Latvijas Universitāte DatZ6016 : Maģistra kursa darbs datorzinātnēs

Tīmekļa lietotnē integrējams WAF analizatora ietvars

Autors: Artūrs Kuļšs, arturs@latnet.lv, stud. apl. Nr.: ak16102 Darba vadītājs: Kārlis Podiņš, Mag. Dat., Doktorants

Ievads

Tīmeklī izvietotās lietotnes skar vitālas svarīgas dzīves jomas: finanses, sakarus, izglītību, veselību, informāciju un citas. Ir būtiski nepieļaut konfidenciālu datu noplūdi un lietotņu darbības traucēšanu, un to satura viltošanu. Tādēļ ir aktuāla tīmekļa lietotņu aizsardzība no kiberuzbrukumiem.
Eksistē risinājumi, kuri risina šo uzdevumu (WAF un RASP), tomēr katra tīmekļa lietotne ir unikāla, un katrai jaunai lietotnei aizsardzība ir jāpielāgo individuāli.
Darba autors piedāvā izmantot Rīgā, MII attīstītu *viedo tehnoloģiju* pieeju, un tās daļu – reālā laika biznesa procesa verifikatoru, tīmekļa lietotņu aizsardzībai.

Esošais risinājums: WAF – Tīmekļa lietotnes ugunsmūris
(*angl.: Web Application Firewall*).
Risinājums tiek novietots starp aizsargājamo tīmekļa serveri un savienojumu ar internetu Lai tas darbotos, ir jādefinē likumi, kā atšķirt datu paketes, kuras ir daļa no uzbrukuma serverim, un kuras nav. Šai pieejai ir savas priekšrocības, bet ir arī vairāki trūkumi:

- Ugunsmūrim nav pieejams lietotnes realizēto biznesa procesu konteksts – aizsardzības likumus ir jādefinē zemā, HTTP vaicājumu līmenī;
- WAF ir viens uz visiem savienojumiem – šaurā vieta veiktspējas kontekstā;
- HTTPS datu plūsmas apstrādei WAF ir jāpiekļūst SSL privātajai atslēgai;
- Datu kodējums paketēs var neļaut WAF saprast saturu un noreaģēt uz draudiem.

Esošais risinājums: RASP – Strādājošas lietotnes pašaizsardzība
(*angl.: Runtime Application Self Protection*).
Risinājums tiek izmitināts vienoti ar aizsargājamo lietotni. RASP kontroles paneļa aģents var būt kā tīmekļa servera spraudnis, ietvara modifikācija (piem. Java, .Net lietotnēs), un kā binära injekcija, ielādējot lietotnes kodu atmiņā.
Pieejai ir vairākas priekšrocības, salīdzinājumā ar WAF:

- Aģentiem ir pieejams lietotnes konteksts. Jebkurā koda vietā ir iespējams saņemt reālā laika datus, tos pārbaudīt un mainīt vadības plūsmu;
- RASP injicētais kods redz datus tā, kā tos redz lietotne – tie ir atkodēti un programmai saprotamā formā pieejami arī aģentam;
- Risinājums ir viegli mērogojams un pārnesams neskarot citas lietotnes.

Piedāvātais risinājums: BWAF – Biznesa procesu līmeņa lietotnes ugunsmūris
Risinājums tiek izmitināts neatkarīgi no aizsargājamās lietotnes.
Tas piekļūst lietotnes datiem netieši – ar aģentiem caur datubāzi, notikumu reģistrēšanas (*angl.: logging*) datnēm, un tieši – saņemot ziņojumus no lietotnes. Ir neatkarīgs no lietotnes platformas.
Lietotne augsta līmeņa biznesa datus sūta izmantojot BWAF API, Tas ļauj veikt:

- Notiekošo biznesa procesu plūsmas kontroli *lietotnes kontekstā*;
- Nesūtīt un neizpaust *konfidenciālu informāciju* ārpus lietotnes;
- Sarežģītāku biznesa likumu pārbaudi reālā laikā;
- Ērti ziņot par biznesa notikumiem no lietotnes caur BWAF API;
- Izvietot reālā laika biznesa procesu verifikatoru drošā datortīkla daļā.

Secinājumi

- Risinājuma izstrāde elastīgi pielāgojamam ugunsmūrim *biznesa procesu līmenī* ir aktuāla;
- Piedāvātais risinājums neaizstāj esošās pieejas, bet papildina tās;
- Ir pamats BWAF attīstīšanai izstrādāt maģistra darbu, izmantojot maģistra kursa darbā veikto izpēti un analīzi;
- Ir aktuāli atrast risinājumu automātiskai vai pusautomātiskai biznesa procesu līmeņa likumu iegūšanai reāllaika verifikatoram;
- Neviena no pieejām īsti nevar lepoties ar spēju adekvāti aizsargāt tīmekļa pakalpojumus (*angl.: web services*).