ОСОБЛИВОСТІ БЕЗПЕКИ АВТОМАТИЗАЦІЇ ІНФОКОМУНІКАЦІЙНИХ МЕРЕЖ

Капуста Р.Д., Горяінова К.О.

Кафедра «Інфокомунікаційної інженерії ім. В.В. Поповського»,
Харківський національний університет радіоелектроніки, Україна

E-mail: roman.kapusta@nure.ua,
karyna.horiainova@nure.ua

Abstract

The paper is devoted to using programming languages to secure and automate infocommunication networks. The rapid development of network technologies entails a specific list of tasks related to the security of users' personal data and ensuring a reliable software component for the operation of the entire system. With the help of programming languages, dedicated software solutions are developed to accelerate the resolution of simple everyday tasks, complex automated information security systems, softwarized wired and wireless networks. The advantages and disadvantages of infocommunication network automation programming languages have been investigated.

Стрімкий розвиток мережних технологій спричиняє появі певного переліку завдань, які пов'язані із забезпеченням безпеки персональних даних користувачів, а також надійної програмної складової для ефективної роботи інфокомунікаційних систем. Ще донедавна адміністратору будо достатньо знань команд для налаштування та конфігурації мережних пристроїв, які розташовані на підприємстві. Однак технології змінюються, і тепер фахівець повинен мати вміння та навички із використання різних мов програмування автоматизації мереж. Це допоможе мережному інженеру створювати певні програмні рішення для прискореного розв'язання простих повсякденних задач, а також розробляти складні автоматизовані системи захисту інформації проводових і безпроводових мереж.

Існує чимало мов програмування, та їх кількість продовжує збільшуватись. Кожна з мов програмування містить у собі певний ряд переваг і недоліків, а також існують ті, що зовсім не підходять до вирішення завдань автоматизації мереж. Існують також вбудовані командні оболонки в операційних системах, які зможуть замінити мову програмування для створення простих алгоритмів автоматизації. Тому розглянемо низку рішень, які допоможуть мережному інженеру обрати найкращій варіант для вирішення різноманітних завдань.

Однією з популярних командних оболонок є `Bash` (`Bourne again shell`) [1-3]. Це скриптова мова програмування, яка використовується у Unix-подібних операційних системах. Оболонка `Bash` являє собою програмну оболонку з інтерфейсом командного рядка та певним вбудованим переліком команд, які дозволяють створювати скрипти, що будуть виконувати завдання, визначені адміністратором. Зазвичай `Bash` використовується для автоматизації повсякденних завдань без створення складних програмних рішень. Основною перевагою даної мови програмування є доступність і гнучкість, сценарії `Bash` можливо використовувати для будь-якої Unix-системи. Недоліком є досить складний синтаксис, також `Bash` важкий для розуміння та використаня звичайними користувачами. Проте він є досить ефективним під час вирішення простих завдань без встановлення допоміжних середовищ розробки.

Однак існують приклади, коли мова програмування може стати на заваді адміністратору мереж. Зазвичай, це мови, які мають спеціалізований принцип застосування або не встановлені на більшості комп'ютерів за замовчуванням. Одним з таких прикладів є `JavaScript` [2, 3].

`JavaScript` дозволяє створювати мережні застосунки, а також використовується веброзробниками. Хоча `NodeJS` і дозволяє розробнику запускати код на віддаленому сервері, це не дозволяє створювати швидкі та ефективні методи автоматизації мереж. `JavaScript` є однопотоковим, тобто лише один потік інформації може буде запущений, що неприпустимо для мережного

програмного засобу, завдання якого передати велику кількість інформації одночасно. Також, ця мова програмування досить складна у вивченні через досить заплутаний синтаксис. У більшості випадків мережні інженери уникають роботу з цією мовою програмування.

Проте існують мови, які були створені для вирішення завдань мережного програмування, прикладом такої мови є `Perl` [1-3]. Здатність мови програмування `Perl` до виконання різноманітних програм на серверах і динамічних вебсайтах є причиною її популярності та визнання серед адміністраторів мереж. Найбільшою перевагою є наявність так званих модулів. Модулем вважають попередньо написаний фрагмент коду, який можна використовувати на власний розсуд у своєму коді. Це дозволяє пришвидшити налаштування великої мережі, адміністратори можуть використовувати модулі інших розробників. Однак слід зазначити, що `Perl` має певну низку проблем, пов'язаних з безпекою програмного коду, серед яких є надання можливості зловмисникам обходити механізми захисту від проходження каталогу та перезаписувати довільні файли за допомогою архівного файлу, що містить символічне посилання, і звичайного файлу з таким самим ім'ям. Отже, потрібно враховувати уразливості під час використання `Perl`.

Також слід згадати про мову програмування загального призначення, яка підходить для вирішення задач автоматизації мереж – `Python` [4]. Завдяки простому синтансису мова програмування `Python` є легкою у вивченні. Гнучність розробки, синтаксис, велика кількість фреймворків та бібліотек, доступні інструменти кібербезпеки надають незрівнянні переваги поміж інших мов програмування. Ще однією перевагою є те, що вона є кросплатформеною, тобто працює на всіх операційних системах. Через це мережні інженери охоче використовують її при написані скриптів для конфігурації мережного обладнання. Також її використовують для створення корисного навантаження, яке використовується для аналізу шкідливого програмного забезпечення, сканування мережі, доступу до серверів, декодування пакетів та сканування портів. `Python` у кібербезпеці використовують для автоматизації, що робить збір інформації набагато легшим та потребує менше часу. Проте існують незначні недоліки у вигляді найбільшої кількості кібератак з боку хакерів через те, що `Python` є простим у розумінні. На сьогоднішній день більшість мережних пристроїв розробляються без API (Application Program Interface), наразі єдиним способом управління є застосування CLI (Command Line Interfaces). Передбачалось, що CLI будуть використовувати тільки інженери. Але через велику кількість модернізації мережних пристроїв, було важко проводити конфігурацію пристроїв вручну. Мова `Python` має дві найважливіші бібліотеки Pexpect та Paramiko, які застосовуються для вирішення таких задач у безпроводових мережах.

Нові підходи до побудови та розгортання сучасних мереж, що базуються на архітектурах програмно-конфігурованих мереж (Software-Defined Networking, SDN), вимагають використання засобів автоматизації взаємодії мережних застосунків, контролерів і пристроїв [5]. Таким чином, програмні засоби керують мережними пристроями. Отже, мови програмування, із застосуванням яких створюється відповідне програмне забезпечення, є ключовими для функціонування SDN.

Саме автоматизація та безпека є пріоритетними завданнями у межах перспективних мережних рішень. Спираючись на ці потреби та порівнявши існуючі мови програмування, що використовуються для автоматизації проводових і безпроводових мереж, робимо висновки, що `Python` є найкращим варіантом для вирішення різноманітних завдань мережного інженера та безпосередньо для безпеки мережного програмного забезпечення.

Література

1. Bash Reference Manual. *The GNU Operating System and the Free Software Movement*. URL: https://www.gnu.org/savannah-checkouts/gnu/bash/manual/bash.html (дата звернення: 05.11.2022).

2. 5 Programming languages that network admins need to learn. *TechGig*. URL: https://content.techgig.com/hiring/5-programming-languages-that-network-admins-need-to-learn/articleshow/77179923.cms (дата звернення: 05.11.2022).

3. Top Programming Languages For Network Engineers. *Admin Hacks*. URL: https://adminhacks.com/programing-network-engineers.html (дата звернення: 05.11.2022).

4. Chou E. Mastering Python Networking: Your one-stop solution to using Python for network automation, programmability, and DevOps. Packt Publishing; 3rd edition, 2020. 576 с.

5. Black C., Goransson P., Culver T. Software Defined Networks: A Comprehensive Approach. Morgan Kaufmann, 2016. 436 с.