

В отчете рассматривается алгоритм аутентификации. Предлагается использовать различные признаки агентов, историю их работы для оценки доверия действию агента. Проводится анализ проблем, признаки и способы их решения. Предлагается метод оценки доверия для улучшения работы алгоритм аутентификации.

In the paper algorithm of authentication is determined. The analysis of problems, sign and methods of their decision are conducted. The method of estimation of trust is offered for the improvement of work algorithm of authentication

УДК 681.3

О. Ф. ЛАНОВИЙ,

канд. техн. наук, доц. ХНУВС

І. В. КОБЗЕВ, *канд. техн. наук, доц. ХНУВС*

О. С. УДОВЕНКО, *аспірант ХНУРЕ*

СИСТЕМА ПІДРАХУНКУ ТРАФІКУ МЕРЕЖІ З ВИКОРИСТАННЯМ ЗАСОБІВ ОБ'ЄКТНО-ОРІЄНТОВАНОГО ПРОГРАМУВАННЯ

Проблема обміну інформацією між комп'ютерами існує з часу виникнення обчислювальної техніки. Вона полягає в ефективній організації спільної роботи окремих комп'ютерів, що дозволяє вирішувати одну задачу за допомогою декількох комп'ютерів, спеціалізувати кожен із комп'ютерів на виконання певної функції, спільно використовувати ресурси тощо. Існує безліч засобів обміну інформацією: від простого перенесення файлів за допомогою носіїв інформації до використання глобальної комп'ютерної мережі Інтернет, яка здатна пов'язати всі існуючі комп'ютери.

Найчастіше під терміном «локальні мережі» (LAN (Local Area Network)) розуміють такі мережі, які мають невеликі, локальні розміри і з'єднують близько розташовані комп'ютери. Проте досить поглянути на характеристики деяких локальних мереж, щоб зрозуміти, що таке визначення не дуже точне. Здебільшого локальна мережа пов'язує від двох до декількох десятків комп'ютерів. Але граничні можливості деяких локальних мереж набагато вищі: максимальна кількість абонентів може досягати тисяч. Називати таку мережу малою, напевно, неправильно.

У свою чергу, більшість локальних мереж підключена до глобальної комп'ютерної мережі Інтернет. Підключення здійснюють провайдери, що надають послуги як з обслуговування локальної мережі, так

і з доступу до мережі Інтернет. Провайдери повинні забезпечувати: якісний підрахунок використовуваного користувачами мережі трафіку, тарифікацію користувачів, своєчасне відключення неплатників, спрощення способів оплати і ін.

Для вирішення цих завдань були запропоновані системи обліку трафіку і системи білінгу. Звичайно ці дві системи використовуються разом і реалізуються як програмний комплекс, призначений для обліку й управління трафіком у локальних мережах.

Найбільш прості системи обліку трафіку виконують такі операції:

- збирання даних;
- підсумовування даних;
- збереження суми в базі даних або лог-файлі;
- надання даних відповідно до запитів;
- сигналізація зовнішній програмі про перевищення трафіку.

Подальшим розвитком систем обліку трафіку є створення білінгових систем, до визначальних особливостей яких можна віднести:

- розвинений інтерфейс управління, налаштування, запитів для клієнтської частини;
- перерахунок значень лічильників у грошове вираження відповідно до тарифного плану;

- ведення рахунків клієнта (платежі, зміна планів, блокування тощо);
- підтримка бухгалтерії (акти, рахунки, звіти тощо);
- можливість відключення клієнта (з повідомленням).

Дана стаття присвячена проблемі розробки і дослідження аспектів створення білінгової системи розрахунку трафіку обчислювальної мережі з використанням сучасних засобів об'єктно-орієнтованого програмування.

Простими системами обліку трафіку називають системи, що дозволяють визначити обсяг вхідного і вихідного трафіку для певної IP-адреси за певний термін, але не дозволяють деталізувати трафік. Такий підрахунок має певні переваги, оскільки в цьому випадку система вимагає менше системних ресурсів і менший обсяг сховища даних [1].

Розглянемо білінгові системи підрахунку трафіку, які передбачають розширення і деталізацію функцій простих систем.

У частині таких систем використовується спосіб підрахунку трафіку Ip accounting. Це засіб реалізовано в Cisco і деяких інших роутерах [2]. Суть його полягає в тому, що для кожного потоку пакетів з унікальною парою IP-адрес створюється окремий лічильник. У багатьох реалізаціях Ip accounting як критерій відмінності також використовуються TCP/UDP-порти і IP-протокол. Зібрана статистика зберігається в пам'яті роутера і періодично знімається й обнуляється за запитом ззовні. Такий метод дуже простий, але має свої недоліки: дуже навантаженому роутеру потрібно багато пам'яті оскільки процес зняття / обнулення статистики інтенсивно завантажує роутер, а перенавантаження роутера призводить до втрати даних, які були зібрані з моменту останнього зняття.

Спосіб підрахунку трафіку ng_ipacct дозволяє проводити перехоплення і підрахунок трафіку в ядрі. Зняття / обнулення трафіку здійснюється за допомогою утиліти ipacctctl, яка здійснює імпорт зібраної статистики в режимі userland.

Переваги методу ng_ipacct:

- вимагає менше ресурсів, ніж ip accounting, оскільки реалізовується повністю в ядрі;

– утиліта netgraph дозволяє використовувати ng_ipacct у різних комп'ютерних конфігураціях.

Спосіб підрахунку трафіку Ipcad є userland-додатком, що перехоплює трафік за допомогою різних програмних механізмів.

Переваги методу Ipcad:

- використовує відомі інтерфейси і працює практично на будь-якій unix ОС;

– дозволяє отримувати дані про трафік за допомогою різних механізмів: pcap, bpf, divert, tee, ulog і libipq під ОС Linux;

– на відміну від методу ip accounting, при перезавантаженні інформація не втрачається;

– процес виведення результатів відповідає аналогічному для Cisco, що дозволяє використовувати розроблені під Cisco скрипти і білінги;

– дозволяє агрегувати трафік.

Недолік методу Ipcad:

– працює в режимі userland, що призводить до надмірного завантаження центрального процесора.

Спосіб підрахунку трафіку Netflow полягає в тому, що роутер ідентифікує унікальні потоки трафіку аналогічно з ip accounting, проте після закінчення потоку зібрана статистика відразу відправляється за UDP-датаграмою до колектора [3]. Таким чином, у пам'яті роутера зберігаються лише ті потоки, які активні протягом заданого часу. Крім того, Netflow дає більше інформації, ніж ip accounting, а саме: вхідний і вихідний інтерфейси, маршрутні маски, час початку і кінця життя потоку тощо.

До найбільш перспективних білінгових систем слід віднести систему StarGazer, яка побудована на клієнт-серверній технології, що забезпечує необхідну гнучкість і швидкодію. Як сервер використовується комп'ютер з ОС Linux або FREEBSD; клієнтами можуть бути комп'ютери з ОС сім'ї Windows або ж клієнти з ОС Linux і FREEBSD. Також клієнтом може бути будь-яка ОС, яка підтримує протоколи TCP-IP і web-браузер.

Система StarGazer підтримує додаткові

модулі, що дозволяє самостійно нарощувати функції системи [4].

Основні можливості системи:

- контроль над клієнтами мережі, управління клієнтами, поточне корегування;
- авторизація клієнта з подальшим дозволом або заборороною доступу до Internet;
- підрахунок трафіку за заздалегідь заданими напрямками і правилами;
- підрахунок витрачених клієнтом коштів і автоматичне відключення в разі їх повного використання;
- зберігання додаткової інформації про клієнта (домашня адреса, телефон тощо);
- автоматичний моніторинг підключення клієнтів до мережі і виведення результатів на екран;
- ведення історії для всіх клієнтів;
- оперативне надання клієнтові інформації про його трафік і наявність грошей на рахунку;
- формування звітів про баланс клієнтів;
- можливість підключення додаткових модулів.

Проведений вище аналіз дозволяє визначити напрямки у сфері розвитку білінгових систем, що вимагають додаткового дослідження. Відповідно найбільш зручною і гнучкою системою обліку трафіку є StarGazer, що має модульну структуру. Проте для управління цією системою потрібне написання правил конфігурації. На нашу думку, доцільно розробити web-інтерфейс, що полегшує завдання практичного використання можливостей StarGazer для білінгу трафіка мережі.

Програмна реалізація білінгової системи є набором скриптів для реалізації функцій обліку трафіку і деяких допоміжних функцій. Для створення такої системи можна використовувати будь-яку мову, яка дозволяє працювати з базами даних (далі – БД) і використовувати клієнт-серверну технологію. Аналіз найбільш поширених мов із такими можливостями, таких як Perl, PHP, Ruby, ASP, показав, що для вирішення завдань справжньої роботи найраціональніше використовувати конструктор, тобто набір готових програмних компонентів, модулів і класів. Це

допомагає скоротити час розробки, а також дозволяє легко розширити функціональність системи.

Розглянемо три конструктори, які використовують комбінації мов web-програмування: Ruby – Ruby on Rails, PHP – Symfony, Python – Django. Для тестування ефективності і доцільності використання цих конструкторів у білінгових системах було написано тестовий додаток для роботи з БД. При цьому мета тестування полягала в оцінюванні оперативності здійснення конструкторами операцій вибору даних (без обробки) з БД і виводу їх на дисплей. Всі операції для простоти були поміщені на одну сторінку.

Кожна з тестованих сторінок виконувалася за певний час і була сконфігурована так, щоб відбувалася обробка однакової кількості байтів інформації.

Результати проведеного тестування можуть бути використані при виборі конструктора для розробки білінгової системи.

За результатами тестування як базова була обрана система Ruby on Rails [5, 6]:

- конструктор Ruby on Rails написано на Ruby. Значить, на кожному етапі розробки використовується одна й та ж мова, що істотно спрощує процес створення додатків;
- у середовищі Rails застосовується принцип повторного використання коду (Don't Repeat Yourself), тобто програма або програмний модуль частково або повністю повинні складатися з написаних раніше компонентів і / або частин інших систем;
- середовище Ruby on Rails ґрунтується на принципі угоди про конфігурацію. Rails не використовує конфігураційні файли, а отримує всю необхідну інформацію з коду додатків і баз даних. Це дозволяє обійтися без компіляції і відразу бачити результат розробки.

До недоліків даного конструктора можна віднести складнощі з хостингом.

На сьогодні офіційний сайт Ruby on Rails наводить список серверів, що підтримують цю технологію. Ruby on Rails може працювати з будь-якими серверами, що підтримують технологію FASTCGI. Серед найпопулярніших варто назвати Apache і Lighttpd.

Як сервер бази даних Ruby on Rails підтримує MySQL, PostgreSQL, SQLite, IBM DB2, Oracle і Microsoft SQL Server, а також вбудовану СУБД SQLite.

Ruby on Rails дозволяє також швидко створювати гнучкі додатки, які можуть звертатися до БД у тому вигляді, в якому це необхідно в кожному конкретному випадку. Класична архітектура надає широкі можливості представлення інформації в системах, що використовують для доступу до даних web-інтерфейси.

Система StarGazer-M (далі – система) призначена для авторизації й обліку трафіку в локальних мережах і є модифікованою версією базової системи StarGazer, яка доповнена розробленими в Ruby on Rails web-додатками. При створенні даної системи було поставлено завдання створити продукт, який відповідав би вимогам більшості локальних мереж для обліку в них трафіку і коштів клієнтів, а також безпечної авторизації клієнтів.

Система побудована на клієнт-серверній технології, що забезпечує необхідну гнучкість і швидкодію. Як сервер використовується машина з ОС Linux або FreeBSD, як клієнти – машини сім'ї Windows, клієнти з ОС Linux або FreeBSD, а також будь-яка інша ОС, яка підтримує мережеві протоколи TCP-IP.

Система підтримує модулі, що дозволяють нарощувати її функції.

Основні можливості системи:

- контроль над клієнтами мережі: додавання, видалення, поточне корегування;
- авторизація клієнта з подальшим дозволом або заборонаю доступу до Інтернет;
- підрахунок трафіку за заздалегідь заданими напрямками і правилами;
- підрахунок витрачених клієнтом коштів і автоматичне відключення в разі їх повного витрачення;
- зберігання додаткової інформації про клієнта (домашня адреса, телефон тощо);
- автоматичний пінг усіх клієнтів мережі і виведення результатів на екран;
- ведення історії для всіх клієнтів;
- оперативне надання клієнтові інформації про його трафік і наявність коштів;
- формування докладних звітів про

стан клієнтів;

– модулі, що підключаються.

Компонентами системи є: сервер; конфігуратор; авторизатор.

Сервер використовується для авторизації клієнтів, маніпуляції правилами підрахунку трафіку, виконання фінансових операцій, зберігання статистичних даних і надання її клієнтам і адміністраторам.

Конфігуратор є основним інструментом для адміністрування системи, а також для перегляду і редагування властивостей клієнтів. Доступ до різних функцій обмежено для різних категорій адміністраторів. За допомогою конфігуратора виконуються всі фінансові операції з клієнтськими рахунками; проглядається докладна статистика клієнтів; налаштовуються деякі параметри сервера, наприклад, тарифні плани й адміністратори.

Авторизатор використовується для клієнтських машин: за допомогою його клієнт авторизується, отримує право на доступ до Інтернет або до інших ресурсів, які дозволені адміністратором. Також клієнт отримує свої статистичні дані і стан грошей на рахунку.

Розроблений web-додаток складається з двох контролерів: контролера адміністратора і контролера користувача. Контролер користувача дозволяє користувачеві мережі проглянути свої статистичні й особисті дані. Контролер адміністратора має розширені функції: управління користувачами, управління адміністраторами, управління тарифами на використання Інтернет, поповнення рахунку.

Аналіз основних систем обліку трафіку в комп'ютерних мережах показує, що найбільш перспективною білінговою системою, можливості якої значною мірою дозволяють реалізувати поставлені завдання, є система StarGazer, що побудована на клієнт-серверній технології і забезпечує необхідну гнучкість і швидкодію. До недоліків системи StarGazer слід віднести відсутність зручного web-інтерфейсу, що обмежує можливості адміністрування мережі. Для обґрунтованого вибору мови програмування, що дозволяє здійснити програмну реалізацію необхідних засто-

сувань, було проведено тестування конструкторів, що використовують комбінації мов web-програмування: Ruby – Ruby on Rails, PHP – Symfony, Python – Django. Тестування показало, що найбільш зручним для розробки білінгової системи, розрахованої на велику кількість користувачів, є використання конструктора Ruby on Rails.

У результаті розроблена і досліджена система StarGazer-M, призначена для авторизації й обліку трафіку в локальних мережах, що є модифікованою версією базової системи StarGazer, яка доповнена створеними з використанням Ruby on Rails web-додатками RoRbil. База даних додатка сформована з урахуванням усіх вимог системи StarGazer.

Література

1. Голомшток Л. Биллинговые системы для операторов традиционной и IP-телефонии / Л. Голомшток // Технологии и средства связи. – 2004. – Спецвыпуск.
2. Caputo R. Cisco Packetized Voice and Data Integration / R. Caputo. – McGrawHill Cisco Technical Expert, 2000.
3. Cisco Netflow. – Режим доступу: <http://www.cisco.com/warp/public/732/netflow>.
4. Help for StarGazer 2.4 rev 2.8. – Режим доступу: <http://StarGazer.dp.ua/about.html>.
5. Thomas D. Programming Ruby: The Pragmatic Programmer's Guide: second edition / David Thomas, Chad Fowler, Andrew Hunt. – Boston: Addison-Wesley, 2004.
6. Thomas D. Programming Ruby: the pragmatic programmer's guide / David Thomas, Andrew Hunt – Boston: Addison-Wesley, 2001.

Анотації

У статті проводиться огляд проблем розробки і дослідження аспектів створення білінгових системи розрахунку трафіку обчислювальних мереж із використанням сучасних програмних засобів. За результатами цього аналізу здійснені розробка і дослідження білінгової системи, яка побудована на використанні Ruby on Rails.

Проведен обзор проблем разработки и исследования аспектов создания биллинговых систем подсчета трафика вычислительных сетей с использованием современных программных средств. По результатам этого анализа осуществлена разработка и исследование биллинговой системы, построенной с использованием Ruby on Rails.

The article is devoted to the problem of development and researching aspects of creation billings systems of counting and traffic of computer local networks is conducted with the using of modern programming tools. On results this analysis development and research the billing system is carried out, based with the using of Ruby on Rails.

УДК 004.02

І. К. СЕЗОНОВА,
канд. техн. наук, проф. ХНУВС
Ю. Є. ХОРОШАЙЛО, *канд. техн. наук, доц. ХНУВС*
Т. П. КОЛІСНИК, *канд. пед. наук, доц. ХНУВС*

РОЛЬ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ У ГЛОБАЛІЗАЦІЇ ВИЩОЇ ОСВІТИ

Вектор сучасної політики і стратегії України в розвитку національної системи освіти спрямований на подальшу адаптацію до умов соціально-орієнтованої економіки, трансформації і інтеграції в європейське і світове співтовариство. Високо-

освічена молодь головний стратегічний резерв соціально-економічних реформ в Україні, без якого неможливий подальший розвиток суспільства. За прогнозами ЮНЕСКО, рівня національного благополуччя, що відповідає світовим стандартам,