
NETWORK PROGRAMMABILITY

ACCB5029

Submission Date: 28/07/2023

Prepared by: Valentin Latinov Miladinov /2024732/

*Network Automation, Frameworks,
Tools, API, Security and Privacy*

Table of Contents

Introduction	3
Network Programmability.....	3
Network Automation	3
Types of Network Automation	3
Implementation of Network Automation	4
Automation Framework.....	4
Types of Automated Testing Frameworks	4
Linear Automation Framework	5
Modular Based Testing Framework.....	5
Library Architecture Testing Framework	6
Data-Driven Framework	6
Honourable mentions:	6
Types of Tools for Network Automation	7
Scheduling network configuration backups automatically:	7
Executing bulk configuration changes across the network:	7
Ensure bullet-grade network security:	7
Generating device information and configuration reports:	7
Supporting multi-vendor network devices:	7
Demonstrating network compliance:.....	7
API Standards	7
Types of API Architectures	8
REST API	8
SOAP	8
Critical analysis of Automation of Security and Compliance Policies and API Security and Privacy	8
Automation of Security and Compliance Policies	8
Security processes that can be automated:	8
API Security	9
API Security Standards:	9
Independent Learning Task	9
Conclusion.....	10
References.....	11

Introduction

In this report one can find the most important components, benefits and disadvantages of Network Programmability, API, Manual Maintenance, Automation, and the transfer from one to the other. Furthermore, this report covers Automation Frameworks and Tools, API Standards and Architectures and a critical analysis of Automation of Security and Compliance Policies and API Security and Privacy.

Network Programmability

There's no shortage of praise when it comes to Network Programmability and its positive effects on the current technology. People and companies both agree that Network Programmability is a very, very important part of the new computing world. To present their awe of this component, it would be better to present a definition for it first.

Network Programmability is a collection of software tools, which find use in deploying, verifying, managing, and troubleshooting network problems [1][2][3] and the number of connected devices is not concerning. [1][3] Network Programmability "glues" traditional networking technology (e.g., switches, routers, firewalls and more) with software development language (e.g., Python, JavaScript). This creates a fully automated, dynamic network. [3] This aims to lower or even eliminate risks, costs, time-consumption, human error, and it makes more space for innovation. [1][3]

Network Automation

Network Automation, oversimplified, is an overall replacement of manual processes dedicated to administration with a software; its goals: efficiency and reliability. In detail, the software takes place of administrative manual processes such as: configuring, managing, testing, optimizing, deploying, and operating of the physical and virtual devices in the same network. It manages all these tasks, then repeats the cycle. [4][5][6][7] As AI find use even in Network Automation, a possible error can be tracked, even before it occurs. [5] According to Cisco.com, up to 95% of network changes are done manually nowadays. However, manual changes could lead to errors and inconsistencies in the network. Furthermore, expanding network changes at scale might could be troublesome. This suggests more network downtime and nonremote troubleshooting time, things companies prefer to avoid. [4]

Types of Network Automation

Command-line or script-base automation – automating of network components; limited scope of automation; network engineer with a solid understanding of various programming languages is a must; more suitable for standard processes; too fragile - customized changes applied to the network infrastructure could be fatal.

Automation software – more efficient way of managing, configuring and provisioning a network, as it eliminates manual time-consuming command-line instructions, as well as it automates the network components efficiently and rapidly; discovering devices, tracking network components, and populating the network inventory with important device details (e.g., IP address and number of ports) is possible. Proven benefit is to have critical device in one, centralized repository. It allows one to execute multiple network tasks. [7]

Implementation of Network Automation

Knowing what Network Automation is, next step is to see how it's achieved. Divided in five steps, one could automate their network by:

Taking Inventory –desktops, laptops, tablets, and smartphones used by employees, network routers, Internet-of-Things devices, printers or any other office hardware; key component of this task – having a clear idea of own infrastructure (e.g., wired and wireless connection).

Choosing a Network Automation Platform – a good pick demands effort, because most corporate networks are manually built and “homegrown”; few components to take into consideration are out-of-the-box features, support, and extensibility.

Devising Policies - a good examination of capabilities is vital; policies (e.g., connecting to a public Wi-Fi network) demand creation upon every single network feature prior to deployment.

Rolling Out Gradually – automation demands consistency; could begin with either providing new device setup or automating one office location prior to all others; for Network Automation's optimal work, continuous integration is to happen.

Troubleshooting and Iterating – even after automation, in the grand scheme of things, it's change management, therefore corporation's senior management needs to be fully and constantly informed. [6]

Automation Framework

Automation Frameworks is a compilation of guidelines or rules, which are used to create and design test cases. [8][9][10] The guidelines in question may consist of coding standards, test-data handling methods, processes for storing tests results, object repositories, or information on how to access external resources. They have no lawful leverage but following them brings benefits. [8]

There are multiple benefits that Automation Frameworks offer: for one, there is higher test speed, test accuracy; it also provides test efficiency, minimal manual intervention, maximum test coverage; furthermore, it lowers the maintenance costs; last, but not least, it allows the reusability of codes, plus easy reporting, and recovery scenario. [8][9]

Types of Automated Testing Frameworks

There are different types of Automation Frameworks, each with its own build-up, positives, and negatives. While structuring a test plan, framework type is to be considered. Some commonly used ones are:

Linear Automation Framework

Else seen as record-and-playback framework; with it, testers don't have to write codes that create functions; the steps are written in sequence. Meanwhile, the tester records the process. To conclude the test, it plays the given script back automatically. [7][9]

Advantages:

- no automation expertise needed, since a custom-made code isn't needed;
- easy and timely recording; making it the hastiest framework;
- test workflow clarity due to scripts, being laid out in sequential manner;
- easiest to use;
- available to most testing tools nowadays;

Disadvantages:

- low script reusability; changed data equals test case manual modification;
- test case cannot be re-run with multiple sets;
- needed maintenance is effortful;
- not scalable. [7][9]

Modular Based Testing Framework

This framework requires the applications to be under a division of units, functions, or sections to be tested; further tests are unique for each group. The application breaks down to individual modules. They receive a unique test script per module. Afterwards, the scripts combine, building a hierarchy amongst the tests. These tests represent various test cases. Construction of an abstraction layer is a must. It protects the overarching module from changes made in individual sections. [7][8]

Advantages:

- modifications in the application demands changes only in the parent module and its associated unique test script, the rest of the application can stay untouched;
- effortless creation of test cases due to script reusability.

Disadvantages:

- the data is also hard-coded into the test scripts, due to separate execution, taking the chance to use multiple data sets away;
- programming knowledge is a must. [7][8]

Library Architecture Testing Framework

To put it shortly, this is Modular Framework, but with upgrades. The application is no more divided by units, functions, or sections, this framework only identifies similar tasks within the scripts and are furthermore grouped by function. Kept in library, these functions could be called upon by the scripts anytime they are needed. [7][8]

Advantages:

- easier and less costly maintenance and scalability;
- higher reusability than in Module Framework;

Disadvantages:

- the data is still hard-coded;
- technical expertise is a must;
- more time-consuming. [7][8]

Data-Driven Framework

Unlike the previously presented frameworks, this one's data isn't hard-coded. The data is separated from script logic; therefore data may be stored externally. This framework is to support testers that need repeatable tests on the same feature or function of a given application, with changed sets of data. Data-Driven Frameworks allow storing and manipulating the input and output parameters to test scripts that come from an external data source (e.g., Excel Spreadsheets, CSV files, SQL Tables Text Files and more). External data source receives the necessary data whenever needed by the test scripts that are connected to it. [7][8][9]

Advantages:

- multiple data sets are available for testing;
- testing a variety of scenarios quickly by changing the data, lowering the necessity for more scripts;
- changes to test scripts and changes to data are not as dependant on each other.

Disadvantages:

- highly-experienced, highly skilled tester is a must; the framework is rather tricky and effortful to set up;
- time-consuming set up. [7][8][9]

Honourable mentions:

Keyword-Driven Framework –scripting knowledge necessity is low; high code reusability; independent of the application under test; test scripts are buildable; costly set-up, time-consuming and difficult; many specifications to fill; skilled employee is a must; scalability causes more maintenance. Hybrid Test Automation Framework – a mixture of all the frameworks.

Types of Tools for Network Automation

The network automation tools vary; categorized either as vendor-specific or vendor-agnostic. Vendor-specific software simplifies network management and automation; organizations involved in multi-vendor sales find it difficult and costly to invest in software like this. In the other hand, managing all devices in a network, regardless of the vendor, is in the powers of the GUI-based platforms. They have simple interfaces and aren't as costly in the long run, in comparison to their vendor-specific counterparts.

Scheduling network configuration backups automatically:

Usage: configurations from network devices (e.g., routers and switches) back-ups are available;

Tools used: Network Configuration Automation and Change Management Software;

Further benefits: customizable time of backup, rising built-in TFTP servers, which ensures organization's continuity during peak hours.

Executing bulk configuration changes across the network:

Usage: changing of templates; standardizing configurations; rolling back to previous stable configurations, if errors are faced due to bulk changes; changes and configurations can be applied to the newly installed network devices whenever required.

Ensure bullet-grade network security:

Usage: alarming for unauthorized configuration changes; if any errors occurred to these changes, this tool could fix them, eliminating further possible breaches or network failures; monitoring.

Generating device information and configuration reports:

Usage: generating insightful reports in a variety of formats about overall network performance; analyses weak spots and fixes them accordingly;

Supporting multi-vendor network devices:

Usage: managing multiple network devices from different vendors;

Further benefits: network administration efficiency; lessen operational expenses.

Demonstrating network compliance:

Usage: audition of device configuration; providing compliance with internal security policies and standards; detecting weaknesses; fixing weaknesses. [7]

API Standards

There are unique sets of standards for different industries. Having API standards is a beneficial guide for users. These standards allow partners to rapidly consume APIs with ease. Standards help organizations to quickly capitalize on success – building one connection and reusing many times. If an API is to be delivered and reached out by an audience, it needs to have an external availability – it needs to make sense to people, outside of the organization. Banking API Standards would focus on security. Healthcare industry – on quick and easy access to information. [11]

Types of API Architectures

Connection between applications is due to clear protocols and architectures of API. API Architecture presents a framework of rules. It is used for creating software interfaces. Users thrive on functionality of a server because of these rules. Different architectures have different rules and structures. Following information is examples of different API Architectures. [12]

REST API

Modern and most popular, Representational State Transfer Architecture finds use in designing client-server applications. Being neither a protocol nor a standard, it finds implementation in many ways. This respectfully rises flexibility. JSON, HTML, XML, even plain text is usable for information transfer via HTTP. The usage of JSON is the most common for this architecture and it is probably due JSON being language-agnostic, it is readable by humans. [12]

SOAP

Abbreviation for Simple Object Access Protocol, SOAP is an official API protocol. It is maintained by W3C and it is one of the earliest architectures. Communication between applications is rather eased by it, when these applications are formed with different languages and platforms. Boosting security, isolation, consistency and durability, SOAP has WSDL and XML in its core. SOAP works with stateful and with stateless data exchanges. [12]

There are also some honourable mentions such as Apache Kafka (supports event streaming), AsyncAPI (supports building and maintaining event-driven architectures) and Remote Procedure Call (RPC, allows communication between different programs on a network), GraphQL that is used for data fetching. [12]

Critical analysis of Automation of Security and Compliance Policies and API Security and Privacy

Automation of Security and Compliance Policies

Technology inevitably grows and it grows by the minute. So do the security threats. To have a modern software delivery, security and compliance cannot be ignored. [13] IT infrastructure grows too, and this causes complexity– manual managing becomes bigger hassle by the day. Workforce being gradually distributed, workers contributing from home, making bigger number of devices to keep track of, which is to say the least, intimidating. This could cause lower quality of detection and acting upon possible issues. This creates variety of weak spots in the system, it becomes an easier target. It rises the possibility of having a downtime and reduced functionality. Redhat.com suggests that a breach lowers by 95% in cost if security automation is concluded. [14]

Security processes that can be automated:

Threat hunting – manual processes become too effortful, especially when the IT environments are more complex. Automating processes of identifying, validating and escalating threats faster could save from troubles. [14]

Security incident response – automated security tools work across the whole system, have faster response time, makes no mistakes. [14]

Endpoint protection –it detects, investigates, and acts upon malicious activities on endpoint devices – the riskiest in the infrastructure. [14]

API Security

As previously mentioned, API could work with personal data – that alone speaks loud enough about the importance of a good security. As mentioned, there are different kinds of API; furthermore, security of every API is each to their own.

REST API – it has number of protocols to secure it via data encryption. Another way of securing REST API is with tokens, which authorize communications to proceed.

SOAP API – it makes use of XML signatures and SAML tokens to control the data traffic, therefore preventing attackers from gaining access. SOAP APIs are easier to design and protect.

API Security Standards:

Vulnerabilities – identifying weak spots and restricting accordingly: SQL injections is a signature-based attack to be on the lookout for; usage of tighter rules for JSON paths and schemas is highly recommendable; rate limits protect API backends.

Encryption – the oldest trick in the bag: data gets encrypted, turning meaningful information to text that makes absolutely no sense to read and gets decrypted only by the end that possesses the proper decryption key.

Tokens – authorising “permit”;

OAuth and OpenID Connect – Open authorization (OAuth) decides how an application receives access tokens, OpenID Connect authenticates and identifies the end-user, lowering transfer of information, by tightening the spectrum of collaborators to only those with proper token or identity.

Throttling and Quotas – limit access to the system, so bandwidth stays protected; there are attacks that would attempt to overwhelm a system (DDoS attack); that’s the reason throttling sets a speed limit in data trafficking, whereas quotas regulate the amount of data allowed to transfer.

API Gateway – as a reverse proxy, it authenticates data going through; it authenticates based on predetermined standards.

Zero-Trust Approach – allows no traffic to be trusted – both external and internal – every user needs to be authenticated; this model wants to protect the system from unauthorized users – even those that use a device that was previously authenticated. [15]

Independent Learning Task

This whole point is based on material from netacad/cisco courses. This task showed what kind of developer program DevNet is with the help of Learning Labs, Sandboxes, Code Exchanges, Developer Support, Developer Documentation. Additionally, presented were Video courses and DevNet events. Sandboxes were the most helpful on a personal level. This task taught Software Development, Design Pattern, Version Control Systems, Coding Basics, Code Review and Testing, Understanding Data Formats with all their subgroups. Furthermore, details around API, their Design Styles, Architecture and so on (some of it was covered by this report itself). Next, the task introduced Network Fundamentals and then explained it in detail. Nearing the end, the course started showing information about Deploying. Furthermore, Cisco presented how it automates infrastructures, showcasing DevOps and SRE, Automation Tools, Basic Automation Scripting, Automation Testing – again, mostly things, covered by this report. The concluding point was Cisco showing its API Platform. Overall, this task was similar to a guideline, that helped me line out this report.

Conclusion

Technology and IT infrastructure grow so fast. Manual maintenance becomes unbearable. Automation saves companies a lot of effort, money, and worries, however it also opens new windows for hackers to be hackers. That said, automation has security implementations. Automation has a variety of opportunities to give, it is up to the company to pick the best one for itself, then implement it and keep it going.

References

- [1] D. Saif, "Why network programmability is a big deal and why should you care as a network engineer!," LinkedIn, <https://www.linkedin.com/pulse/why-network-programmability-big-deal-should-you-care-engineer-saif#:~:text=Network%20Programmability%20is%20currently%20defined,human%20errors%2C%20innovation%20and%20customization> (accessed Jul. 16, 2023).
- [2] J. Edelman, S. Lowe, and M. Oswalt, *Network Programmability and Automation: Skills for the next-Generation Network Engineer*, 1st ed. Sebastopol, CA: O'Reilly Media, Inc, 2018.
- [3] G. Singh, "What is network programmability and different types of methods for it?," Networking Signal, <https://www.networkingsignal.com/network-programmability-detail-explained/> (accessed Jul. 16, 2023).
- [4] "What is network automation?," Cisco, https://www.cisco.com/c/en_uk/solutions/automation/network-automation.html (accessed Jul. 18, 2023).
- [5] "What is network automation?: Juniper Networks UK&I," Juniper Networks, <https://www.juniper.net/gb/en/research-topics/what-is-network-automation.html> (accessed Jul. 18, 2023).
- [6] "What is network automation? how does it work?," Fortinet, <https://www.fortinet.com/resources/cyberglossary/network-automation> (accessed Jul. 18, 2023).
- [7] "What is network automation?" SolarWinds, <https://www.solarwinds.com/resources/it-glossary/network-automation> (accessed Jul. 18, 2023).
- [8] "Test automation frameworks," smartbear.com, <https://smartbear.com/learn/automated-testing/test-automation-frameworks/> (accessed Jul. 18, 2023).
- [9] "Most popular test automation frameworks with pros and cons of each," Software Testing Help, <https://www.softwaretestinghelp.com/test-automation-frameworks-selenium-tutorial-20/> (accessed Jul. 20, 2023).
- [10] T. Hamilton, "Test automation framework: What is, Architecture & Types," Guru99, <https://www.guru99.com/test-automation-framework.html> (accessed Jul. 20, 2023).
- [11] L. Defranchi, "API standards: API standards in different industries," Axway Blog, <https://blog.axway.com/learning-center/apis/basics/api-industry-standards> (accessed Jul. 22, 2023).
- [12] S. Dindi, "6 types of API architectures and how they work," MUO, <https://www.makeuseof.com/api-architecture-types-how-work/> (accessed Jul. 22, 2023).
- [13] J. Sahoo, "How organizations can automate security and compliance," OpsMx Blog |, <https://www.opsmx.com/blog/how-to-automate-security-and-compliance-with-opsmx/> (accessed Jul. 22, 2023).
- [14] "What is security automation?," Red Hat - We make open source technologies for the enterprise, <https://www.redhat.com/en/topics/automation/what-is-security-automation#:~:text=Security%20automation%20is%20the%20use,processes%2C%20applications%2C%20and%20infrastructure> (accessed Jul. 24, 2023).
- [15] "What is API security?," Fortinet, <https://www.fortinet.com/resources/cyberglossary/api-security> (accessed Jul. 26, 2023).