Detekce NATů na lokální síti – semestrální práce předmětu SPS

Jakub Fišer, FEL ČVUT, 2008

ZADÁNÍ

Vytvořit systém, který bude na kolejní síti klubu Pod-O-Lee detekovat „neposlušné“ uživatele, kteří sdílejí svou konektivitu s jinými lidmi, nebo mají připojené nepovolené síťové zařízení (další počítač, router, …)

Tento systém by měl spolupracovat s DUSPSem (Databáze Uživatelů Studentské Počítačové Sítě) a měl by být schopen počítat detekce jednotlivých uživatelů (uživatel je nejprve varován, poté teprve odpojen od sítě a to na (n-2)*7 dní, kde n je počet dní, kdy byl detekován NAT), o jejich prohřešcích je informovat a informovat o nich i Správce Bezpečnosti a příslušné skupiny televizorů.

MOTIVACE

Uživatel používající NAT pro připojení dalších lidí, nebo zařízení, nejenže připravuje klub o finance, což je pro neziskovou organizaci poměrně zásadní, ale zároveň vystavuje svým jednáním síť riziku zneužití (především např. použitím nezabezpečených Wi-Fi routerů). Zároveň tím porušuje interní předpisy Klubu.

MOŽNOSTI DETEKCE

Běžný NAT funguje pouze jako překladač adres – v podstatě pouze přepíše zdrojovou adresu v paketu, ale ostatních údajů si nevšímá. Je proto poměrně snadné takový NAT odhalit, umístíme-li vhodný detektor na vhodné místo.

DETEKČNÍ PRINCIPY

Podle zkušeností lze použití NATu jednoznačně odhalit pomocí

- rozdílu v TTL
- rozdílu v OS fingerprint

Existují další údaje, které lze vyčíst z hlaviček paketů a které mohou razantně zvyšovat pravděpodobnost detekce:

- rozdíl v časových razítkách
- vysoké odchozí porty
- změny velikosti okna

DETEKČNÍ SOFTWARE

Po prozkoumání situace na poli OpenSource software jsem vybral dva vhodné kandidáty – **NatDet** a **MasqDet**. Na doporučení bývalého správce IP konektivity (člověka, kterého považuji za velmi zkušeného) jsem se rozhodl preferovat NatDet, který se nakonec po krátkém testování osvědčil a stal se

tak srdcem celého systému.

UMÍSTĚNÍ DETEKTORU

Logické umístění je místo, kudy musí projít každý paket opouštějící hranice klubové sítě – centrální prvek. V našem případě je centrální prvek zapůjčen od VIC ČVUT a jedná se o Cisco router, u kterého přímé nasazení vyloučené.

Nicméně naštěstí pro mě existuje na klubové síti systém počítání trafiku, který funguje velice jednoduše – trafik počítá jeden z méně vytížených serverů a to tak, že veškerý provoz centrálního prvku (In i Out) je mirrorován na dedikované síťové rozhraní tohoto serveru. Umístění detektoru na tento server je tedy v této síti více než logické.

PŘÍPRAVA DETEKČNÍHO SOFTWARE

Program NatDet pouze generuje jakýsi textový výstup, který je třeba dále zpracovat. Navíc, tento výstup je občas ne zcela deterministický, proto bylo třeba trochu poupravit zdrojový kód (je to jednodušší, než upravovat parser a z logů se pak lépe počítají statistiky).

NatDet je psaný v jazyce C a jeho kód je poměrně jednoduchý, takže pro zkušeného správce by neměl být problém si kód případně dále upravit. Samotné odchytávání paketů je řešeno pomocí libpcap, kód NatDetu je tedy zaměřen pouze na analýzu hlaviček.

Chování NatDetu je třeba přizpůsobit rozsahu sítě, na které bude pracovat: v souboru *config.h* je nutné nastavit především velikosti cache pro detekované IP adresy a důležitost jednotlivých kritérií. V mém případě bylo pouze třeba zvětšit cache tak, aby pokryla celý IP rozsah, ostatní nastavení zůstala výchozí.

Vzhledem k tomu, že program NatDet je postaven nad libpcap, umí filtrovat pakety pomocí jednoduchých pravidel, psaných stejnou syntaxí, jako program tcpdump. **Silně** proto **doporučuji** omezit detekci pouze na odchozí trafik – výrazně se tím sníží zátěž serveru a zvýší relevantnost logů.

„BUSINESS LOGIKA“

Logy programu NatDet je třeba dále zpracovat. Na to slouží sada bash skriptů, které jsem vytvořil speciálně za tímto účelem. Tyto skripty jsem se ale z bezpečnostních důvodů rozhodl nezveřejňovat, nicméně pokud by byl zájem, rád je soukromě ukážu i s příslušným komentářem. Bash jsem zvolil jednak pro to, že případné úpravy systému se dají dělat prakticky za běhu a druhak pro to, že Bash ovládám ze skriptovacích „jazyků“ nejlépe.

Principiálně se v bash skriptech ošetřuje několik věcí:

1. třídění logů podle data
2. relevantnost detekcí (ošetření false-positive, viz níže)
3. job control na úrovni skriptů – signální řízení (viz níže)
4. počítání detekcí
5. informování správců a uživatelů

Skripty by mohly ošetřovat i samotné „odpojování“ zlobivých uživatelů, ale

tato funkcionalita zatím implementována není, neboť pro ní neexistuje vhodný backend v systému DUSPS. V současné době spolupracuji se Správce Autority na vytvoření vhodného modulu.

FALSE-POSITIVE DETEKCE

Při testovacím provozu systému se objevil problém – program NatDet detekoval „NATy“ u některých počítačů s OS Windows. Program sice vypsal kritéria, která ale logům neodpovídala. Tento problém jsem vyřešil zvýšením kontroly logů na úrovni kontrolních skriptů. Již jsem požádal o spoluprácí uživatele (řádově 5 lidí z 800), kteří byli tímto problémem postiženi, ale bohužel jsem se nesetkal s pochopením. Mě samotnému se situaci zreplikovat nepodařilo, proto se zatím musím spokojit s řešením důsledku, i když bych mnohem raději nalezl příčinu. Doufám, že se objeví nějaký rozumný uživatel, který bude chtít aktivně spolupracovat a problém se podaří vyřešit.

SIGNÁLNÍ ŘÍZENÍ

Nejen kvůli init scriptům bylo zapotřebí naučit skripty přijímat a zpracovávat řídící signály, které používám k rotaci logů bez restartování NatDetu a aniž by došlo k vypnutí skriptů. Vypnutím NatDetu bych totiž přišel o cache a dočasně snížil pravděpodobnost detekce. Dalším využitím signálů je možnost znovunačtení konfigurace bez restartu. To ale většinou není zapotřebí, neboť většina zásadních změn v konfiguraci se bez restartu neobejde (např. filtr pro pakety, apod.)

SUDO A OŠETŘENÍ BEZPEČNOSTI

Program NatDet potřebuje ke své činnosti práva superuživatele, detekční skript pak potřebuje tato práva při zjišťování PID běžíciho NatDetu. Normálně jsou všichni uživatelé schopni vidět všechny procesy, ale na našich serverech je nainstalován grsec, který to znemožňuje.

Jsou dvě možnosti, jak zajistit rootovská práva pro NatDet a jediná pro zjištění PID. Pro NatDet lze zvolit buď SUID bit, nebo sudo.

Výhoda SUID je, že není třeba upravovat skripty. Nevýhoda je, že soubor musí být vlastněn rootem a tedy je nemožné případně NatDet upgradovat bez pomoci roota.

Výhoda SUDO je, že je možné provést upgrade i bez roota, ale zase se tím zvyšuje bezpečnostní riziko při napadení účtu detektoru. Po dohodě se správcem serveru jsme dospěli k variantě SUDO, protože u správce detektoru se předpokládá důvěryhodnost a k účtu je možné přistupovat pouze pomocí ssh klíče (což trochu zvyšuje jeho zabezpečení).

Detekční skript pak lze použít pouze pomocí SUDO, nicméně jediný program, který je pomocí SUDO volán, je **pgrep**. Tedy v souboru sudoers je nastaveno sudo pouze pro **pgrep** s příslušnými parametry. Bezpečnostní riziko je v tomto případě výrazně nižší.

BEZPEČNOSTNÍ PROBLÉM S VARIANTOU SUDO U NATDETU

Jak již bylo zmíněno výše, pro detekci používám upravenou verzi NatDetu,

jehož binárka je pro jednoduchost umístěna v domovském adresáři detektoru. Původní idea byla možnost správce detektoru upgradovat na novou verzi NatDetu bez práv roota. Ale vzhledem k tomu, že nedávno došlo k „politickému převratu“ a do pozice Správce Bezpečnosti se dostal neznámý a tedy nedůvěryhodný člověk, bylo třeba variantu SUDO přehodnotit.

Po krátké úvaze nad SUID bitem jsem se rozhodl pro řešení na úrovni práv souborového systému, neboť SUID bit by znamenal přepsání skriptů a nutnost odstavení systému na dobu potřebnou k odladění změny. Stačilo změnit vlastníka binárky a vlastníka adresáře s binárkou na roota, adresáři nastavit sticky bit a zakázat zápis do binárky. O zbytek se postará kernel. V případě, že nový správce získá důvěru správce serveru, není pak problém uvést nastavení do původního stavu.

UCHOVÁVÁNÍ INFORMACÍ O DETEKCÍCH, VÝJÍMKY

Pro každý detekovaný počítač existuje záznam v podobě adresáře a informačních souborů v něm. Název koresponduje s fqdn detekovaného počítače. Program NatDet umí při detekci překládat IP adresy na hostname.

FQDN vs. IP ADRESA

Při psaní systému jsem byl postaven před problém, zda detekci založit na IP adresách, nebo na jménech počítačů. Systém přidělování IP adres na koleji funguje tak, že IP adresa je napevno přidělena konkrétní zásuvce, nikoliv osobě (počítači). Detekce podle IP adres by tímpádem vyžadovala neustálou kontrolu změn v IP <=> Hostname, čímž by došlo k drobnému zvýšení zátěže serveru.

Proto jsem se pro identifikaci počítače rozhodl používat FQDN. Tento systém je jištěn lokální DNS cachí, pro případ výpadku NS serveru. Lokální DNS cache je pomocí jednoduchého php skriptu (dodaného tvůrcem DUSPSu) každý den aktualizována přímo z DUSPSu společně s e-mailovými adresami uživatelů.

SYSTÉM VÝJIMEK

Podle interních předpisů klubu může představenstvo udělit výjimku z pravidla o zákazu NATů. Takovou výjimku je samozřejmě třeba do systému zavést. Záznam pro počítač se ve spoolu záznamů objeví až po nějaké detekci. Proto jsem „vycucnul“ funkci pro vytvoření záznamu do extra skriptu, který lze spustit ručně a tím vytvořit torzo záznamu pro zatím nedetekovaný počítač. V tomto torzu je pak třeba patřičně upravit soubor allow a to přidáním řádku
`ALLOW=BYPASS`
čímž dojde k udělení výjimky – při detekci bude stroj ignorovám, ale zalogován.

STRUKTURA ADRESÁŘE

V domovském adresáři detektoru jsou adresáře unixově zdatnému člověku povědomé: bin, etc, log, share, spool. Toto rozvržení je úmyslné pro případ, že by systém byl nějak výrazně rozpracován, zveřejněn a distribuován.

```
|-- bin
|   |-- natdet          binárka NatDetu
|   |-- nufs.sh         hlavní spouštěcí skript (spouští démona)
|   |-- nufs-cron.sh    skript pro cron (updaty dns, mailů...)
|   |-- nufsd.sh        démon, který zpracovává výstup NatDetu
|   `-- rc.nufs
|-- etc
|   |-- nufs.conf.sh    skript dynamické konfigurace (datum, apod.)
|   `-- nufs.conf       statická konfigurace (cesty,...)
|-- log
|   |-- 01-08           adresář pro každý měsíc
|   |   `-- 01-01-08.log soubor pro každý den
|   |-- current.log     aktuální den (po půlnoci bude přesunut)
|   `-- nufs.log        hlavní log detektoru
|-- share
|   |-- mail.1          text mailu s prvním varováním
|   |-- mail.2          test mailu s odpojením
|   |-- signatures      signatury OS fingerprintů, pro Natdet
|   `-- nufs.functions  funkce pro démona nufsd.sh
|-- spool
|   |-- pepa.pod.cvut.cz hostname
|   |   |-- allow       nastavení výjimek
|   |   |-- ip          ip adresa počítače
|   |   |-- mail        e-mail majitele
|   |   |-- lastmail    text posledního zaslaného mailu
|   |   |-- points      počet detekcí NATu
|   |   `-- mailed      datumy jednotlivých upozornění
|   `-- allow.proto     prototyp nastavení výjimek
`-- src
    `-- natdet          adresář s upravenými zdrojáky NatDetu
```

SOUČASNÝ STAV

Projekt detektoru funguje úspěšně již několik měsíců, na kontě má desítky úspěšných detekcí a po odladění false-positive problémů se neobjevila žádná chybná detekce. Pozorováním provozu systému mě napadlo několik možných rozšíření funkcionality, např. generování statistiky, přiřazování logů konkrétním počítačům, rozšíření systému výjimek na různá kritéria, apod., ale ty jsou pro běh systému nepodstatné a proto se jim budu věnovat až ve svém volném čase.

Skripty jsou, dle mého skromného názoru, psané přehledně a jsou rozumě členěné. Člověku znalému unixových poměrů by nemělo dělat problémy se v nich vyznat, nicméně bude potřebovat širší znalosti Bashe.

Automatické odpojování zlobivých uživatelů je závislé na modulu v systému DUSPS, který není pod mojí správou, proto bude detektor o tuto možnost rozšířen až po té, co se se správcem DUSPSu dohodneme na konkrétních implementačních záležitostech.

ZÁVĚR

Systém je v podstatě hotový a případný pokračovatel, pokud nebude chtít systém nějak vylepšovat v podstatě nepotřebuje vědět, jak funguje. Pokud by však někdo v systému pokračovat chtěl, měl by mu tento text a zběžné pročtení skriptů stačit k získání dobrého přehledu (znalost bashe a unixových nástrojů je v tomto případě nutnou podmínkou).

Detektor se zatím dobře osvědčil, skripty jsou napsány poměrně robustně a i přes několik málo výpadků serveru nikdo nezaznamenal žádné problémy (např. s nenastartováním služby, apod.).

Program NatDet se ukázal být dobrou volbou, především díky rychlosti a malým paměťovým nárokům. Program MasqDet nebyl nijak výrazně testován – nebylo to zapotřebí.