HRVATSKA AKADEMSKA I ISTRAŽIVAČKA MREŽA
CROATIAN ACADEMIC AND RESEARCH NETWORK

Analiza Stumbler/55808 trojanskog konja

CCERT-PUBDOC-2003-07-31

CARNet CERT u suradnji s LS&S

Sigurnosni problemi u računalnim programima i operativnim sustavima područje je na kojem CARNet CERT kontinuirano radi.

Rezultat toga rada ovaj je dokument koji je nastao suradnjom CARNet CERT-a i LS&S-a, a za koji se nadamo se da će Vam koristiti u poboljšanju sigurnosti Vašeg sustava.

CARNet CERT, www.cert.hr - nacionalno središte za **sigurnost računalnih mreža** i sustava.

LS&S, www.lss.hr - laboratorij za sustave i signale pri Zavodu za elektroničke sustave i obradbu informacija Fakulteta elektrotehnike i računarstva Sveučilišta u Zagrebu.

Sadržaj

1. Uvod

Mapiranje mreža i skeniranje portova (engl. *network mapping*, *port scanning*) uobičajene su tehnike koje neovlašteni korisnici upotrebljavaju prilikom planiranja napada na računalne sustave. Prve alate za identifikaciju računalnih sustava i pregledavanje portova bilo je prilično jednostavno otkriti, budući da su koristili primitivne i lako uočljive metode. S vremenom su tehnike i alati koje neovlašteni korisnici upotrebljavaju u ovu svrhu znatno napredovali, tako da novije generacije u pravilu koriste različite tehnike kojima se omogućuje prikrivanje malicioznih aktivnosti.
Ovaj dokument sadrži kraći pregled poznatijih metoda skeniranja te analizu novog distribuiranog trojanskog konja pod nazivom 55808/*Stumbler*, koji se služi upravo jednom od tehnika. Nedavno provedene analize pokazale su da je upravo Stumbler trojanski konj odgovoran za povećani intenzitet malicioznog prometa na Internetu. Promet je specifičan po veličini TCP Window parametra koji u ovom slučaju iznosi 55808.

2. Tehnike skeniranja

2.1. TCP *connect()* pregledavanja portova

Ovo je najjednostavniji oblik skeniranja pomoću TCP protokola. TCP *Connect()* sistemski poziv otvara vezu prema proizvoljno odabranom TCP mrežnom portu na udaljenom računalu. Ukoliko je port u stanju *listen*, TCP *connect()* poziv će uspjeti, dok je u suprotnom slučaju port nedostupan. Budući da nisu potrebne posebne ovlasti za korištenje TCP *connect()* poziva, ova metoda pregledavanja mrežnih portova ne zahtjeva administratorske ovlasti.

2.2. TCP SYN pregledavanja portova

Ova tehnika skeniranja pomoću TCP protokola često se naziva i "polu-otvorena" ("*half-opened*"), budući da se u ovom slučaju ne dovršava postupak uspostave TCP veze u tri koraka (engl. *three way handshake*). Pregledavanje portova se izvodi tako da se prvo na ciljni port pošalje SYN TCP paket. Ako računalo odgovori s TCP SYN/ACK paketom, to znači da je port otvoren, dok TCP RST paket znači da je port zatvoren. U prvom slučaju odmah se šalje TCP RST paket na otvoreni port da bi prekinuli vezu na samom početku. Glavna prednost ove tehnike je mala vjerojatnost otkrivanja, no za korištenje TCP SYN paketa potrebne su administratorske ovlasti na računalu.

2.3. TCP FIN pregledavanja portova

Za razliku od TCP SYN tehnike skeniranja koja može biti otkrivena vatrozidima ili alatima za praćenje i bilježenje TCP SYN paketa (poput *synlogger-a* ili *Courtney*), TCP FIN tehnika skeniranja gotovo se i ne može otkriti. Na TCP FIN paket zatvoreni port će odgovoriti TCP RST paketom, dok će ga otvoreni port u pravilu ignorirati. Iako ovakvo ponašanje u potpunosti zadovoljava specifikaciju TCP protokola, neki operacijski sustavi (većinom Windows) šalju TCP RST paket bez obzira na stanje porta. Ova tehnika skeniranja može se, dakle, osim za pregledavanje portova iskoristiti i za identifikaciju operacijskog sustava udaljenog računala.

2.4. Pregledavanje potova fragmentacijom IP paketa

Metoda pregledavanja mrežnih portova fragmentacijom IP paketa modifikacija je ranije opisanih tehnika pregledavanja baziranih na TCP protokolu. Umjesto slanja jednog TCP/IP paketa za utvrđivanje stanja mrežnih portova, šalje se nekoliko IP fragmenata istog paketa. TCP zaglavlje dijeli se u nekoliko manjih paketa pa se time otežava posao alatima za filtriranje mrežnog prometa i detekciju neovlaštenih aktivnosti.

2.5. TCP *reverse ident* pregledavanje portova

Protokolom *ident* (RFC 1413) moguće je doći do korisničkog imena vlasnika bilo kojeg procesa spojenog preko TCP protokola, neovisno o tome da li je on inicirao vezu. Tako se na primjer

neovlašteni korisnik može spojiti na *http* port te iskoristiti *ident* da bi utvrdio da li je vlasnik servisa *root* korisnik.

2.6. FTP *bounce* napad

Budući da FTP protokol podržava "*proxy*" veze moguće je npr. inicirati konekciju s adrese izvor.hr na FTP *server-PI* (*protocol interpreter*) na adresi meta.hr te tako uspostaviti kontrolnu vezu. Nakon toga pomoću *server-PI*-a moguće je pokrenuti *server-DTP* (*data transfer process*) te poslati datoteku na bilo koju udaljenu IP adresu. Na većini današnjih FTP poslužitelja napadi poput opisanog, naravno, nisu mogući jer bi to značilo da možemo poslati *mail*-ove kojima je nemoguće ući u trag ili napuniti nečiji disk nepotrebnim podacima. Ova tehnika ipak može poslužiti za pregledavanje mrežnih portova putem FTP poslužitelja koji se nalazi iza npr. vatrozida.

2.7. UDP ICMP *port unreachable* pregledavanje portova

Ova tehnika skeniranja, umjesto TCP protokola, koristi UDP protokol. Većina računala šalje ICMP_PORT_UNREACH poruku o grešci ako je UDP paket poslan na zatvoreni port pa je na temelju toga moguće utvrditi stanje porta na udaljenom računalu. Budući da nije garantirano da će UDP i ICMP paketi stići do odredišta, UDP skeneri moraju imati mehanizam za ponovno slanje paketa za koje se pretpostavlja da su izgubljeni. Za korištenje ove tehnike potrebne su administratorske ovlasti.

2.8. UDP *recvfrom()* i *write()* pregledavanje portova

Korisnici koji nemaju administratorske ovlasti mogu doći do informacija o zatvorenim i otvorenim portovima preko UDP protokola ako pošalju *write()* komandu na zatvoreni port. Ako je port zatvoren rezultat će biti *fail* poruka.
Recvfrom() komanda od neblokirajućih UDP *socketa* obično dobiva EAGAIN ("*try again*") odgovor ako ICMP greška nije primljena i ECONNREFUSED ("*connection refused*") ako je primljena.

3. Analiza trojanskog konja Stumbler/55808

Internet Security Systems organizacija (http://www.iss.net/) nazvala je ovog trojanskog konja *Stumbler*, a zabilježeni su i njegovi klonovi poput *Intrusec-ova* 55808.a. U oba slučaja radi se o distribuiranom skeneru koji za pregledavanje mrežnih portova koristi ranije opisanu TCP SYN tehniku skeniranja portova (Poglavlje 2.2).
Inačice Stumbler trojanskog konja zasad su primijećene samo na Linux operacijskim sustavima iako se s obzirom na portabilnost koda može pretpostaviti da postoje i inačice za druge operacijske sustave.
Prilikom skeniranja portova program lažira izvorišnu adresu inicijalnih TCP SYN paketa kako bi se na taj način prikrio izvor skeniranja i otežao postupak identifikacije računala zaraženih Stumbler trojanskim konjem. Izvorišna IP adresa koja se koristi za lažiranje TCP SYN paketa odabire se nasumično, kao i odredišna IP adresa računala čiji se portovi pregledavaju. Kako bi se dodatno prikrio izvor skeniranja, Stumbler lažira i Ethernet MAC adresu inicijalnih TCP SYN paketa.
S obzirom da je izvorišna IP adresa TCP SYN paketa lažirana, program nije u mogućnosti primiti TCP SYN-ACK odgovor ciljnog računala na temelju kojega je moguće utvrditi stanje ispitivanoga porta.
U tu svrhu *Stumbler* trojanski konj osim slanja TCP SYN paketa osluškuje i mrežu, ne bi li uočio pakete veličine prozora (engl. *Window size*) 55808, koje su generirali upiti ostalih *Stumbler* trojanskih konja prilikom pregledavanja portova, na nasumično odabranim računalima na Internetu. Kako raste broj računala inficiranih *Stumbler* trojanskim konjem, raste i broj dobivenih informacija o otvorenim portovima na udaljenim računalima. Kada prepozna takav paket, *Stumbler* bilježi podatke o IP adresama i otvorenim portovima u datoteku na lokalnom tvrdom disku te ih jednom dnevno šalje na unaprijed definiranu IP adresu: 12.108.65.76, TCP port 22 . Pri tome koristi uobičajene mrežne biblioteke "*Libpcap*" i "*Libnet*".
Unaprijed definiranu IP adresu na koju *Stumbler* šalje snimljene podatke o stanju mrežnih portova skeniranih računala moguće je promijeniti slanjem posebnog paketa na segment računalne mreže na kojoj *Stumbler* trojanski konj prati mrežni promet. Sekvencijalni broj takvog TCP paketa sadrži podatke o novoj IP adresi, na koju će *Stumbler* ubuduće slati zabilježene podatke. Ova funkcionalnost u sadašnjim verzijama *Stumbler* trojanskog konja nije aktivirana.

Informacije o skeniranim računalima i portovima nalaze se u datoteci "r" trenutnog direktorija (obično /tmp/…/), a sam *Stumbler* program identificiran je kao datoteka pod nazivom "a" u istom direktoriju.
Stumbler ne sadrži programski kod kojim bi omogućio samostalnu instalaciju na računalo, već mora biti instaliran ili ručno ili preko nekog drugog malicioznog programa koji nije u izravnoj vezi sa samim *Stumblerom*. Prisutnost trojanskog konja prilično je lako otkriti budući da generira iznimno velike količine mrežnog prometa sa TCP paketima veličine prozora 55808.
Osim povećanja mrežnog prometa ovaj trojanski konj ne izaziva druge maliciozne posljedice, a njegove mogućnosti pregledavanja računalnih sustava koriste se samo za prikupljanje podataka. Moguća opasnost mogla bi se pojaviti u slučaju većeg širenja Stumbler trojanskog konja, kada bi oni postali vrlo efikasna platforma za neovlašteno distribuirano skeniranje računala i mrežnih portova. Prikupljeni podaci mogli bi se upotrijebiti za npr. DoS (engl. *Denial of Service*) napade ili napade kontroliranim *bot* programima.

4. Metode zaštite

Preporuča se nadgledanje TCP konekcija prema IP adresi 12.108.65.76, port 22. Uz kontrolu TCP konekcija, prisutnost *Stumbler* trojanskog konja može se otkriti i ukoliko su na računalu prisutne datoteke /tmp/…/a i /tmp/…/b.

5. Zaključak

U ovom dokumentu iznesena je analiza *Stumbler* trojanskog konja koji je karakterističan po svojoj distribuiranoj TCP SYN tehnici pregledavanja mrežnih portova. Za sada *Stumbler* nema ozbiljnijih malicioznih posljedica, osim povećanja TCP prometa na mreži, pa se može zaključiti kako mu je jedina funkcija bilježenje podataka o stanju portova i računala na mreži. Iako trenutno ne predstavlja veću opasnost zbog malog broja instaliranih trojanskih konja, u budućnosti bi unaprjeđenjem tehnike širenja ovaj trojanski konj, zbog informacija o mreži koje prikuplja, mogao postati osnova za druge vrste napada na računalne sustave.