PaloAlto imelapsest tulemüür

6 aastat tagasi Autor: Ivar Pikker

PaloAlto firma loodi 2005a endise CheckPoint inseneri poolt. Iisraeli firma CheckPoint oli senini absoluutne liider küberturvalisuse valdkonnas ja ka esimese *stateful* tulemüüri looja1994. aastal (senine "uus põlvkond"). Nüüd on CheckPoint saanud PaloAlto näol omale tõsise konkurendi.

Nagu CheckPoint, on ka PaloAlto esindatud Gartner Inc Magic Quadrant edetabeli TOP-is. Ja mitte ainult. PaloAlto tulemüüri võib pidada täiesti uue põlvkonna tulemüüriks. Mitte sellepärast, et see on Next-Generation-Firewall (NGFW) kategooriasse kuuluv, vaid et see on ka sisuliselt midagi täiesti uut.

Kolm omadust, mis teevad PaloAltost uue põlvkonna tulemüüri

Laias laastus on olemas kolm omadust, mille alusel võib PaloAltot pidada uue põlvkonna tulemüüriks ja mis pole seni kasutusel olnud veel ühegi teise tulemüüri puhul. Need on portidest sõltumatu inspektsioon ja *stateful policy route* koos *return traffic´*u täpse tagasijuhtimisega (*Policy Based Forwarding*). Portidest sõltumatu inspektsioon hõlmab nii viirustõrjet, *Intrusion Prevention System´*it (IPS, sissetungiennetuse süsteem) ehk ka *vulnerability protection´*it kui ka rakenduste (aplikatsioonide) filtrit.

Tähelepanuväärne on, et oluliselt suuremat koormust protsessorile pole märgata isegi sellise põhjaliku inspektsiooni puhul.

PaloAlto tulemüür on monoliitne, haldamiseks ja võrguliikluseks on eraldi protsessorid. Nii riistvaraseadmetel kui ka ESXi virtuaalsel (tarkvaralisel) versioonil on täpselt sama tulemüür. Ja ka riistvaralistel mudelitel on see täpselt sama. Erinevus on vaid kasutatava riistvara jõudluses. Kõikidel teiste firmade tulemüüridel on pordid kinnistatud mingi konkreetse inspektsiooniga.

Standardpordid on saamas erandiks

Probleem on juba ammu selles, et standardsete portide kasutus on pigem erand kui reegel. Seni oli ainus võimalus, kuidas kontrollida ühendust kõigil portidel, kasutada veebiproksit ja muudel TCP-UDP portidel kasutada *socks-proxy´*t, mis on suhteliselt ebastabiilne nähtus. *Proxy* kasutus on pisut ebamugav, sest nõuab klientarvutite seadistamist ja on enamasti võimalik ainult veebibrauserite kaudu toimuva liikluse puhul.

PaloAlto tulemüüride tulekuga seega ka *proxy´*de roll viirusetõrjes kaob ja jääb ainult *cache* ehk vahepuhvri funktsioon.

Teine sisuline tehnoloogiline muutus on *Policy Based Forwarding*. See on täiesti unikaalne nähtus ja sellele puudub isegi analoog mõne teise tootja poolt. Tegemist on üldises plaanis *policy routing´*uga, mis on korraga nii *stateless* kui *stateful*. Esmapilgul on kõik *policy routing´*u reeglid *stateless*, siis reeglid mõjuvad nii sessioonile kui ka sessiooni nn *return* pakettidele, millel vahet ei tehta. *Stateful* seis ilmub peale "*Enforce Symmetric Return*" aktiveerimist (pane linnuke).

Selline seadistus teeb seda, et sessiooni tarbeks, mis seda reeglit kasutas, salvestatakse *state* tabelisse sessiooni info ja see garanteerib, et *return* paketid tulevad tagasi täpselt sama teed pidi, mida mööda sessioon loodi.

See *return* info on kõrgeima prioriteediga ja pakettide õige tagasitee on garanteeritud. *Policy route´i* reeglid aga jäävad endiselt *stateless* reegliteks, seega nad mõjuvad endiselt ka *return* pakettidele, mille sessiooni algatus oli seotud hoopis teise *policy route* reegliga või polnud üldse reeglit.

Selline olukord on enamus juhtudel muidugi lubamatu ja ebasoovitav. Enamikel juhtudel on *policy route* kautuse mõte suunata sessioone soovitud sihtpunkti ja sessiooni *return* paketid peaksid tagasi tulema sama reegli kontekstis, siis mitte mingi teise reegli kontekstis ja teist reeglit kasutades. Seda nimetatakse stateful *policy routing´*uks.

Tulemüüre on kahte liiki, sõltumata sellest et security-policid on kõigil stateful - on stateless ja stateful policy routinguga tulemüürid.

Stateless policy routingut võib siiski mingites eriolukordades vaja minna, kuid reeglina mitte.

CheckPointi policy routing näiteks on stateless ja seda smartcenteri kaudu ei saa üldse seadistada vaid ainult gateway webiinterface kaudu. Stateless policy routinguga tulemüürid on veel Mikrotik, Zyxel. Stateful policy routinguga aga Pfsense, Sophos, Fortigate.

Nüüd selleks, et panna PaloAlto *policy routing* soovitud viisil tööle, on tarvis teha veel üks *PolicyBasedForwarding* reegel.

Oletame, et on kaks sessiooni. Esimene sessioon on *policy routing* reegli poolt suunatud interfeisilt A interfeisile B. Teine sessioon on ilma reeglita ja on interfeisilt C interfeisile A.

Esimese sessiooni return pakettidega on kõik korras, kuna sai aktiveeritud "*Enforce Symmetric Retur*n", mis garanteerib õige tee *return* pakettidele.

Teise sessiooni *return* paketid aga lähevad valesti ja sessioon katkeb. Sest esimese sessiooni tarbeks loodud reegel mõjutab ka teise sessiooni *return* pakette (*stateless policy routing* reegel) ja suunab need interfeisilt A interfeisile B, mis on vale. Õige oleks interfeisilt A interfeisile C. Selleks tuleb ka teise sessiooni tarbeks teha reegel (kuigi sessiooni loomise ühendus leiab ka ilma reeglita õige tee) ja kindlasti aktiveerida "*Enforce Symmetric Return*". See tagab, et ka teise sessiooni *return* paketid tulevad õigesti tagasi ja esimese sessiooni tarbeks loodud reegel enam ei avalda mõju, sest "*Enforce Symmetric Return*" on kõrgema prioriteediga kui kõik teised *policy routing´*u reeglid.

Selle reegli järjekorranumber pole oluline ja selguse mõttes ma nimetan selle reegli "FIX" märkega nimes, tähendades seda, et reegel pole loodud mitte sessiooni enda juhtimiseks, vaid üksnes *return* pakettide tarbeks, isoleerides teiste reeglite mõju neile.

Jah, natuke keeruline on, aga sellega harjub ära

Nüüd aga küsimus, et miks siis ikkagi see *Policy Based Forwarding* nii tähelepanuväärne on. Teeks ju sama töö ära ka tavaline *stateful policy routing*, kus poleks tarvis üldse muretseda *return* pakettide pärast, kuna kõigis teistes *stateful policy routing´*uga tulemüürides reeglid üldse *return* pakettidele ei mõju ja *return* paketid tulevad tagasi automaatselt sessiooni tarbeks loodud *state* tabeli alusel.

Just siit areneb asi veel edasi.

Kõikides senistes *stateful policy routing* tulemüürides *return* pakette juhitakse tagasi state tabeli alusel, kus on kirjas interfeis, kust sessioon alguse sai. Kui pole lokaalne *subnet*, siis võetakse kasutusele *interface default gateway*. See paneb piirangu, et *interface* peab olema WAN-tüüpi tsoonis. *Policy route* reeglites saab kasutada küll suvaliselt määratud *gateway´*sid sessiooni juhtimiseks, kuid mitte *return* pakettide tarbeks. *Return* paketid juhitakse *interface default gateway´*sse. Nüüd aga kujutame olukorda, kus sessioon ei tulnud üldse *interface default gateway´*lt, vaid mõne teise lokaalse *subnet*i masina kaudu (origin aga kusagil veel kaugemal). Sellisel juhul automaatne *return* pakettide tagasi õige tee leidmine enam ei toimi. Vaja on teha staatilisse ruutimistabelisse kirje, et sessiooni origini juurde tagasi jõudmiseks tuleks minna mingi teise lokaalse subneti masina kaudu.

Kuid mis siis, kui origin pole üldse sisevõrgust, vaid on internetist?

Sellisel juhul internetiaadresside kohta pole võimalik staatilisse ruutimistabelisse midagi teha ja kogu ülesanne osutubki võimatuks. PaloAltol on siin samm edasi astutud ja ka *return* pakettidele on võimalik määrata igas reeglis eraldi *gateway* (*next-hop*). Reegli samas aknas üleval määrad ära *next-hop´*i sessiooni enda kohta ja sama akna allosas määrad ära *next-hop´*i *return* pakettide jaoks. Muidugi koos "*Enforce Symmetric Return*" linnukesega kastis, mis reeglina alati tuleb ka aktiveerida. See teeb PaloAlto tulemüürid täna kõige paindlikumateks ja suurimate võimalustega ruuteriteks.

Puudub ka tähtsus, millises *zone´*is mingi interfeis asub, PolicyBasedForwarding põhimõtteliselt võimaldab edukalt ruutida ükskõik millist ühendust ükskõik kuhu.

Kolmas omadus, mille puhul samuti võiks PaloAlto tulemüür pretendeerida uue põlvkonna tooteks, on virtuaalsed ruuterid. Siis on võimalik luua loogilisi ruutereid sama müüri sisse. Igaühel oma ruutimistabel koos *default gateway*ga ja võimalusega ruutida otse teise virtuaalsesse ruuterisse. Iga adminni oma vaba fantaasia võib määrata, mille tarbeks seda kasutada. Üks kasutusvõimalus on näiteks kui on tarvis panna mingi interfeis täpselt samasse subnetti, kui mingi teine interfeis. Siis teed talle eraldi virtuaalse ruuteri ja mingit konflikti enam pole.

Mis on vahet PaloAltol ja Checkpointil?

Nüüd kerkib küsimus, et mis siis veel on vahet PaloAltol ja CheckPointil, kumb on siis parem ja võimsam?

Ei saagi nii võrrelda, need on erinevad asjad. PaloAlto on kindlasti kaasaegsem, uuema põlvkonna tulemüür aga uute võimalustega. CheckPoint aga hiiglane peenhäälestuse osas, nagu näiteks "*Data Loss Prevention*", igasugu erinevad VPN-i nüansid, erinevad kasutajate autentimised ja IPS-peenhäälestus.

Võib öelda, et konkurents PaloAlto näol surus CheckPointi kindlalt ainult suurettevõtete (kontserni mõõtu ettevõtted või riigiasutused) nišši. Sest enamasti sellist peenhäälestust vajavad ainult suured kontsernid. Ja see, et tegemist on suurettevõttega ei ütle veel, et just CheckPointi tarvis on, vaid tarvis on ainult siis, kui sa kindlalt tead, et sul seda peenhäälestust tingimata vaja on. Muudel juhtudel rahuldab PaloAlto vabalt ka suurettevõtte vajadused.

Loomulikult on ka PaloAlto müüril olemas kõik VPN-id, VLAN-id, kasutajate autentimised, *data-loss-prevention´*id (*data filtering*) jne. Kaks asja, millest ise olen puudust tundnud PaloAlto puhul, on kokkusobimatus Google´i mobiiltelefoni PIN-kalkulaatoriga

kahefaktorilise autentimise kasutamisel ja *zoneedit* DDNS funktsionaalsus. Nagu juba eelnevalt sai mainitud, on riistvaralised seadmed ja virtuaalsed ESXi tarkvaralised versioonid sisuliselt identsed. Virtuaalse versiooni litsentsid on ainult kallimad. Kui CheckPoint väikeettevõttele on reeglina üle jõu käiv ost, siis näiteks PaloAlto PA-220 pisike karbike on täitsa optimaalne võrgu planeerimise komponent.

- Lahendused
- Võrguseadmed