ovi napadi uključuju pozive operacijskom sustavu (eng. «system call»), korištenje vanjskih programa (eng. «shell commands») te komuniciranje sa bazom podataka (SQL injection). Čitava skripta pisana programskim jezicima perl, python i sl. može biti umetnuta u aplikaciju i pokrenuta. Scenarij napada prikazan je na slici 2.

Napad podmetanjem SQL upita (eng. SQL injection) je obično najraširenija i najopasnija od svih napada podmetanjem naredbi. Za iskorištavanje ove ranjivosti napadač mora pronaći parametre koji se prenose kroz bazu podataka. Pažljivom manipulacijom sadržaja parametra napadač može navesti web aplikaciju da prenese posebno oblikovan upit bazi podataka. Ove napade je lako izvesti, a napadač može otkriti, izmijeniti ili uništiti sadržaj baze podataka.

Moguće je provesti zaštitu osiguravajući da se web aplikacija pokreće samo s ovlastima koja su potrebna za neku funkciju. Potrebno je provjeriti sve podatke koje unose korisnici kao i sve izlazne podatke.



Slika 2. Izvođenje napada podmetanjem naredbi

2.8. Pogrešno rukovanje porukama o pogreškama

Nepravilno rukovanje porukama o pogreškama može dovesti do raznih sigurnosnih problema kod web stranice. Osnovni problem stvara prikaz detaljnih poruka o pogreškama, spremišta baze podataka (eng. database dump) i kôda pogrešaka korisniku. Napadaču takve informacije mogu otkriti postojanje nedostatka.

Web aplikacije generiraju stanja pogreške prilikom prekoračena memorije (eng. out of memory), pojave pokazivača sa *null* vrijednosti (eng. null pointer evceptions), nemogućnosti dohvaćanja baze podataka (eng. database unvailable) te mnoga druga stanja. Čak i u slučaju kad poruke o pogreškama ne donose mnogo podataka moguće je doznati način rada stranice.

Jedan od osnovnih problema koji je uzrokovan nepravilnim rukovanjem pogreškama je loša sigurnosna provjera. Mehanizam rukovanja pogreškama ne fokusira se samo na korisnika, nego uključuje pogreške koje mogu generirati unutrašnje komponente.

U svrhu zaštite potrebno je dokumentirati rukovanje pogreškama, uključujući tip pogreške te koje informacije otkriva korisniku, a koje ne.

2.9. Nesigurno korištenje kriptografije

Mnoge web aplikacije pohranjuju osjetljive informacije u bazama podataka ili ne drugim sustavima. Obično se koriste razne metode kriptografije kako bi se zaštitili ti osjetljivi podaci. Iako je implementacija poprilično jednostavna, pogreške se obično pojavljuju u sljedećim područjima:

- Nesigurno pohranjivanje ključeva, certifikata i lozinki
- Nepravilno pohranjivanje u memoriju
- Loš izvor slučajno generiranih informacija
- Loš izbor algoritama
- Propusti pri kodiranju kritičnih podataka
- Namjera za uvođenje novih algoritama enkripcije
- Propusti prilikom uvođenja podrške za promjenu ključeva

Najlakši način zaštite od pojave kriptografskih nedostataka je minimizirati uporabu kriptografije te spremati samo podatke koji su stvarno neophodni.

Ako je potrebno korištenje kriptografije učinkovitije je koristiti biblioteke u kojima sigurno nema ranjivosti. Također, važni tajni podaci mogu se podijeliti na dvije udaljene lokacije te se spajati za vrijeme izvršavanja.

2.10. Administracijski nedostaci

Sučelje namijenjeno administratorima pruža snažnu okolinu za upravljanje web aplikacijama u svrhu upravljanja korisničkim računima, podacima i sadržajem stranice.

Osnovni problem koji se javlja u ovom području uključuje manjak autentikacije i enkripcije pristupnih sučelja, nemogućnost sprečavanja pristupa administratora s manjim ovlastima, propusti u mehanizmu odvajanja korisnika i administratora i sl.

Osnovna preporuka u svrhu zaštite je onemogućiti pristup administratora na isti način kao i drugih korisnika (na početnoj stranici i sl.). Upotreba VPN (eng. Virtual Private Network) tehnologije može pružiti udaljenom administratoru pristup unutarnjoj mreži iz koje može pristupiti i stranici kroz osiguranu vezu. Također prilikom pristupa web sučelju potrebno je osigurati odgovarajuće postupke autentikacije. Kada se definira način administratorske autentifikacije i zaštite sjednice, treba donijeti određena previla o tome koji administrator može pristupiti kojem sučelju.

Kao još jedan način zaštite preporuča se odvajanje administratorskih sučelja od korisničkih, što može spriječiti obične korisnike da povećaju ovlasti. Moguće ih je pokrenuti na istim poslužiteljima koristeći drugačije portove, ili na potpuno drugim poslužiteljima. Kada se odvoje korisnička i administratorska sučelja moguće je koristiti filtriranje IP adresa, ako se pristup sučelja treba dopustiti samo sa određenog mjesta.

2.11. Pogreške u konfiguraciji web i aplikacijskih poslužitelja

Konfiguracija web i aplikacijskih poslužitelja igra ključnu ulogu u sigurnosti web aplikacija. Postoje razni problemi u konfiguraciji poslužitelja koji mogu zahvatiti stranicu, a uključuju:

- Neispravljeni sigurnosni propusti kod poslužitelja
- Pogreške u konfiguraciji koje omogućava pristup ograničenim direktoriju i izvršavanje naredbi izvan web direktorija poslužitelja (eng. Directory Traversal napad)
- Nepotrebne izvorne datoteke ili sigurnosne kopije
- Nepravilne dozvole pristupa direktorijima i datotekama
- Uključivanje nepotrebnih usluga (upravljanje sadržajem, udaljeni administratori)
- Izvorni korisnički računi i izvorne lozinke
- Mogućnost pristupa administratorskim funkcijama
- Otkivanje previše informacija u porukama o pogrešci
- Pogreške u konfiguraciji SSL (eng. Secure Sockets Layer) certifikata i postavka kriptografije
- Korištenje vlastitih certifikata za autentikaciju i zaštitu protiv aktivnog prisluškivanja komunikacije među korisnicima (eng. man-in-the-middle napada)
- Korištenje izvornih certifikata

Kada su otkriveni, ovi problemi se mogu lako iskoristiti za ugrožavanje cijele stranice.

Prvi korak uvođenja zaštite je kreiranje pravilnih smjernica za konfiguriranje web poslužitelja, koje bi trebale uključiti:

- Konfiguraciju svih sigurnosnih mehanizama
- Gašenje svih nekorištenih usluga
- Definiranje uloga, prava pristupa i korisničkih računa
- Postavljanje alarma

2.12. Ostali problemi web aplikacija

Iako opisani nedostaci predstavljaju najozbiljniji rizik sigurnosti web aplikacija postoje mnogi drugi sigurnosni nedostaci koji predstavljaju potencijalnu opasnost sigurnosti. Ovdje spadaju:

- Nepotreban i posebno oblikovan kôd
- Napad uskraćivanja usluga (eng. Denial of Service)
- Neautorizirano prikupljanje informacija
- Krađa podataka
- Pogreške u spremljenim inačicama stranica

3. Besplatni alati za ispitivanje sigurnosti web aplikacija

Sigurnosne probleme web aplikacija moguće je otkriti pomoću raznih alata dostupnih na Internetu. U nastavku su dana glavna obilježja nekih besplatnih skenara ranjivosti web aplikacija kao što su programi Nikto, Paros proxy WebScarab te Whisker. Također, ukratko su opisani i alati: Wikro, Wapiti i Grabber.

3.1. Program Nikto

Program Nikto 1.00 Beta je izdan 27. prosinca 2001, a gotovo odmah slijedi i 1.01 inačica. Tijekom dvije godine program Nikto evoluirala u najpopularniji i besplatno dostupni skener web ranjivosti. Inačica 2.0, izdana u prosincu 2007. godine predstavlja nekoliko godina poboljšanja.

Nikto je PERL program osmišljen kako bi pronašao razne vrste sigurnosnih problema web poslužitelja, uključujući:

- Pogreške u konfiguraciji poslužitelja i programa
- Izvorne datoteke i programe

- Nesigurne datoteke i programe
- Zastarjele poslužitelje i programe

Izgrađen je na *LibWhisker* biblioteci i može se pokrenuti na bilo kojoj platformi koja ima *Perl runtime*, a podržava SSL, posrednike (eng.Proxy), autentikaciju poslužitelja, IDS (eng. Intrusion Detection System) i sl. Može biti obnovljen automatski iz naredbenog retka, s time da podržava izborno obnavljanje ažurirane verzije podataka natrag na izvornu verziju.

3.1.1. Zahtjevi operacijskim sustavima

Bilo koji sustav koji podržava osnovne PERL instalacije može pokrenuti program Nikto. Opsežno je testiran na:

1) Windows (koristeći ActiveState Perl)
2) Mac OSX
3) Raznim Linux i Unix instalacijama (uključujući i RedHat, Solaris, Debian, Knoppix itd.)

Inačica 2 je također distribuira kao Windows izvršna datoteka za korištenje na Win32 platformi (ne zahtijeva instalaciju PERL).

Jedini zahtijevani perl modul koji ne dolaze standardno je biblioteka LibWhisker. Nikto dolazi konfiguriran da koristi lokalnu LW.pm datoteku (u «plugins» direktoriju), ali korisnici mogu promijeniti postavke.

Za SSL podršku mora biti instaliran *Net:: SSLeay* perl modul (što opet zahtijeva OpenBSD na UnixWare platformama). Windows podrška za SSL ovisi o instalacijskom paketu, ali poznato je da postoji za ActiveState Perl.

Program Nikto ne pruža grafičko sučelje korisnicima što je za većinu korisnika veliki nedostatak.

3.1.2. Instalacija, održavanje i korištenje

Postupak instalacije pokreće se jednostavnim otvaranjem preuzetih datoteka (uz pretpostavku standardne OS/perl instalacije):

```
tar -xvf nikto-current.tar.gz tar-xvf nikto-current.tar.gz
gzip -d nikto-current.tar gzip-d nikto-current.tar
```

Nikto može biti automatski ažuriran, uz pretpostavku da je računalo na kojem je instaliran povezivano s Internetom, pa je potrebno samo jednostavno pokrenuti Nikto s naredbom za ažuriranje.

```
perl nikto.pl -update perl nikto.pl-update
```

Osnovno skeniranje zahtjeva odgovor jednostavnog poslužitelja preko porta 80, ako drugačije nije specificirano. Poslužitelj može biti IP adresa ili DNS naziv, a potrebno je koristiti -h (-host) opciju.

Provjera poslužitelja na adresi IP 192.168.0.1 na TCP portu 80 glasila bi:

```
perl nikto.pl -h 192.168.0.1 perl nikto.pl-h 192.168.0.1
```

Za provjeru preko drugog porta potrebno je navesti broj porta s -p (-port) opcijom npr. provjera poslužitelja na adresi IP 192.168.0.1 na TCP portu 443:

```
perl nikto.pl -h 192.168.0.1 -p 443 perl nikto.pl-h
192.168.0.1-p 443
```

Domaćini (poslužitelji koji se testiraju), priključci i protokoli mogu također biti navedeni pomoću cijelih URL sintaksa, npr:

```
Perl nikto.pl -h https://192.168.0.1:443/ perl nikto.pl-h
https://192.168.0.1:443/
```

Ako se radi o SSL poslužitelju pomoću opcije *-SSL* ubrzava se test.

```
perl nikto.pl -h 192.168.0.1 -p 443 -ssl perl nikto.pl-h
192.168.0.1-p 443-SSL
```

Program Nikto može skenirati više priključaka navođenjem popisa priključaka u -p (-port) opciju. Portovi mogu biti navedeni kao niz (tj. 80-90) ili kao lista brojeva odvojena zarezima (tj., 80,88,90). Primjer ispitivanja portova 80, 88 i 443:

```
perl nikto.pl -h 192.168.0.1 -p 80,88,443 perl nikto.pl-h
192.168.0.1-p 80,88,443
```

Moguće je skenirti više domaćina i putem tekstualnih datoteka poslužiteljevih imena ili IP adresa. Datoteka poslužitelja mora biti formatirana tako da je zapisan jedan domaćin po redku s tim da je broj porta naveden na kraju svakog retka. Portovi mogu biti odvojeni od pružatelja i drugih priključaka preko dvotočke ili zareza.

Primjer valjane datoteke poslužitelja:

```
192.168.0.1:80
192.168.0.2,80
192.168.0.3
192.168.0.1,80,443
192.168.0.1:80:443
localhost:8888 localhost: 8888
```

Test može biti izveden i ako računalo na kojem je pokrenut Nikto ima samo pristup do ciljanog poslužitelja preko HTTP posrednika. Tada je potrebno postaviti proxy varijable, a zatim pokrenuti Nikto s -u (-useproxy) opcijom. Sve veze koje se odvijaju preko HTTP proxy biti će navedene u konfiguracijsku datoteku.

```
perl nikto.pl -h 192.168.0.1 -p 80 -u perl nikto.pl-h
192.168.0.1-p 80-u
```

Program Nikto omogućuje izlaz u tri oblika: tekstualni, CSV (eng. comma-separated values) ili HTML. Korištenjem -o (-output) moguće je zadati izlazni format specificiran s -F (-Format). Ako se format ne zada podrazumijeva se tekstualni oblik.

3.1.3. Ostale opcije

Program Nikto sadrži brojne opcije koje se mogu dobiti pokretanjem programa s opcijom -h (-help). Također, jedna od pogodnosti su i tehnike promjene (eng. mutation) koje omogućavaju kombiniranje testova ili pogađanje vrijednosti. Koriste se navođenjem odgovarajućeg broja koji definira vrstu testa:

1 - Testirati sve datoteke sa svim korijenskim direktorijima

2 - Pogađanje zaporka za imena datoteka

3 - nabrajanje imena korisnika putem Apache sustava

4 - nabrajanje imena korisnika putem cgiwrap programa

Program Nikto također sadrži razne opcije prikaza:

1 - Prikaz preusmjeravanja

2 - Prikaz primljenih kolačića (eng. Cookies)

3 - Prikaz svih 200/OK odgovora

4 - Prikaz URL adresa koje zahtijevaju provjeru autentičnosti

D - Debug Output

V - Verbose Output

Postoje i načini poboljšanja skeniranja koji smanjuju broj testova na ciljano računalo pomoću -T (-Tuning) opcije.

Postoje sljedeće –T opcije:

0 - Poslane datoteke (eng. upload)

1 - Zanimljive datoteke

2 - Pogreške u konfiguraciji / Izvorne datoteke

3 - Otkrivanje informacija

4 - Napadi podmetanjem (XSS / skripte / HTML)

5 - Udaljeni pristup datotekama – unutar web korijena

6 - Denial of Service

7 - Udaljeni pristup datotekama – udaljen poslužitelj

8 - Pokretanje naredbi

9 - SQL Injection

a - Zaobilažene autentikacije

b - Identifikacija programa

c - Udaljeno uključivanje izvora

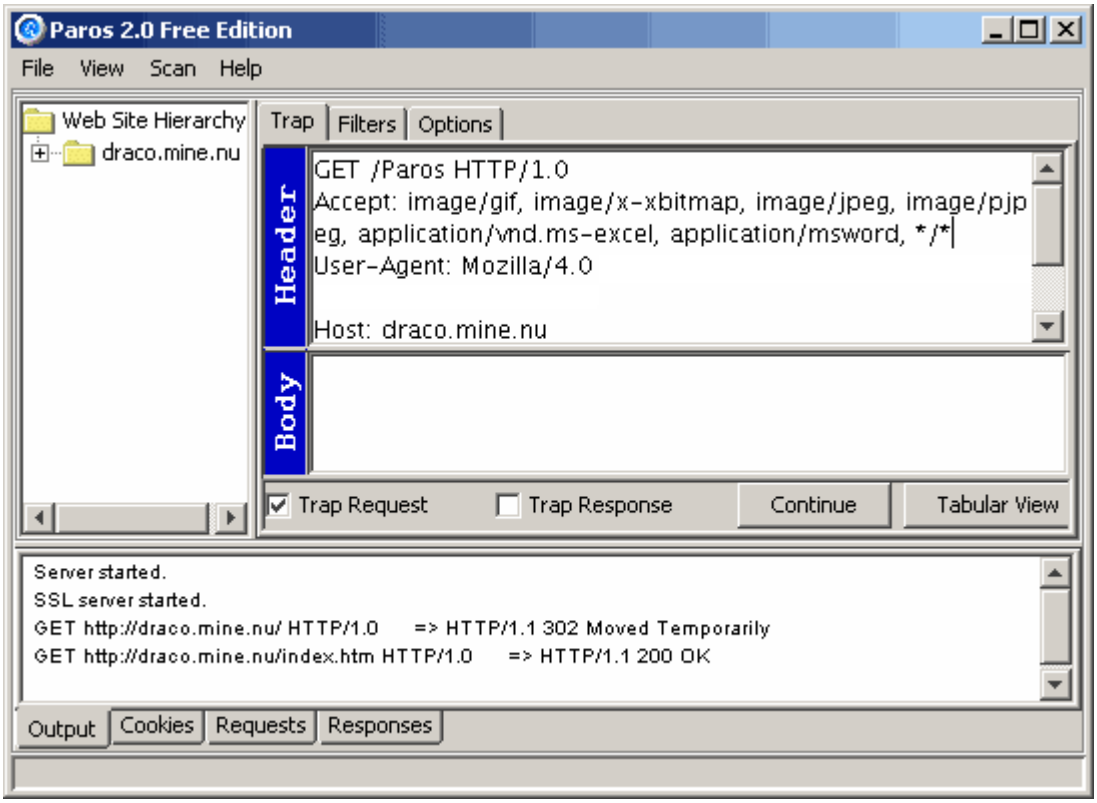
x - Invertirane opcije poboljšanja

3.2. Program Paros proxy

Program Paros je alat za analiziranje sigurnosti web aplikacija napisan u programskom jeziku Java. Koristeći Paros posrednik (eng. proxy) moguće je modificirati sve HTTP podatke između korisnika i poslužitelja, uključujući kolačiće (eng. cookies) i polja za unos.

Dostupan je za Microsoft Windows, UNIX, Linux te Aplle Mac OS X operacijske sustave, a korištenje zahtjeva prethodnu instalaciju JRE 1.4.x. (Java Run Time Enviroment). Korisnici Windows operacijskih sustava trebaju samo pratiti upute za danju instalaciju, dok korisnici UNIX (i drugih) operacijskog sustava trebaju ručno otpakirati datoteke u novi direktorij i pokrenuti .jar datoteku. Paros pruža pregledno korisničko sučelje kao što je prikazano na slici 3.

Prva inačica, Paros v1.0 je izdana u kolovozu 2002.g. , dok je trenutna inačica Paros v3.2.0Alpha izdana 10 Studenog 2004.g.



Slika3. Korisničko sučelje programa Paros

3.2.1. Osnovne funkcije

Pauk (eng. spider) ulazi u web stranice i prikuplja što je više moguće URL adresa, što omogućava bolje razumijevanje hijerarhije web stranica.

Uključuje:

- Ulazak u HTTP i HTTRS web stranice temeljene na danom URL
- Podrška kolačićima
- Podrška zamjenu posrednicima
- Automatsko dodavanje URL adresa u stablo hijerarhije web stranice za potrebe kasnijeg skeniranja

Ograničenja:

- Nemogućnost ulaženja u SSL web stranice sa nepravilnim certifikatom
- Nemogućnost prepoznavanja nekih posebno oblikovanih URL adresa u HTML stranicama
- Nemogućnost pronalaženja URL adresa pisanih u Javascript

Funkcija skeniranja je skenirati poslužitelje temeljene na hijerarhiji web stranica, a također moguće je pretraživati pogreške u konfiguraciji.

Trenutno, program Paros ima sljedeće opcije:

- Provjera da li je uključena opcija PUT u direktorijima poslužitelja
- Provjera da li je moguće pregledati direktorije poslužitelja
- Provjera da li postoje datoteke koje su zastarile
- Provjera za XSS
- Provjera da li postoje izvorne datoteke na poslužitelju

Kako su sve provjere temeljene na URL adresama u hijerarhiji web stranica, skener provjerava svaku adresu za svaki nedostatak.

Kao filtar program Paros obavlja:

- Otkrivanje pojave prethodno definiranih obilježja u HTTP porukama, te obavještavanje korisnika
- Zapis informacija koje su zanimljive (npr. kolačići)

Kako svaki filtar izvodi svaku HTTP poruku, korištenje svih može usporiti rad posrednika.

Dostupni filtri:

- LogCookie - zapis svih primljenih kolačića poslanih iz preglednika do poslužitelja
- LogGetQuery - zapis svih HTTP GET upita poslanih iz preglednika
- LogPostQuery - zapis svih HTTP POST upita poslanih iz preglednika
- CookieDetectFilter - obavijest korisnika o postavljenoj opciji „Set-Cookie“ u HTTP odgovoru
- IfModifiedSinceFilter - uklanjanje polja „If-Modified-Since“ i „If-None-Match“ u HTTP zahtjevu

Ostale funkcije:

- Dekodiranje podataka u drugi oblik: Base64, SHA1 i MD5
- Hvatanje HTTP zahtjeva i odgovora - Uključivanjem opcije „Trap Request“ u polju „Trap“ prate se svi zahtjevi, dok se uključivanjem opcije „Trap Response“ prate odgovori.

3.3. Program WebScarab

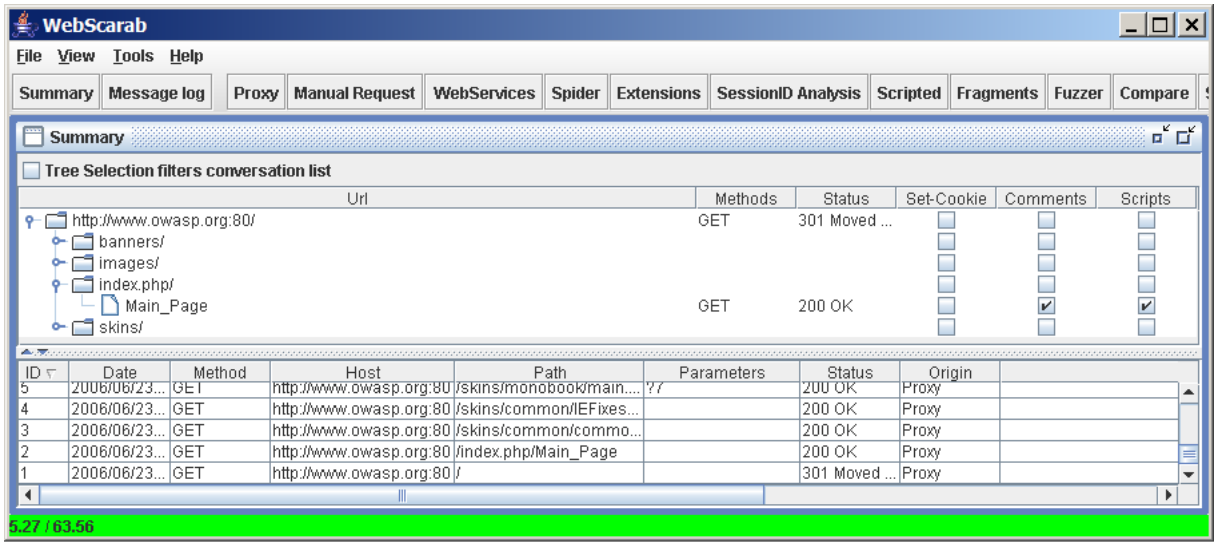
WebScarab je okvir za analizu aplikacija koje komuniciraju pomoću HTTP i HTTPS protokola, napisan u programskom jeziku Java. U najčešćem obliku, alat WebScarab radi kao zadržavajući posrednik (eng. intercepting proxy), čime omogućava operateru pregled i izmjenu zahtjeva kreiranih u pregledniku prije slanja poslužitelju te odgovora koje vraćaju poslužitelji (prije primanja od strane preglednika). Operater može također pregledati i konverzacije (zahtjeve i odgovore) koji su prolazili kroz WebScarab.

WebScarab je osmišljen kao alat za potrebe otkrivanja događaja HTTP(S) aplikacija, bilo da omogući programerima ispravljanje pogrešaka u problemima ili da stručnjacima za sigurnost identifikaciju ranjivosti.

Nakon preuzimanja potrebnih datoteka instalacija se obavlja na sljedeći način:

```
Linux: java -jar ./webscarab-selfcontained-[numbers].jar
Windows: pokretanje .jar datoteke
```

Dostupna je inačica i za operacijski sustav Mac OS X , a postoji i Java Web Start inačica (aplikacija kojoj se pristupa putem web sučelja). Korisničko sučelje programa prikazano je na slici 4.



Slika4. Korisničko sučelje programa WebScarab

3.3.1. Osnovne značajke

Alat WebScarab pruža brojne dodatke koji uključuju:

- Ulomci (eng. fragments) - izvučen HTML sadržaj HTML stranice kao da se radi s posrednikom
- Posrednik (eng. Proxy) - prati promet između preglednika i web poslužitelja (HTTP i kriptirani HTTPS promet) stvarajući SSL vezu, umjesto izravnog povezivanja.
- Ručno presretanje - omogućuje korisniku izmjenu HTTP i HTTPS zahtjeva i odgovora, prije nego dođu do poslužitelja ili preglednika.
- Beanshell - dozvoljava izvršavanje kompleksnih operacija na zahtjeve i odgovore.
- Otkrivanje skrivenih polja - ponekad je lakše modificirati skrivena polja (eng. hidden field) u stranici, nego zadržati zahtjev nakon što je poslan. Ovaj dodatak omogućuje jednostavne promjene skrivenih polja u HTML stranicama u tekstualna polja čineći ih vidljivima.
- Simulator širine pojasa - omogućava korisniku testiranje sporije mreže te testiranje učitavanja web stranice (npr. pristup modemom).
- Spider - identificira nove URL-ove na ciljnoj stranici
- Ručni zahtjev - uređivanje i ponavljanje prethodnih zahtjeva, ili stvaranje potpuno novih.
- SessionID analiza - prikuplja i analizira nekoliko kolačića
- Scripted - operateri mogu koristiti BeanShell za pisanje skripte za kreiranje zahtjeva i dostaviti ju poslužitelju pa skripta tada može izvesti analizu odgovora.
- Parametri fuzzer - obavlja automatsku zamjenu za vrijednosti parametra koji neće proći provjeru parametara (eng. parameter validation), dovodeći do ranjivosti (poput Cross Site Scripting i SQL Injection).
- Pretraga - omogućuje korisniku da oblikuje proizvoljni BeanShell izraz da identificira konverzacije koje bi trebale biti prikazane u popisu.
- Usporedba - izračunava udaljenost (eng. edit distance) između tijela odgovora promatrane i odabrane konverzacije. Vrijednost predstavlja broj uređivanja potrebnih za pretvaranje jednog dokumenta u drugi.
- SOAP - dodatak koji obrađuje WSDL, te predstavlja razne funkcije i potrebne parametre koji se mogu uređivati prije slanja poslužitelju.
- Proširenja (eng. Extensions) - automatizira pretrage za datoteke koje su zabunom ostale u korijenskom direktoriju poslužitelja

- XSS / CRLF - pretražuje korisnički kontrolirane podatke u zaglavlju i tijelu HTTP odgovora kako bi identificirali potencijalne CRLF injection i reflektirane cross-site scripting (XSS) ranjivosti.

3.4. Program Whisker

Whisker je napredni CGI skener sigurnosnih problema koji obavlja skeniranje na temelju prikupljenih podataka. U radu se oslanja na modul *libWhisker* pisan u programskom jeziku Perl, koji je namijenjen isključivo testiranju HTTP protokola. Pruža funkcije za testiranje HTTP poslužitelja za mnoge sigurnosne probleme pri čemu može koristiti bazu podataka drugih CGI skenera.

Dostupan je za sljedeće operacijske sustave: Microsoft Windows, Linux, UNIX (OpenBSD, FreBSD, Solaris i dr.) te Aplle Mac OS X, a postupak instaliranja zahtjeva preuzimanje odgovarajućih datoteka.

Program Whisker uključuje sljedeće značajke:

- CGI direktorij može biti po izboru preimenovan iz izvornog '/ cgi-bin'
- Prije provjere ranjivosti program će potvrditi da postoji CGI direktorij
- Tip i inačica poslužitelja moraju se označiti prije bilo kakvih ispitivanja što smanjuje broj provjera za nepodržane CGI (tj. test za details.idc ranjivost na Apache poslužitelju je neuspješan, budući da je to IIS ranjivost).
- «Virtual Hosting» je u potpunosti podržan, čime program može testirati ranjivosti pod domena unutar istog poslužitelja (ova značajka nije podržana od strane svih CGI skenera)
- Moguće je postaviti pretragu kroz korisnički stvorene stranice (“success” stranice) koje su obično rezultat pogreške “not found”
- Mogućnost međudjelovanja proizvoda/datoteka kao što su datoteke odvojene naredbama, «nmap» datoteke rezultata, IP subnets i sl.
- Dekodiranje URL adresa skriva skeniranje IDS (eng. Intrusion Detection System) programa
- Podrška za skriptni jezik omogućava jednostavno dodavanje novih skripta za skeniranje.
- Podržava višedretvenost (samo za UNIX operacijski sustav)
- Podrška za distribuirane posrednike
- Podrška za skeniranje više datoteka istovremeno

Jedan od nedostataka za krajnjeg korisnika je svakako to što program ne pruža grafičko sučelje.
Napomena: program Whisker se ne održava i ne nadograđuje od 2003.g.

3.5. Ostali alati

Program Wikto je alat za analizu ranjivosti web poslužitelja, ali ne i web aplikacija. Omogućava pretraživanje različitih direktorija i datoteka na poslužitelju te skripta koje bi se mogle zloupotrijebiti. Pretraživanje autorizacijskih problema, SQL injection ranjivosti i sl. nije implementirano. Pisan je u .NET C# te sadrži grafičko sučelje. Nakon preuzimanja instalacijskih datoteka potrebno je konfigurirati program prije prvog korištenja. Nedostatak ovog paketa je u tome što je dostupan samo za operacijski sustav Migrosoft Windows.

Program Wapiti je skener ranjivosti web aplikacija. Zasniva se na "black-box" skeniranju, tj. ne proučava kod web aplikacije, ali pretražuje ga da bi pronašao mjesta u koja je moguće umetnuti podatke. Razvijen je u programskom jeziku Python (zahtjeva podršku inačice 2.4 ili više) te koristi Python biblioteku nazvanu *lswww*. Ne oslanja se na bazu podataka o ranjivostima (kao npr. Nikto) te ne pruža grafičko sučelje.

Može detektirati sljedeće ranjivosti:

- Osnovne injection ranjivosti (PHP/ASP/JSP SQL i XPath injestion)
- XSS (Cross Site Scripting)
- Nepravilno rukovanje datotekama
- LDAP injection
- Detektiranje pokretanja naredbi
- GRLF injection

Program Grabber je jednostavni skener sigurnosnih problema web aplikacija pisan u programskom jeziku Phyton. Jedna od prednosti mu je velika mogućnost adaptacije, dok mu je mana brzina pa je namijenjen skeniranju manjih aplikacija (poput osobnih stranica, foruma i sl). Ne pruža grafičko sučelje ni pdf izvješća (samo XML).
Problemi koje otkriva program Grabber:

- XSS (Cross-Site Scripting)
- SQL Injection
- Uključivanje datoteka (eng. File Inclusion)
- Provjera sigurnosnih kopija
- AJAX provjera
- Analiza javaScript izvornog koda
- Generalizacija datoteka za sljedeće analize

4. Usporedba funkcionalnosti alata

4.1. Usporedba besplatnih alata

Jedan od najboljih skenera otvorenog koda svakako je program Nikto, a svoju funkcionalnost zahvaljuje biblioteci *LibWhisker*. Alat testira ranjivosti web aplikacije uključujući preko 3200 potencijalno opasnih datoteka u poprilični kratkom roku i jako dobrom točnošću (u odnosu na ostale alate). Nedostatak programa je u potrebi za svakodnevnim ažuriranjem (u svrhu obnove podataka o ranjivostima), jer novije i kritičnije ranjivosti ne moraju biti detektirane. Korištenje alata (kao i sve dodatne funkcije) je detaljno objašnjeno u dostupnoj dokumentaciji, a jedina mana za krajnjeg korisnika je nedostatak grafičkog sučelja.

Za razliku od programa Nikto, program Paros proxy sadrži grafičko sučelje što ga čini prihvatljivijim za korisnika s manje računalskih vještina. Prednost ovog programa je mogućnost modificirana svih HTTP podataka između korisnika i poslužitelja. Sadrži brojevne funkcije poput praćenja web prometa, računanje hash funkcija te skeniranje osnovnih web ranjivosti (SQL injection i XSS). Program Paros proxy svakako je jedan od boljih web skenera, a kao i Nikto, dostupan je za većinu operacijskih sustava.

Program WebScarab je zapravo okvir koji služi za analizu aplikacija koje komuniciraju HTTP protokolom, a prvotno je osmišljen za korisnike koji dobro razumiju protokol HTTP. Prednost programa je upravo u tome što je zamišljen kao okvir pa se nekorištene funkcije mogu ukloniti što ubrzava vrijeme izvođenja. Dostupan je za većinu operacijskih sustava, a također pruža i grafičko sučelje.

Jedan od boljih CGI skenera je i program Whisker, koji se poput programa Nikto oslanja na biblioteku *LibWhisker*. Nedostatak je u tome što nije ažuriran neko vrijeme, iako je autor razvio velike zakrpe koje će dodati više provjera i značajki. Trenutna verzija također ima sposobnost skeniranja preko SSL poslužitelja, ali prvotno je namijenjena za skeniranje URL adresa. Korisnik će dobiti obavijest ako se tražena stranica na može dohvatiti, ali provjera IIS (eng. Internet Information Server) ranjivosti (poput Unicode ili Double Decode directory traversal ili Netscape's PageServices) nije implementirana. Iako nije jednostavno implementirati pretragu za navedene ranjivosti, male izmjenom koda u „scan.db“ datoteci mogu poslužiti kao privremeno rješenje.

4.2. Usporedba s komercijalnim inačicama

Postoje brojni komercijalni skeneri koji služe za provjeru ranjivosti web aplikacija. Neki od poznatijih alata su: Acunetix Web Vulnerability Scanner (WVS), N-Stalker, HP WebInspect, Parasoft WebKing, MileSCAN…

Glavna prednost komercijalnih alata nad besplatnim je automatizacija skeniranja koju pružaju gotovo sve komercijalne inačice. Prednosti komercijalnih inačica prikazane su kroz pregled karakteristika programa Acunetix Web Vulnerability Scanner.

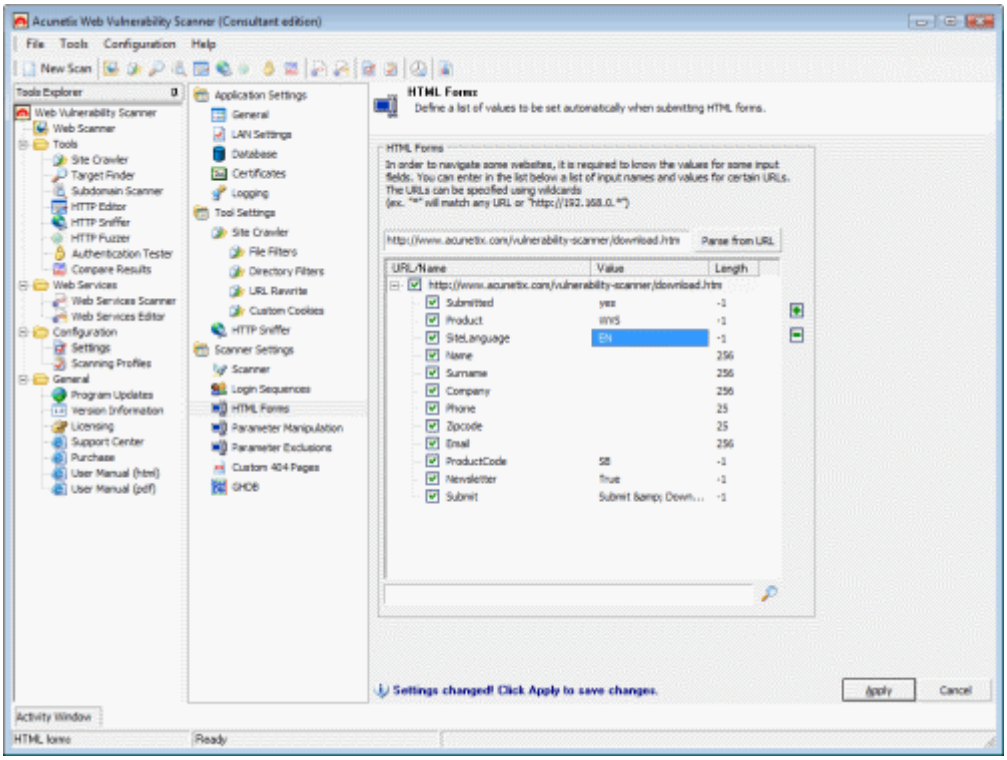
Acunetix Web Vulnerability Scanner je automatizirani skener web ranjivosti koji pretražuje stranice tražeći ranjivosti koje mogu zlonamjerni korisnici iskoristi. Pretražuje sve web ranjivosti uključujući SQL injection, Cross site scripting, CRLF injection, pokretanje proizvoljnog koda, Directory Traversal napade, autetikacijske ranjivosti te umetanje datoteka. Prednost alata je brzina i mali broj pogrešnih detekcija.

Također, omogućava skeniranje najsloženije AJAX/Web 2.0 web aplikacije. Sadrži opsežan modul za pružanje izvješća koji može generirati izlazne datoteke. Koristeći GHDB (eng. Google Hacking Database) pokreće upite na sadržaj stranice otkivajući osjetljive podatke i ranjivosti.

Sadrži napredne alate poput:

- HTTP Uređivač - konstruiranje HTTP/HTTPS zahtjeva i analiza odgovora poslužitelja
- HTTP Sniffer – preuzimanje, zapis i modificiranje svog HTTP/HTTPS prometa te prikaz svih podataka koje šalje web aplikacija
- HTTP Fuzzer – izvodi napredno testiranje ranjivosti poput prepisivanja spremnika i provjere ulaznih znakova

Jedna od značajka koju pruža rijetko koji web skener je mogućnost automatskog ispunjavanja polja i autentikacije prilikom logiranja kao je prikazano na slici 5.



Slika 5. Sučelje Acunetix Web Vulnerability skenera

5. Zaključak

Web aplikacija koja sadrži ranjivosti ugrožava sigurnost cijele baze podataka te cijelog računalnog sustava jer web stranice moraju biti konstantno dostupne da bi pružale usluge korisnicima. Vatrozid i slični programi u ovom slučaju ne osiguravaju zaštitu od zlonamjernih aktivnosti zato što web aplikacije često imaju izravan pristup korisničkim bazama podataka, a moraju biti dostupne i izvan lokalne mreže, pa je teško osigurati sigurnost. Jedan od glavnih problema je kako otkriti ranjivosti web aplikacija prije nego ih napadači iskoriste.

Skeniranje sigurnosnih ranjivosti obavljaju razni alati dostupni na Internetu za slobodnu uporabu. Jednostavni su za instaliranje i korištenje, a većina pruža funkcionalno korisničko sučelje (ili postoje jednostavne i intuitivne naredbe koje se pokreću uz naredbenog retka). Učinkovitost pojedinog alata ovisi o sadržaju koji se pretražuje, ali većina može provesti skeniranje osnovnih ranjivosti. Proučavanjem osnovnih obilježja alata lako je pronaći odgovarajući skener koji treba primijeniti za pretraživanje ranjivosti pojedine aplikacije.

Nakon izgradnje vlastite web aplikacije svakako se preporuča pokretanje nekog od opisanih alata kako bi se uočili sigurnosni nedostaci te ispravili prije korištenja same aplikacije.

6. Reference

[1] The Ten Most Critical Web Application Security Vulnerabilities, OWSAP (The Open Web Application Security Project), 13. siječanj 2003

[2] Top 10 Web Vulnerability Scanners: http://sectools.org/web-scanners.html

[3] Program Nikto: http://www.cirt.net/nikto2

[4] Program Paros: http://www.parosproxy.org/index.shtml

[5] Program WebScarab: http://www.owasp.org/index.php/Category:OWASP_WebScarab_Project

[6] Program Whisker: http://www.wiretrip.net/rfp/

[7] Program Wikto: http://www.sensepost.com/research/wikto/

[8] Program Wapiti: http://wapiti.sourceforge.net/

[9] Program Grabber: http://rgaucher.info/beta/grabber

[10] Program Acunetix Web Vulnerability Scanner: http://www.acunetix.com/