Sustavi za privlačenje i detekciju napadača

Sustavi za privlačenje i detekciju napadača služe kako bi se namamili potencijalni napadači zbog otkrivanja načina napada na sustav. Obično je cilj da takav sustav izgleda vrlo bitno na mreži kako bi bio dovoljno interesantan, ali i vrlo ranjiv. Tako bi se veći broj napadača mogao odlučiti za napad na sustav. Ovakvi sustavi se obično koriste za sakupljanje kvantitativnih (količinskih) podataka i proučavanje općenitih smjerova napada. Jedna od glavnih karakteristika ovih sustava, kao i sustava za praćenje napadača, jest da ti sustavi nemaju apsolutno nikakvu proizvodnu vrijednost – odnosno, sva aktivnost na tim računalima je ilegalna (Provos, Holz, 2002).

Zbog velikog porasta broja računala na Internetu, sve je lakše pronaći potencijalne mete, jer mnoga računala nisu pravilno konfigurirana. Sve što je potrebno jest napraviti automatizirano skeniranje za upaljenim računalima (engl. ping sweep), a zatim skenirati njihove otvorene mrežne priključke (engl. port). Ukoliko je neki servis krivo konfiguriran, ili je instalirana starija inačica servisa, postoji velika mogućnost da isti bude kompromitiran. Statistička istraživanja sprovedena 2000. godine na sustavima za privlačenje i detekciju napadača pokazuju da je očekivano vrijeme života jednog prosječnog sustava sa nekom distribucijom Linux operacijskog sustava na otvorenoj mreži manje od 72 sata, dok se u računala s instaliranim operacijskim sustavom Windows 2000 provaljivalo u roku od petnaestak minuta (Spitzner, 2002). Ta statistika jest poražavajuća, ali s gledišta da se zapravo želi privući napadače, to je vrlo pozitivno – ne mora se dugo čekati da se dobiju neki opipljivi rezultati.

Idealan sustav osmišljen samo za privlačenje i detekciju napadača ne bi trebao nuditi ništa više nego vanjski izgled sustava. Takvi sustavi obično emuliraju postojanje ranjivih usluga i servisa na sustavu, kako bi napadač dobio dojam da je sustav lako kompromitirati. No, budući da su usluge emulirane, ukoliko se pokušaju kompromitirati, sustav neće uistinu biti ugrožen, jer zapravo nikad nije ni bio ranjiv (osim ako se ne pogriješi u konfiguraciji sustava). Emulacija usluga može i ne mora biti visoke kvalitete, ovisno o tome kakvu vrstu podataka tražimo. Recimo, ukoliko je cilj saznati koliko će vremena proteći prije nego što će neki napadač postati svjestan našeg sustava na mreži tada emulacija usluga skoro nije ni potrebna. No, ukoliko je potrebno izmjeriti statistiku korištenja pojedinih vrsta skeniranja (npr. alatom nmap), ili koji se program za iskorištavanje ranjivosti (engl. exploit) najčešće koristi za dotičnu usugu ili servis, tada je potrebno emulirati usluge s većom razinom uvjerljivosti. Ipak, to ne znači da je potrebno implementirati cijelu specifikaciju za svaku uslugu, nego samo podskup funkcija i radnji koje čine sustav dovoljno uvjerljivim (npr. slanje bannera usluge, odgovaranje porukama na određene zahtjeve, ali bez izvršavanja zahtjeva na sustavu i sl.). Ovakvi sustavi su upravo zbog svoje niske razine interakcije nazvani low-interaction honeypots, i u pravilu su prilično jednostavni za izgradnju i održavanje, a nude i mnogo manju razinu rizika jer sustav ne može biti u potpunosti kompromitiran.

Jedan važan detalj u implementaciji ovakvih sustava (kao i sustava za praćenje napadača koji će biti obrađeni u narednom poglavlju) jest odabir vrste sustava – želi li se odvojiti posebno računalo na koje se instalira operacijski sustav i emulirane usluge ili se na nekom računalu koje služi potpuno drugoj svrsi samo dodatno instalira još jedno virtualno računalo koje će emulirati cijeli sustav. Prvi način implementacije se naziva fizički, a drugi virtualni. Za implementaciju sustava za privlačenje i detekciju napadača obično su pogodniji virtualni sustavi jer nije potrebno emulirati svaku uslugu potpuno detaljno, a također nije potrebno ni napadaču dati pristup do drugih resursa sustava, što znači da se može koristiti konfiguracija s minimalnim zahtjevima na „hardware" sustava (u pogledu memorije, brzine procesora i slično). Također, ovakve sustave je lakše održavati, jer imaju mnogo manju funkcionalnost, a i u slučaju sigurnosnog ispada ili „kvara", moguće je vrlo brzo uspostaviti novi virtualni sustav. Pomoću virtualizacije moguće je napraviti i velike mreže honeypotova, koje se onda nazivaju honeynets ili honeyfarms. Očevidno je da je poprilično nepraktično držati stotinjak fizičkih

računalnih sustava posvećenih samo detekciji i praćenju napadača, dok je moguće istu toliko količinu uređaja virtualizirati na, primjerice, desetak sustava.

Postoji jako velik broj komercijalnih i nekomercijalnih rješenja za niskointeraktivne honeypotove, od kojih su neki honeyd, LaBrea, Tiny Honeypot, te mnogi drugi. TinyHoneypot je vrlo jednostavan program, kojem se nakon instalacije „kaže" (upisom u konfiguracijsku datoteku) koje se usluge želi „upaliti". Nakon pokretanja usluga, svatko tko se spoji na dotične usluge dobije banner usluge i prividnu root korisničku ljusku (ljusku s administratorskim ovlastima). Poanta ovog trika je iskoristiti socijalni inženjering – probati uvjeriti napadača da mu se nekako posrećilo, u nadi da će on pokušati dalje kompromitirati sustav (npr. skidanjem rootkita s neke stranice). TinyHoneypot prati korisnikovo ponašanje (bilježi sav unos) za kasniju analizu. LaBrea je program koji pokušava maksimalno usporiti napadača korištenjem trikova u TCP komunikaciji (npr. postavljanjem vrlo male veličine prozora za primanje). Ovaj program se koristi najčešće kako bi se usporili spammeri (svaka aktivna konekcija sa spammerovog sustava znači da mu je preostalo manje resursa za slanje spama prema drugim sustavima). Honeyd je framework koji služi za prikaz cijele mreže ranjivih „sustava" (npr. par tisuća). Za svaki „sustav" jednostavno je postaviti pravila ponašanja dotičnog sustava (možemo emulirati karakteristike TCP/IP sloga određenih operacijskih sustava s određenim upaljenim uslugama). Ovo je jedan od najčešće korištenih nekomercijalnih programa za detekciju napada.