Inspiratron.org - Natural language processing, machine learning and cybersecurity

Naj?eš?i sigurnosni propusti u web aplikacijama

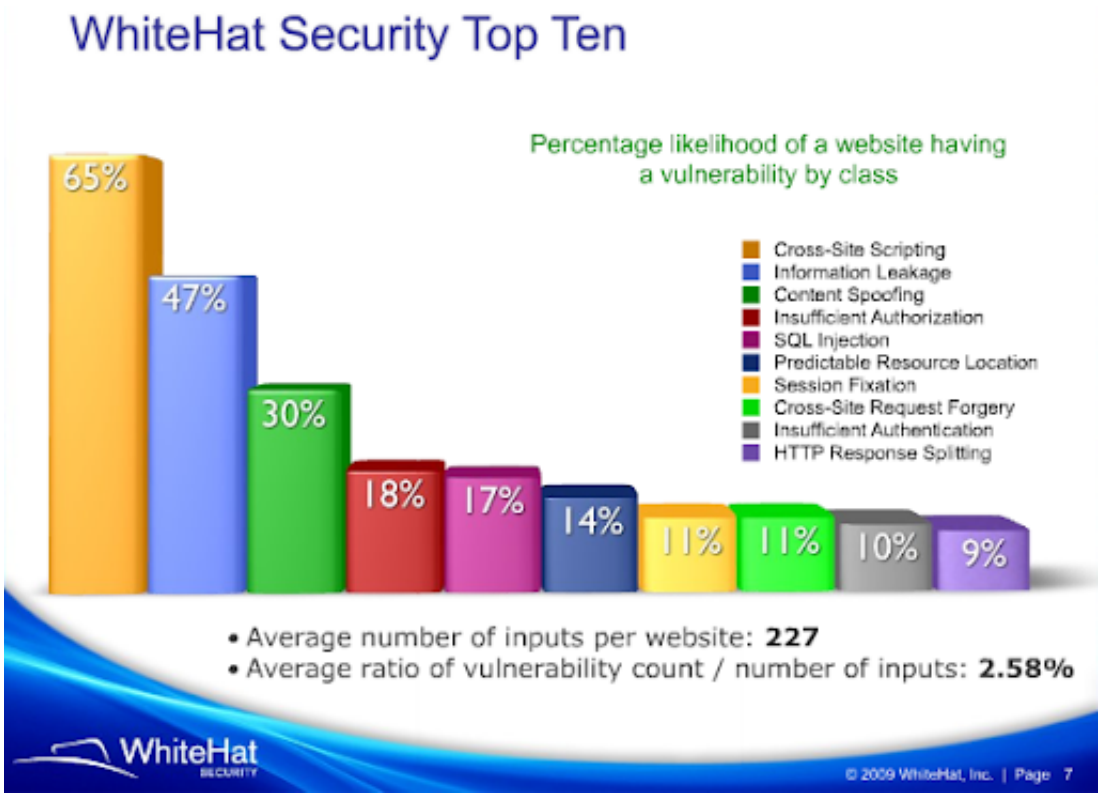
by Nikola Miloševi? - Tuesday, October 30, 2012

https://inspiratron.org/blog/2012/10/30/najcesci-sigurnosni-propusti-u-web-aplikacijama/

S obzirom da ima malo toga napisano na srpskom o bezbednostnim propustima na srpskom jeziku, mislim da je pravo mesto ovde da napišem o tome koju re?. Pokuša?u da opišem naj?eš?e bezbednostne propuste koji se javljaju u web aplikacijama i koji se uglavnom oslanjaju na 2 liste - OWASP top 10 i CWE25.

Injekcije (Injection) - Ova kategorija propusta se odnosi na ubacivanje napada?evog, malicijoznog koda u kod web aplikacije. Naj?eš?e se pominje SQL injection kada se govori o ovoj klasi napada. Me?utim postoje i mnogi drugi tipova injekcijonih napada poput code injection, sa podvrstama PHP injection, javascript injection. Cross site scripting je tako?e vrsta injection napada, jer se ubacuje maliciozni HTML kod u kod sajta. Mogu?e je injectovati i komande operativnog sistema, što je jedan od najtežih oblika injekcija, jer daje napada?u kontrolu nad operativnim sistemom i samim ra?unarom. Težina napada može da varira, od bezazlenog do veoma opasnog u zavisnosti šta je korisniku omogu?eno. SQL injection je naro?ito opasan je je napada? u mogu?nosti da kontroliše bazu podataka, a samim tim i da menja ili briše podatke. Kod SQL injection-a tako?e postoji nekoliko vrsta napada poput blind SQL injection, DOM based SQL injection itd. PHP ili server side code injection tako?e predstavlja jedan od težih propusta, jer se omogu?ava ubacivanje koda koji se izvršava na serveru i koji može da upravlja svim podacima. Prilikom obog napada mogu se ubaciti i kodovi sa eksternih stranica ili redirektovati sa sajta na drugi sajt, uz kra?u sesije ili kola?i?a. Rešavanje problema injekcije varira od vrste injekcije na koju je web aplikacija ranjiva. Uglavnom je potrebno filtrirati specijalne karaktere kako bi se onemogu?ilo izvršavanje koda. Potrebno je obratiti pažnju da se onemogu?i preskakanje tih karaktera nekim specijalnim znakom i tako zaobi?e zaštita. Korisni?ki unos je potrebno uvek proveriti, po mogu?nosti i na klijentskoj i na serverskoj strani da li je validan i ne sadrži neki napad, izfiltrirati ga ili odbiti pre bilo kakve dalje obrade.

Slomljena autentifikacija i upravljanje sesijama (Broken authentification and session management) - ?est je slu?aj da timovi za razvoj softvera implementiraju svoj na?in autentifikacije ili upravljanja sesijom. Ove stvari spadaju u oblast kriptografije, a kriptografija je teška. Dakle implementirati ovo na korektan i neranjiv na?in je teško. U ovakvim aplikacijama napada? može biti u mogu?nosti da reverznim inzenjeringom dozna kako radi algoritam i uspe da na?e na?in kako da krade tu?e sesije i identitete. Pronalaženje ovakvih grešaka može biti teško, jer zavisi od svake implementacije, ali ne i nemogu?e. ?esto se u funkcijama za logout, pam?enje lozinke, upravljanje timeout-ima i sl nalaze ovakve greške. Preproru?uje se uvek koristit framework ili algoritme koji je proveren za obezbe?ivanje sesija i upravljanje autentifikacijom.



white hat security statistics

Nebezbedno referenciranje objekata (Insecure direct object references) - ?esta greška programera je da prilokom pristupa odre?enim objektima u web aplikaciji se ne proveravaju prava pristupa korisnika, oslanjaju?i se na to da samo ono ?emu je korisniku omogu?eno da pristupi preko korisni?kog interfejsa aplikacije ?e korisnik samo da pristupa. U praksi to ne mora biti slu?aj. Menjanjem URL-a ili slanjem snimljenih i izmenjenih zahteva korisnik može zatražiti pristup bilo kom objektu. Ukoliko se ne proveravaju prava pristupa korisnik ?e mo?i i da pristupi tim objektima iako ne bi trebalo da im ima pristup. Bitno je proveravati prava pristupa prilikom svakog pristupa nekom objektu ili stranici.

Cross site request forgery (CSRF) - U ovoj ranjivosti se omogu?ava napada?u da kreira zahtev i da ga na neki na?in ubaci u web aplikaciju, tako da kada autentifikovan korisnik pri?e toj stranici pošalje se napada?ki zahtev, ali kao da ga je poslao autentifikovan korisnik. Ovo je naro?ito bitno da se ne pojavljuje na stranicama online bankarstva ili online prodaje, gde ovakvi zahtevi mogu da prouzrokuju veliku nov?anu štetu. ?esto je ovaj napad povezan sa injection napadom, odnosno XSS, kako bi napada?u omogu?io da preko žrtvinog browsera pokrene legitiman zahtev. Kako bi se odbranili potrebno je dodati neki token u skriveno polje, koje ?e se slati u telu HTTP zahteva i koje ne?e biti otvoreno preko URL-a. Mogu?e je koristiti neke frameworke kao na primer OWASP CSRF Guard koji ovo na automatizovan na?in radi.

Bezbednostna miskonfiguracija - Konfiguracija aplikacije, kao i servera, platforme treba da bude specifikacijom definisana. Greška u konfiguraciji otvara prostor napada?u za razli?ite napade. Ovaj problem je naj?eš?i izvor site deface-a. Tako?e kako bi se izbegli napadi potregno je držati softver ažuran uklju?uju?i sav serverski softver, aplikaciju, kao i biblioteke koje aplikacija koristi.

Nerestriktiranje URL pristupa - Greška mnogih aplikacija je da kontrolišu prava pristupa prilikom pritiska na odre?eni link ili dugme, pre nego što pošalju korisnika na odre?enu stranu. Opet predpostavljaju?i da korisnik ne?e pristupiti ne?em što mu ne omogu?ava korisni?ki interfejs, ne kontroliše se pristup samoj stranici. Tako napada? iako nema pravo pristupa unosom URL-a može da pristupi stranici za koju nema prava pristupa. Potrebno je proveriti prava pristupa i prilikom svakog pristupa stranici.

Nedovoljna zaštita transportnog sloja - ?est slu?aj je da podaci ?iji transport treba da bude zašti?en poput lozinki ili drugih tajnih informacija se prenose kroz mrežu u ?istom tekstu, bez koriš?enja bilo kakvog algoritma za kripciju ili hash. U ovom slu?aju aplikacija je ranjiva na napad poznat pod imenom Man in the middle ili prisluškivanje. Kako bi se odbranili potrebno je koristiti kripciju, bilo koriste?i protokol koji ve? koristi kripciju poput HTTPS-a ili da se koristi neki jak algoritam kripcije.

Nevalidirani redirekti i prosle?ivanja - ?est slu?aj je da sajtovi prosle?uju korisnike na druge web stranice, bilo svojih partnera ili ne. ?esto se ne proverava sa kojim se sve podacima to prosle?ivanje odvija, pa može napada? da ubaci prosle?ivanje do svog fishing site-a kojim ?e da krade uz pomo? parametara sesijske podatke. Potrebno je validirati korisni?ke parametre prilikom svakog redirecta ili forward-a. Tako?e ukoliko je mogu?e pokušati izbe?i parametrizovane redirect-e

Preplavljivanje bafera (Buffer overflow) - napad ?est sistemima pisanim u programskim jezicima koje zahtevaju od programera upravljanje memorijskog prostora. Ukoliko se to upravljanje ne radi na odgovaraju?i na?in napada? je u mogu?nosti da preplavljivanjem bafera jedne aplikacije u?e u memorijski prostor druge i na taj na?in sruši aplikaciju ili ?ak preuzme kontrolu nad operativnim sistemom.

Nedovoljna sigurnost senzitivnih podataka - ?est je slu?aj da se senzitivni podaci ?uvaju u bazi podataka bez kripcije. Tako ne bi trebalo, jer se ne smemo osloniti na to da napada? ne?e do?i u posed naše baze podataka. A ukoliko do?e, automatski su kompromitovani svi podaci, kao i identiteti svih korisnika. Sigurnost treba planirati na više slojeva, pa tako senzitivne informacije treba da budu kriptovane uvek ili nad njima iskoriš?ena jednostrana hash funkcija.

Nikola Miloševi?
