Smart Com

O varnosti spletnih aplikacij

..Tokrat smo se o varnosti spletnih aplikacij pogovorili z Vladimirjem Banom iz podjetja Smart Com..

Vladimir Ban, IT strokovnjak, je del ekipe za varnostne storitve pri Smart Comu. Z nenehnim spremljanjem in uvajanjem novih varnostnih mehanizmov ter rešitev že več kot 15 let pomaga podjetjem, organizacijam in inštitutom reševati varnostne izzive sodobnega poslovanja.

RN: Kaj sploh so spletne aplikacije? Zakaj vsi o njih toliko govorijo in kako so spremenile naše vsakdanjike?

V. B.: Spletne aplikacije so vse aplikacije, za katere uporabnik na svoji strani uporablja spletni brskalnik. Takšne aplikacije danes srečamo praktično na vsakem koraku.

RN: Zakaj so spletne aplikacije tako razširjene?

V. B.: HTML, Java, PHP, .NET in ostale podobne tehnologije so se v zadnjih letih močno razvile in omogočile razvijalcem skoraj popolno svobodo pri razvoju uporabnikom prijaznih, enostavnih in všečnih aplikacij.

Niso pa uporabniki in razvijalci edini zagovorniki spletnih aplikacij. Dejstvo, da za njihovo uporabo uporabnik potrebuje zgolj spletni brskalnik (ki je danes sestavni del vsake uporabnikove naprave), je zelo koristno tudi za skrbnike IS. Skrb za programsko opremo na strani uporabnikov ni več potrebna, kar močno olajša njihovo delo.

RN: Zakaj bi morali razmišljati o varnosti spletnih aplikacij?

V. B.: Zaradi razširjenosti spletnih aplikacij so te hkrati tudi magnet za napadalce. Poleg tega spletne tehnologije vsebujejo določene specifike, ki jih uporabniki oz. skrbniki včasih niso vajeni, kar napadalci radi izkoristijo. V zadnjih letih so močno napredovale tudi metodologije zlorab teh aplikacij.

RN: Kako preverimo varnost že nameščenih aplikacij? Kako preverimo varnost pred namestitvijo novih?

V. B.: Izziva varnosti pri že nameščenih spletnih aplikacijah se lotimo z izvedbo varnostnega pregleda. To je metoda, kjer se izvajalec varnostnega pregleda postavi v vlogo napadalca in na različne načine poskuša zlorabiti aplikacijo.
Pojem »varnostni pregled« sicer marsikdo že pozna iz segmenta mrežne varnosti. Izkušeni skrbniki vedo, da obstaja cela vrsta kakovostnih orodij, ki lahko dokaj učinkovito zaznajo večino varnostnih pomanjkljivosti na omrežju. Pri tem poglobljeno poznavanje omrežij včasih niti ni potrebno. Ko gre za spletne aplikacije, pa to ne drži. Tudi tu so sicer na voljo orodja, a skrbnikom ponujajo vpogled zgolj v površinske pomanjkljivosti. Za odkrivanje pravih pomanjkljivosti je potrebnega veliko ročnega dela s poglobljenim znanjem spletnih tehnologij. Bližnjic tukaj žal ni. Še najbolj učinkovit način je, da za varnost poskrbimo že v fazi razvoja aplikacije. Ta strošek je bistveno nižji kot pa strošek popravljanja že narejenih aplikacij.

RN: Kaj so glavni varnostni problemi na spletnih aplikacijah? Kako rešujemo te probleme?

V. B.: Varnostni problemi na aplikacijah so lahko zelo različni. Najprej spregovorimo o 4 glavnih načelih, ki so pomembna za razumevanje varnosti aplikacij, potem pa o konkretnih pomanjkljivostih.

- **Arhitektura aplikacije**

Spletna aplikacija je v osnovi skupek različnih direktorijev in datotek. Če želi biti napadalec uspešen, mora arhitekturo dobro spoznati. Razvijalci zato poskušajo skriti dele aplikacij. Tipičen primer je skrit direktorij, do katerega z linkom ne kaže nobena stran, ampak mora uporabnik za dostop poznati točen URL naslov. Takšno »skrivanje« je sicer zelo dobra praksa, saj onemogoča napadalca, ki zgolj bežno preverja svojo tarčo. V primeru veščih napadalcev, ki uporabijo več različnih metod za razkrivanje skritih delov aplikacije, pa je to le pomožni varnostni mehanizem.

- **Predvidevanje uporabe brskalnika**

Tipičen uporabnik pri spletni aplikaciji uporablja spletni brskalnik. Brskalnik predstavlja filter med tem, kaj uporabnik naredi, in kaj aplikacija prejme. Brskalnik izvaja čiščenje vnesene vsebine ter določa podatke, ki se pošiljajo aplikaciji v glavah paketov. Tovrstne funkcije so sicer lahko zelo koristne pri pohitritvi aplikacij, za zagotavljanje varnosti pa so neuporabne. Kontrola na strani brskalnika lahko onemogoči napadalca, ki zgolj bežno preverja aplikacijo. Vendar pravi napadalci nikoli niso omejeni z brskalnikom in lahko z različnimi prijemi poskrbijo, da se proti aplikaciji pošlje poljubna informacija. Vsi kontrolni mehanizmi morajo torej biti postavljeni na strani aplikacije in ne na strani uporabnika.

- **Avtentikacija uporabnika**

Prepoznavanje uporabnikov je ključni del marsikatere spletne aplikacije. Pri »navadnih« TCP sejah smo navajeni, da se avtentikacija izvede na začetku seje. Če je ta uspešna, se seja vzpostavi. Posebnost spletnih aplikacij je v tem, da se pri uporabi sproža veliko http sej. Pri tem tovrstni princip avtentikacije ni primeren, saj bi sicer moral uporabnik »vsakih 10 sekund« ponovno vpisati geslo. Mehanizmi za avtentikacijo spletnih aplikacij so zato raje povezani s sledenjem aktivnosti uporabnika na strani aplikacije ter podatki (»session-cookie« ipd.), ki se prenašajo iz paketa v paket. Ti zagotavljajo aplikaciji vedenje, kateri paket pripada kateremu uporabniku. Takoj je jasno, da so ti mehanizmi lahko zelo kompleksni in posledično ranljivi.

- Pomanjkljiva logika aplikacij

Včasih zaradi dreves ne vidimo gozda. Pomembna je namreč celotna logika aplikacije in ne samo tehnično pravilen zapis posameznega stavka izvorne kode. Če je logika aplikacije pomanjkljiva, napadalci lahko s spretnimi menjavami klicev, piškotkov in podobnimi prijemi potencialno še vedno prelisičijo aplikacijo. Tovrstne pomanjkljivosti so najbolj skrite, ugotovi jih le vešč napadalec.

»Sedaj, ko poznamo štiri glavna načela, na katerih se tipično lomi varnost aplikacije, poglejmo še nekaj tipičnih varnostnih pomanjkljivosti.«

- Zmožnost napada na gesla in uporabniška imena

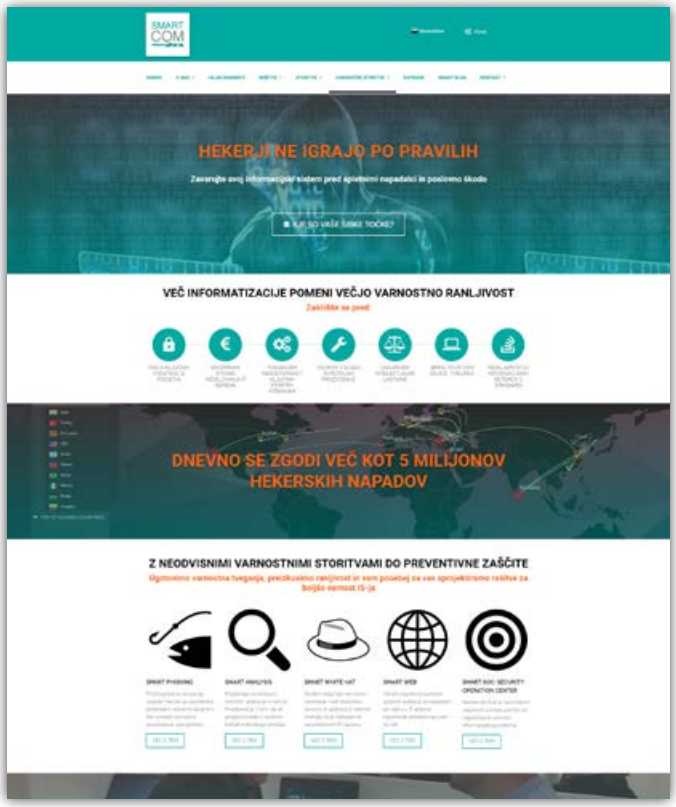
Z enkratnimi gesli in podobnimi mehanizmi lahko preprečimo ugibanje gesel, vendar ni nujno, da aplikacija takšne mehanizme uporablja. Velikokrat vidimo primere, ko zgolj posega po mehanizmih za preprečevanje ugibanja gesel (zaklepanje dostopa ob nepravilnih poskusih, t. i. »I'm not a robot« preverjanja ipd.). V duhu zgoraj naštetih načel so ti mehanizmi lahko pomanjkljivi, še posebno, če so odvisni od vsebine in parametrov, ki jih aplikaciji posreduje brskalnik. Veščega napadalca, ki manipulira s podatki, vse to ne bo ustavilo.

- Izogibanje avtentikaciji

Drugi pristop napadalcev pri avtentikaciji je njeno izogibanje. Po avtentikaciji začne aplikacija z internimi podatki in podatki, ki se prenašajo iz paketa v paket, slediti uporabniku. Razvijalci se sicer trudijo, da so ti podatki neuganljivi za napadalce. A v mehanizmih in logiki, ki naj bi to zagotovili, se skriva veliko potencialnih ranljivosti.

- Vrivanje »zlobnih« podatkov v vnosna polja

Potencialno težavo pomeni tudi funkcija na strani aplikacije, ki čaka na vnos uporabnika. Vnos uporabnika lahko vsebuje sistemske klice, ki funkcijo prisilijo k izvedbi česa zlonamernega. Preden se podatki predajo funkciji, se morajo torej ustrezno prečistiti. Marsikatera pomanjkljivost je povezana prav z nepravim za filtriranjem podatkov. Bodisi so razvijalci pozabili na filtriranje, bodisi se aplikacija preveč zanaša na filtriranje že na strani brskalnika. Lahko pa je na videz vse v redu, ampak je filtriranje vsebinsko pomanjkljivo. To slednje niti ni tako nenavadno, saj obstaja cela vrsta situacij, kjer lahko filtriranje preveč omejuje funkcionalnost aplikacije. Razvijalci morajo zares podrobno vedeti, kje in kaj želijo filtrirati. Nikoli ne podcenjujmo domišljije in znanja napadalcev glede podatkov, ki jih lahko posredujejo aplikaciji.



Smart Com varnostne storitve podrobneje predstavlja na spletni strani http://www.smart-com.si/varnostne-storitve/.

V omenjenih kontrolnih mehanizmih se skrivajo varnostne pomanjkljivosti, za katere je verjetno mnogo bralcev že slišalo. Recimo »SQL injection«, ki pride do izraza, ko se podatki iz vnosnega polja uporabljajo na zalednih bazah ter »Cross-site-scripting«, v primerih, ko aplikacija vsebino vnesenih podatkov prikazuje uporabnikom.

- Slabo skrbništvo aplikacij

Marsikatero pomanjkljivost lahko najdemo tudi pri skrbništvu aplikacij. Od tega, da je ranljiv strežnik, na katerem teče aplikacija, do tega, da aplikacija po nepotrebnem izdaja celo vrsto informacij (recimo ob napakah, ki jih napadalec namenoma sproži ipd.). Lahko pa napadalec na direktorijih najde pozabljene stare skripte.