Sniffing v rukou správce sítě

Vložil/a Thorough [1], 12 Únor, 2007 - 12:51

- Networks & Protocols [2]
- Security [3]

Sniffing je často spojován s aktivitou hackera, který odposlouchává data nic netušících uživatelů. Pojďme se však podívat na praktické ukázky, kdy sniffing pomáhá řešit problémy.

V naší modelové síti je brána (router- PC s Linuxem) připojená rozhranim eth0 k internetu a eth1 do LAN. Na eth0 je veřejná IP adresa kterou pridělil provider a na eth1 IP 1.1.1.1 + nahozená síť 1.1.1.0/24 (PC v síti používají IP adresy 1.1.1.2 - 1.1.1.254 s defaultní bránou 1.1.1.1). Rozhraní eth1 je propojeno s ostatními PC v síti switchem u něhož předpokládejme, že je vždy OK. IP adresu 1.1.1.2 používá uživatel kterému stále něco nefunguje :) a na IP adrese 1.1.1.254 je spuštěn veřejně přístupný webový server (na routeru 1.1.1.1 se provádí forward portů).

Použitým snifferem je program tcpdump (www.tcpdump.org [4]), který je součástí mnoha distribucí linuxu. Jeho popis nechme stranou, avšak vysvětlení námi použitých parametrů (options) je důležité:

n - ve výpisu nepřekládá IP adresy na hostname
i - označení interface
port - číslo sledovaného portu
host - v našem případě IP adresa sledovaného PC
w - v našem případě se nic nebude ukládat do mezipaměti

Tcpdump zobrazuje IP adresy a porty ve formátu ip_adresa.port.. Výpisy programu tcpdump níže jsou zkráceny.

Nyní už k praktickým ukázkám.

Uživatel s IP adresou 1.1.1.2 si stěžuje, že mu nefunguje připojení k internetu a upřesňuje, že se mu nedaří zobrazit webové stránky.

Na bráně si spustíme v jednom okně terminálu PING na klientovu IP adresu a ve druhém okně tcpdump:

tcpdump -ni eth1 host 1.1.1.2

V okně terminálu můžeme vidět tento výstup:

```
10:24:59.037870 arp who-has 1.1.1.2 tell 1.1.1.1
10:25:00.031154 arp who-has 1.1.1.2 tell 1.1.1.1
10:25:01.031155 arp who-has 1.1.1.2 tell 1.1.1.1
```

Výpis znamená, že zdrojový počítač s IP adresou 1.1.1.1 posílá protokolem ARP přes broadcast (ke všem počítačům) požadavek na získání MAC adresy cílového počítače s IP 1.1.1.2, ten však nevrací odpověď. Problém je na úrovní síťové vrstvy.

Pokud se na odeslaný PING (ICMP Echo request) vrací odpověď (ICMP Echo reply) což vypadá v tcpdump takto:

```
10:43:21.691169 IP 1.1.1.1 > 1.1.1.2: ICMP echo request .....
```

```
10:43:21.691499 IP 1.1.1.2 > 1.1.1.1: ICMP echo reply .......
10:43:22.692784 IP 1.1.1.1 > 1.1.1.2: ICMP echo request .....
10:43:22.693112 IP 1.1.1.2 > 1.1.1.1: ICMP echo reply .......
```

... tak je potřeba prozkoumat komunikaci na úrovni aplikace uživatele. Tady už se jedná o přenos dat mezi TCP nebo UDP porty. Na bráně (IP 1.1.1.1) si opět spustíme tcpdump:

tcpdump -ni eth1 host 1.1.1.2 and port 80 // výraz "and" znamená spojení filtru (v našem případě "host" a "port")

V době kdy je tcpdump spuštěn si uživatel zkusí webovým prohlížečem načíst stránku webu o kterém víme, že je právě funkční - např. google.com.. Pokud tcpdump nezobrazí žádná data, tak padá podezření na špatně nastavený webový prohlížeč, který neposílá na webový server žádné požadavky. Správná komunikace by v tcpdump vypadala takto:

```
10:54:34.913365 IP 1.1.1.2.37329 > 209.85.129.99.80: . ack 1431 win 8580
10:54:34.913571 IP 209.85.129.99.80 > 1.1.1.2.37329: P 1431:2545(1114) ack 692 win
6687
10:54:34.913758 IP 209.85.129.99.80 > 1.1.1.2.37329: P 2545:2702(157) ack 692 win
6687
10:54:34.913827 IP 209.85.129.99.80 > 1.1.1.2.37329: F 2702:2702(0) ack 692 win 6687
```

Ve výpisu je důležité písmeno "P" (PUSH), což je hodnota tcpflags a znamená, že paket nese data. Důležité je taky to, že jde komunikace oběma směry. Písmeno "F" znamená ukončení spojení (FIN).

To že je jinak připojení v pořádku si ještě můžeme potvrdít tak, že se podíváme na komunikaci uživatelského PC s DNS serverem. Spustíme si na bráně tcpdump tak, aby sledoval komunikaci na portu 53 (DNS):

tcpdump -ni eth1 host 1.1.1.2 and port 53

Uživatel si "pingne" hostname stroje, k terým od posledního spuštění PC ještě nekomunikoval (resolver ho nema v cache - např. security-portal.cz :)). Tcpdump by měl zobrazit tato data:

```
11:07:22.547417 IP 1.1.1.2.41179 > 81.30.225.2.53: 6668+ A? security-portal.cz. (30)
11:07:22.556693 IP 81.30.225.2.53 > 1.1.1.2.41179: 6668* 1/3/3 A 87.236.197.80 (157)
```

Ve výpisu je vidět, že se PC s IP adresou 1.1.1.2 dotazuje DNS serveru s IP adresou 81.30.225.2 na A záznam domény security-portal.cz.. DNS server odpovídá, že se jedná o IP adresu 87.236.197.80..

Tím jsme došli k tomu, že uživateli nefunguje webový prohlížeč a kauzu nefunkčního připojení můžeme uzavřít.

Podívejme se na jiný modelový příklad. Na stroji s IP adresou 1.1.1.254 je spuštěn webový server, na bráně 1.1.1.1 se provádí forward portů. Správce webového serveru si stěžuje, že najednou nemá připojeného žádného uživatele a žádá nápravu.

Opět pomůže tcpdump spuštěný na bráně tak, aby zobrazoval komunikaci týkající se IP 1.1.1.254 a (and) portu 80 (HTTP):

tcpdump -ni eth1 host 1.1.1.254 and port 80

V takovém případě bývají nejčastěji vidět tyto dva výpisy:

```
12:09:03.525242 IP 81.50.25.256.2533 > 192.168.1.254.80: S .....
12:09:03.525318 IP 192.168.1.254.80 > 81.50.25.256.2533: R .....
```

```
12:09:03.525242 IP 81.50.25.256.2533 > 192.168.1.254.80: S .....
12:09:03.525318 IP 192.168.1.254.80 > 81.50.25.256.2533: R .....
12:09:03.525242 IP 81.50.25.256.2533 > 192.168.1.254.80: S .....
12:09:03.525318 IP 192.168.1.254.80 > 81.50.25.256.2533: R .....
```

Opět sledujeme hodnotu tcpflags. Písmeno "S" (SYN) znamená, že klient zahajuje spojení a písmeno "R" (RESET) že ho cílový systém aktivně odmítá. Důvod je zpravidla jednoduchý - webový server nebo TCP Wrap je shozen.

Může se zobrazit rovněž tento výpis:

```
12:19:03.525242 IP 81.50.25.256.2533 > 192.168.1.254.80: S .....
12:19:03.525242 IP 81.50.25.256.2533 > 192.168.1.254.80: S .....
12:19:03.525242 IP 81.50.25.256.2533 > 192.168.1.254.80: S .....
12:19:03.525242 IP 194.228.2.11.2533 > 192.168.1.254.80: S .....
12:19:03.525242 IP 194.228.2.11.2533 > 192.168.1.254.80: S .....
12:19:03.525242 IP 194.228.2.11.2533 > 192.168.1.254.80: S .....
```

Tady je vidět, že požadavky na webový server sice přicházejí, ale odezvy z něho neodchází. Důvodem bývá buď "vytuhlý" webový server, nebo špatně nastavený firewall.

Pokud by nebyla v tcpdump vidět žádná data, tak může být problém na bráně, pravděpodobně s forwardem portů.

Dejme si další modelový příklad - **na bráně běží monitorovací engine, který kreslí grafy**. Jednou z hodnot v grafu je rovněž množství odeslaných a přijatých paketů. Správce zjistil, že u PC s IP adresou 1.1.1.2 došlo k prudkému nárustu množství odeslaných a přijatých paketů, což nevěstí nic dobrého. Nejčastěji se jedná o tři příčiny - v uvedeném PC je vir který hledá oběti, exploit který prohledává síť a hledá službu kterou by mohl napadnout, nebo živatel skenuje porty. Opět pomůže tcpdump spuštěný na bráně:

tcpdump -ni eth1 host 1.1.1.2

Virus nebo exploit, který prohledává síť a hledá zranitelnou službu na portu 135:

```
12:26:53.959229 IP 1.1.1.2.1516 > 194.85.57.72.135: S .....
12:26:53.960435 IP 1.1.1.2.1518 > 194.85.57.71.135: S .....
12:26:53.990078 IP 1.1.1.2.1628 > 194.85.57.73.135: S .....
12:26:54.130167 IP 1.1.1.2.1416 > 194.85.57.46.135: S .....
12:26:54.194831 IP 1.1.1.2.1742 > 194.85.57.74.135: S .....
12:26:54.317808 IP 1.1.1.2.1746 > 194.85.57.75.135: S .....
```

Za pozornost stojí, že se jedná o odesílání dat "naslepo" (hodně pokusů o zahájení spojení) a že je cílový port jen jeden, v našem případě 135.

Skenování portů vypadá takto:

```
13:10:10.508508 1.1.1.2.60637 > 194.228.2.1.85: S .....
13:10:10.508582 194.228.2.1.85 > 1.1.1.2.60637: R .....
13:10:10.508509 1.1.1.2.60638 > 194.228.2.1.86: S .....
13:10:10.508591 194.228.2.1.86 > 1.1.1.2.60638: R .....
13:10:10.508510 1.1.1.2.60639 > 194.228.2.1.87: S .....
13:10:10.508598 194.228.2.1.87 > 1.1.1.2.60639: R .....
```

Tady je důležité, že se jedná o jednu cílovou IP adresu (194.228.2.1) a různé cílové porty (85, 86, 87

...). Dále je vidět, že probíhá komunikace s cílovým počítačem, který "žije" (viz tcpflags "R", tzn. aktivně odmítané připojení).

Jak najít počítač, který odesílá SPAM?

Pokud budeme mít štěstí, tak opět pomocí tcpdump. Předpokládejme, že uživatelé v síti používají pro odesílání pošty SMTP server, který běží na bráně 1.1.1.1.. Na bráně si spustíme:

tcpdump -ni eth1 port 25 and no host 1.1.1.1 // výraz "and no" znamená negaci, tzn. vyloučení ze sledování

Pokud některý z počítačů odesílá SPAM, tak se zobrazí bohatá komunikace s IP adresama v internetu na cílovém portu 25/TCP a zdrojová IP adresa, např. 1.1.1.2. Tady ale ještě jisté, že se jedná o SPAM. Pomůžeme si tak, že se podíváme, komu je pošta adresována.

Nejdříve si musíme uložit veškerou sledovanou komunikaci na disk, protože vysledovat cílové e-mailové adresy v paketech přímo v reálném čase prostě nejde :)) Tcpdump spustíme v tomto formátu:

tcpdump -ni eth1 host 1.1.1.2 and port 25 -s 0 -w /smtp.bin

Tcpdump tady zachytává komunikaci, ve které figuruje IP adresa 1.1.1.2 a port 25.. Žádná data se neukládají do mezipaměti a veškerá komunikace vyhovující pravidlům se ukládá do souboru /smtp.bin.. Rád bych tady vysvětlil parametr -s . Tcpdump bez toho parametru zachytává vždy 80 bytů z každého paketu, který postačí k analýze síťového provozu (IP, ICMP, TCP a UDP). Pokud u tohoto parametru použijeme hodnotu "0", tak tcpdump zachytává vše, co zachytit stihne.

Pokud se chceme do souboru /smtp.bin následně podívat, tak nedoporučuji použít textový editor, protože se rozhodí okno terminálu. Použijeme program strings, který dokáže ze souboru vytáhnout tisknutelné znaky a zobrazit je v terminálu. My však necheme vidět celý obsah souboru, takže výpis omezíme programem grep na řetězec "RCPT" což znamená, že se nám vypíšou všechny řádky, kde se tento řetězec nachází. Příkaz bude následující:

strings /smtp.bin | grep RCPT

Vidět bychom mohli něco podobného:

```
RCPT TO: <niuqoeaj@earthlink.com>
RCPT TO: <www.skip@longfellow.com>
RCPT TO: <r.tessier@infoteck.qc.ca>
RCPT TO: <56b69.00097124@ose.com.tw>
RCPT TO: <rodgers@falconjet.com>
RCPT TO: <cshearer@spss.com>
RCPT TO: <asbury.edunhim@asbury.edu>
RCPT TO: <trevelene.coyle@ngc.com>
RCPT TO: <vhunt@partners.org>
```

Tady už hraničí podezření s jistotou, že nám ze sítě odchází SPAM.

Tak a na závěr si **pojďme vydloubnout ze síťové komunikace nějaké loginy**. Pro pochopení si to opět ukažme na uživateli s IP adresou 1.1.1.2 který si stěžuje, že se nemůže připojit k našemu POP3 serveru, který běží na bráně (IP 1.1.1.1). Spustíme si tcpdump:

tcpdump -ni eth1 host 1.1.1.2 and port 110 -s 0 -w /pop.bin

Nyní tcpdump ukládá do souboru /pop.bin komunikaci, ve které figuruje IP adresa 1.1.1.2 a port 110. Požádáme uživatele, aby se několikrát pokusil připojit k POP3 serveru (stáhl si poštu). Jakmile to udělá, tak se podíváme do souboru tímto příkazem:

strings /pop.bin | grep -E "user|pass"

Tímto příkazem si zobrazíme řádky ze souboru /pop.bin, které obsahují řetězec user nebo pass. Výsledek může být následující:

```
ZPuser zajda
hpass 3635
user zajda
pass 3635
```

Nyní stačí porovnat přihlašovací údaje které používá klient s těmi které jsou nastaveny na serveru.

No a možnosti jsou různé. Odposlechneme si na bráně a interface které je připojeno k poskytovateli připojení k internetu všechnu odchozí poštu. Tu si uložíme do souboru smtp.bin.. Následně si do souboru smtp.txt vytáhneme jen tisknutelné znaky (resp. čitelný text) a tento soubor si odešleme na svůj server:

```
tcpdump -ni eth0 dst port 25 -s 0 -w /smtp.bin
strings /smtp.bin > /smtp.txt
scp /smtp.txt muj_server:/
rm /smtp.bin
rm /smtp.txt
```

URL článku: https://security-portal.cz/clanky/sniffing-v-rukou-spr%C3%A1vce-s%C3%ADt%C4%9B

Odkazy:
[1] https://security-portal.cz/users/thorough
[2] https://security-portal.cz/category/tagy/networks-protocols
[3] https://security-portal.cz/category/tagy/security
[4] http://www.tcpdump.org