Zjišťování konfigurace zařízení, analýza Wireshark

ISA - Laboratorní cvičení č.1

Vysoké učení technické v Brně

`https://github.com/nesfit/ISA/tree/master/wireshark`

Cíle cvičení

- Seznámení se se základní prací v OS Linux.
- Seznámení se se základními nástroji pro zjišťování konfigurace zařízení.
- Analýza síťového provozu pomocí Wireshark.

Pokyny

- Do zadání nepište, slouží pro další skupiny. K zápisu používejte pouze záznamový arch, který vám zůstane.
- Pro práci v laboratoři budeme používat OS Linux, při bootu volba F3.
- Uživatelé a hesla pro přihlášení: `user - user4lab`, `root - root4lab`.

1 Zadání

Přihlaste se jako uživatel `user`. Veškeré potřebné příkazy následně spouštějte jako `root`.

1.1 Zjišťování konfigurace

V případě, že se v OS Linux úplně neorientujete, přečtěte si sekci 1.1 Základní orientace v Linuxu. V této části se budeme zabývat převážně zjišťováním síťové konfigurace systému. Veškeré potřebné informace, které budete potřebovat ke splnění této části cvičení, naleznete v sekci 1.2 Síťová konfigurace. V případě, že si nejste jistí některým příkazem, neváhejte nahlédnout do manuálové stránky.

1. Vypište konfiguraci vašeho stroje (IP adresu, masku, síť, broadcastovou adresu).
2. Zobrazte si záznamy v routovací a ARP tabulce. Zapište pravidlo týkající se defaultní cesty. Následně k IP ve vypsaném záznamu přiřaďte MAC adresu. Jaký je rozdíl mezi defaultní cestou a defaultní bránou?
3. Otestujte konektivitu k default gateway a následně konektivitu do internetu.
4. Vypište implicitní servery DNS a soubor, ve kterém jste tuto informaci nalezli.
5. Upravte patřičný soubor tak, aby po spuštění příkazu `ping google`, byl ping proveden vůči IP adrese 8.8.8.8. Zapište jak a který soubor jste upravili.

6. Vypište aktivní TCP spojení, vyberte jeden záznam, zapište si ho a popište význam jednotlivých položek. Pokud se žádné TCP spojení nezobrazuje, nějaké vygenerujte, například pomocí webového prohlížeče.

7. Zobrazte systémové události.

8. Zobrazte pouze události týkající se NetworkManager.

9. Pokuste se spustit Wireshark jako `sudo`. Následně nalezněte v logu zprávu, která byla zaznamenána v případě neúspěšného přihlášení.

1.2 Wireshark

V této části cvičení se budeme zabývat analýzou a zachytáváním provozu v programu Wireshark. Spuštění Wireshark provedete příkazem `wireshark` pod uživatelem `root`. Veškeré potřebné informace, které budete potřebovat k této části cvičení, naleznete v sekci 2 Analýza Wireshark.

1. Pomocí programu Wireshark začněte zachytávat pouze HTTP komunikaci. Jakmile začnete zachytávat, spusťte si prohlížeč a načtěte stránku `http://www.fit.vutbr.cz`. Zachycený provoz uložte do svého domovského adresáře. (přípona `.pcap`)

2. Zahajte znovu zachytávání komunikace, nyní bez použití filtru pro HTTP. V příkazové řádce odstraňte ARP záznamy (příkaz `arp ...` nebo `ip neightbour ...`). Ve wiresharku zobrazte veškerou komunikaci, následně vyfiltrujte pouze ARP a ICMP pakety. Vygenerujte ICMP komunikaci. Analyzujte obsah ARP paketů, zapište obsah sender, target MAC a IP adresy pro dvojici ARP request a response. Zapište, co jste zadali do filtru.

3. Ve Wireshark si otevřete soubor s příponou `.pcap` z adresáře `/root/isa1`. Zobrazte si graf síťových toků (flow graph) pouze pro SIP. Zapište datum a čas, kdy byl vyslán první SIP paket.

4. Zachyťte pouze HTTP a DNS provoz. Ve webovém prohlížeči zkuste otevřít několik stránek na různých URL adresách. Analyzujte obsah a posloupnost DNS paketů a následných HTTP paketů. Zkuste opakovaně načítat stejnou stránku. Proč v některých případech není zachycena DNS komunikace?

5. Zachyťte HTTP komunikaci (nešifrovanou), zobrazte si TCP stream této komunikace. Zaznamenejte URL, které jste použili pro vygenerování HTTP provozu, a co Vám TCP stream zobrazil.

1.3 Ukončení práce v laboratoři

Jakmile máte veškerou práci hotovou, spusťte jako `root` skript s názvem `clean`, který se nachází v adresáři `/root/isa1`.

Teorie

1 Zjišťování konfigurace zařízení

Detailní popis včetně možných voleb a příkladu použití k jednotlivým níže zmíněným příkazům můžete nalézt v manuálových stránkách (`man <příkaz>`).

1.1 Základní orientace v Linuxu

Většina práce v OS Linux bude probíhat v terminálu. Terminál na školních PC spustíte pomocí `Alt+F2`, zde zadáte "`gnome-terminal`". Případně můžete vybrat aplikaci terminálu z postraní nabídky.

1.1.1 Hierarchie souborového systému

Všechny soubory jsou uloženy v souborovém systému. Ten je organizován jako invertovaný strom adresářů, kde kořenovým adresářem je `root` adresář (označení `"/"`). Ten obsahuje podadresáře, kde každý má svůj standardizovaný účel pro zařazení různých souborů.

Některé podadresáře a jejich význam:

- **/etc** - obsahuje konfigurační soubory systému,
- **/var** - proměnlivá boot perzistentní data, dynamicky se mění, obsahuje databáze, cache adresáře, obsah www stránek (`/var/html/www`), logy (`/var/log`),
- **/dev** - obsahuje speciální soubory pro přístup k hardware zařízením.

1.1.2 Základní příkazy

Některé základní příkazy pro práci v terminálu OS Linux:

- `cd` - posun v adresářové hierarchii,
- `ls` - zobrazení obsahu adresáře,
- `cp` - kopírování souborů,
- `mv` - přesun souborů,
- `mkdir` - vytvoření adresáře,
- `touch` - vytvoření prázdného souboru,
- `head` - zobrazení x řádků od začátku souboru,
- `tail` - zobrazení x řádků z konce souboru,
- `cat` - zobrazení obsahu souboru,
- `grep` - filtrování/hledání v textu,
- `sed` - editace textu v příkazové řádce,
- `awk` - skenování a zpracování textu,
- `ps` - zobrazení informací o běžících procesech.

Mezi command line editory patří například `nano` nebo `vim`. Kromě nich můžete taktéž použít grafické editory, například `gedit`.

1.1.3 Uživatelé

Každý uživatel v OS Linux má své jedinečné `uid`. Každý proces (program) v systému je spuštěn pod nějakým uživatelem. Každý soubor je vlastněn uživatelem a omezen přístupovými právy. Tato přístupová práva se vztahují taktéž na procesy spuštěné daným uživatelem.

- `root` - super uživatel, má veškerá práva a kontrolu nad systémem.
- `user` - pouze základní správa, bez možnosti zasahovat do systémového nastavení.
- `sudo` - delegace některých práv super uživatele na normálního uživatele (nastavení v `/etc/sudoers`).

Při správě OS se pokud možno snažíme vyhnout používání systému jako `root`. K tomuto slouží `sudo`, kde má daný uživatel přesně specifikováno, které operace smí se systémem provádět, a je pak lépe dohledatelné, kdo co nastavil.

Přepínání uživatele: `su [user]`

- `su` - přepnutí na uživatele `root`.
- `su user` - přepnutí na uživatele `user`.

Spouštění příkazů jako `sudo`: `sudo command...`

1.2 Síťová konfigurace zařízení

Ke zjišťování síťové konfigurace na daném zařízení můžeme použít několik nástrojů, které jsou na OS Linux k dispozici.

1.2.1 Zobrazení konfigurace

Jedním ze základních příkazů je `ip`. Pomocí tohoto příkazu můžeme zjistit konfiguraci všech síťových zařízení (fyzických i virtuálních), obsah routovací tabulky aj. Manuálové stránky pro dané možnosti zobrazíte jako `ip-<volba>`.

- `ip address` - zobrazí adresu na všech síťových zařízení.
- `ip route` - zobrazí všechna pravidla v routovací tabulce.
- `ip link` - vylistuje všechna síťová zařízení.
- `ip neighbor` - zobrazí obsah ARP záznamů.

Tento nástroj mimo jiné slouží také ke konfiguraci, ne jen k jejímu zobrazení. [1]

Alternativou k zobrazení obsahu routovací a ARP tabulky jsou tyto příkazy:

- `netstat` - zobrazí mimo jiné obsah routovací tabulky (volba `-rn`)

Příkaz `arp` zobrazí obsah ARP tabulky a také umožňuje provádět její správu (například odstranit záznamy).

K zobrazení aktuálně běžících spojení včetně protokolu, portu, zdrojové a cílové adresy slouží příkaz `ss` (`sockstat`). Příkaz `lsof -ni` zobrazuje otevřená spojení.

[1] Obdobou tohoto nástroje je `ifconfig`.

1.2.2 Konfigurační soubory a logy

Na OS Linux je veškeré nastavení systému a služeb uloženo v adresáři `/etc`. Jednotlivé služby zde mají svůj konfigurační soubor s příslušným názvem. Pro lepší přehlednost je možné vytvářet více konfiguračních souborů, pro tyto účely zde pak existují adresáře `"nazevsluzby.d/"` (například `/etc/rsyslog.d/`). Veškeré nastavení v souborech v těchto adresářích je pak aplikováno na danou službu.

Některé ze základních podstatných konfiguračních souborů:

- `/etc/hostname` - obsahuje hostname systému.
- `/etc/hosts` - obsahuje mapování hostname na IP adresu (zde můžete libovolné adrese přiřadit hostname, následné použití tohoto hostname předchází samotnému vyhledání pomocí DNS resolveru).
- `/etc/host.conf` - obsahuje konfiguraci specifickou pro resolver.
- `/etc/resolv.conf` - obsahuje direktivy specifikující výchozí servery, na které je poslán dotaz pro překlad doménového jména na IP adresu.
- `/etc/rsyslog.conf` - obsahuje nastavení syslogu, rozřazování jednotlivých zpráv do příslušných logovacích souborů.

Veškeré podstatné události jsou zaznamenávány do logů. Všechny logovací soubory jsou uloženy v adresáři `/var/log`. Tyto soubory (logy) jsou cyklicky promazávány po určitém čase či po dosažení určité velikosti. Dobu, po kterou jsou logy udržovány, lze nastavit v konfiguračním souboru `/etc/logrotate.conf`.

Některé významné logovací soubory:

- `/var/log/messages` - obsahuje obecné zprávy systému (bez kritických a debugovacích).
- `/var/log/auth.log` - zde jsou uloženy zprávy týkající se autentizace uživatelů.
- `/var/log/kern.log` - ukládá zprávy týkající se jádra OS.

Tyto logovací soubory můžeme prohlížet přímo pomocí standardních zobrazovacích příkazů.

Systémové události můžeme také prohlížet pomocí nástroje `journalctl`, ten nám umožní procházet podrobnější informace daných zpráv, filtrovat a vyhledávat.

Příklad použití `journalctl`:

- `journalctl -n <x>` - zobrazí posledních x záznamů.
- `journalctl -p <priority>` - zobrazí pouze záznamy s danou syslog prioritou.
- `journalctl -u <unit|patern>` - umožní vyfiltrovat pouze záznamy s daným paternem/službou.
- `journalctl -f` - zobrazí kontinuálně posledních 10 událostí.
- `journalctl --since <whenstart> --until <whenend>` - zobrazí záznamy v daném rozmezí.

1.2.3 Aktivní zjišťování

Zjišťování dostupnosti zařízení a konektivity. K tomuto účelu poslouží příkazy:

- `ping ipv4|domain` - zašle ICMP ECHO_REQUEST k danému hostu (pro ipv4).
- `ping6 ipv6|domain` - zašle ICMP ECHO_REQUEST k danému hostu (pro ipv6).

- `traceroute ipv4|domain` - zobrazí cestu, kudy packet prochází skrz síť k danému hostu (pro ipv4).
- `traceroute6 ipv6|domain` - zobrazí cestu, kudy packet prochází skrz síť k danému hostu (pro ipv6).
- `tcptraceroute ipv4/ipv6|domain` - `traceroute` používající TCP.

Mimo základní výše uvedené příkazy `ping` a `traceroute` existuje užitečný nástroj `nmap`.

- `nmap` - umožňuje pomocí různých protokolů aktivně zjišťovat otevřené porty, aktivní hosty na dané síti apod.

Pro přihlášení na vzdálený host můžeme použít dva nástroje:

- `telnet ip` - nešifrované přípojení
- `ssh ip` - šifrované přípojení

2 Analýza Wireshark

Wireshark je aplikace sloužící k analýze protokolů a zachytávání paketů. Zachytává síťový provoz z jednotlivých síťových zařízení (jak fyzických, tak virtuálních) a umožňuje inspekci jednotlivých polí v daných paketech.

Alternativou ke grafickému Wireshark je command line nástroj `tcpdump`. Wireshark má oproti `tcpdump` například integrované volby pro řazení, filtrování a jiné.

2.0.1 Zachytávání provozu

Zachytávat provoz můžeme z libovolného síťového zařízení, dokonce z více zařízení najednou.

Odchytíme pomocí `Capture -> Interfaces...` a zde vybereme zařízení, které chceme snímat. Spustíme pomocí `Start`. Takto odchytíme veškerý provoz.

Existuje zde volba filtrace provozu již při zachytávání. Tato vlastnost může být užitečná při dlouhodobějším zachytávání, především z hlediska redukce množství ukládaných paketů.

Paketové filtry můžeme aplikovat před spuštěním zachytávání. Použijeme `Options` a v poli `Capture Filter:` můžeme zvolit jeden z předdefinovaných filtrů, případně specifikovat vlastní.

Odchycený provoz můžeme uložit do souboru (přípona `.pcap`). Následně je možné tyto soubory znovu analyzovat.

2.0.2 Filtrovací operátory

V některých situacích, například pokud zachytáváme veškerý provoz, je velmi vhodné použít filtrování. Jelikož zde pracujeme s velkým množstvím různých paketů, můžeme si tak vyfiltrovat pouze ty relevantní. K tomu slouží tzv. display filtry, jejichž syntax je popsána v Tabulce 1.

2.1 Flow graph

Wireshark umožňuje zobrazit vybranou komunikaci v `flow graph`. Zde můžeme přehledně vidět, které stroje, rozlišené IP adresou, spolu komunikují a kdo komu, zasílá kterou zprávu v jakém pořadí. Flow graph si zobrazíme pomocí `Statistics -> Flow Graph..`, zde si vybereme, co chceme zobrazit.

Porovnávání	`==, >=, <=, !=, contains`
Logické operátory	`||, or, &&, and, !, not`
Kombinace filtrů	`(ip.src==192.168.0.105 and udp.port==53)`
Filtrování na základě existence pole	`http.cookie or http.set_cookie`
Filtrování specifických bytů	`eth.src[4:2]==22:1b`
Regex filtrování	`http.host && !http.host matches "com$"`

Tabulka 1: Filtrovací operátory

2.2 Zobrazení streamu

Zachycenou komunikaci vidíme v podobě jednotlivých paketů. Pokud například zachytíme HTTP komunikaci, jeden paket nese většinou pouze část obsahu HTTP zprávy. V tomto případě můžeme pro zobrazení celé zprávy (celého streamu) použít volbu `Flow <protokol> Stream`. Zobrazení provedeme tak, že vybereme jeden zachycený paket, označíme ho a následně `Analyze -> Follow <protokol> Stream`.

2.3 Čas

V rámci zachytávání paketů nese každý záznam informaci o čase. Defaultně je zde čas zobrazen v sekundách, od počátku zachytávání komunikace. Nicméně v `View -> Time Display Format` můžeme vybrat konkrétní zobrazení času a zjistit tak například, ve který den byl daný paket zachycen.