Dizajn sustava za automatsku evaluaciju napada

Prvi korak u izgradnji bilo kakvog sustava jest donošenje odluke za što će sustav služiti. Ako je potrebno samo pratiti statistiku pristupa, radi se o sustavu za privlačenje napadača, i u tu svrhu se najčešće koriste niskointeraktivni honeypotovi; ukoliko se želi detaljnije istražiti napadačevo ponašanje, koristit će se sustav za praćenje napadača za koji su najprikladniji visokointeraktivni honeypotovi.

Idući korak jest odluka o smještaju honeypota – smješta li se honeypot na virtualni stroj ili stvarni fizički sustav. Za niskointeraktivne honeypotove je prikladnija virtualizacija sustava, jer su mogućnosti takvih honeypotova prilično skromne, dok je za visokointeraktivne honeypotove ponekad dobro odvojiti i cijeli fizički sustav.

Potom se odlučuje o operacijskom sustavu na kojem ćemo implementirati ovakav sustav, a najčešće odluke su Microsoft Windows ili neka distribucija operacijskog sustava Linux.

Zatim je potrebno odlučiti o sigurnosnim rupama koje se žele učiniti dostupnim napadačima. Kod sustava za privlačenje napadača nije potrebno stvoriti “stvarne” propuste, već samo usluge koje se čine ranjivima (šalju pozdravnu poruku s određenom verzijom ranjivog programa). Kod sustava za praćenje, potrebno je stvoriti stvarne propuste, čime dovodimo u opasnost cijeli sustav, a time i ostatak mreže. Obično se razmatraju tipične ranjivosti, poput HTTP, FTP i SMTP usluga, jer je takve najlakše naći, a često i iskoristiti. Osim onih usluga koje smo namjerno postavili, ostatak sustava mora biti čvrsto izoliran i otporan, kako neki napadač ne bi možda otkrio stvarnu ranjivost koja nije pod našim nadzorom.

Razmatranje za sustave za privlačenje napadača ovdje prestaje jer napadač nakon faze izviđanja i prikupljanja podataka ne može ostvariti pristup sustavu. No, sustave za praćenje napadača (i automatsku evaluaciju tehnika napada) je potrebno izolirati i iznutra. Naime, nakon što napadač iskoristi neku ranjivost, on se nalazi “fizički” na sustavu. U tom trenutku on može činiti što ga je volja – može pokušati zamesti tragove, napasti druge sustave koristeći trenutni sustav kao posrednika, uništiti sustav i slično. Kako bi se taj problem riješio, potrebno je onemogućiti sve izlazne veze sa sustava, pokrenuti praćenje svih pokrenutih naredbi, te nadzirati spremnički prostor (u slučaju instalacije novih alata i programa). No, ako se napadača previše izolira, on može posumnjati u ispravnost sustava, te ga napustiti prije nego što se prikupi dovoljno podataka. To bi se moglo riješiti na način da se zamijeni skup naredbi i skup datoteka nad kojima rade, kako bi se napadaču dao privid da se naredba uspješno izvršila, dok u stvarnosti sustav ostaje nepromijenjen. To zahtijeva značajnu intervenciju sa strane osobe koje implementira ovaj sustav, jer je potrebno napraviti “nove” (odnosno modificirane) naredbe. Potrebno je pratiti sav mrežni promet kako bismo dobili uvid u daljnje napadačeve namjere.

Konačno, sustav je potrebno spojiti na Internet i čekati napadače. Kako bi se napadi kasnije automatski evaluirali i analizirali, moguće je upotrijebiti metode strojnog učenja kako bi se napravili klasifikatori kvalitete napada. Osim toga, metodama strojnog učenja bilo bi moguće i profilirati napadače. No, za učenje modela potrebno je imati velik uzorak napada svakog pojedinog napadača, kako bi se mogle izdvojiti karakteristike napada.