COURSE DESCRIPTION CARD - SYLLABUS

Course name
Automation of configuration, maintenance and testing of ICT system networks [S1Cybez1>AKUiTSST]

Course

Field of study
Cybersecurity

Year/Semester
4/7

Area of study (specialization)
–

Profile of study
general academic

Level of study
first-cycle

Course offered in
Polish

Form of study
full-time

Requirements
elective

Number of hours

Lecture
16

Laboratory classes
30

Other
0

Tutorials
0

Projects/seminars
12

Number of credit points

4,00

Coordinators

dr hab. inż. Maciej Sobieraj
maciej.sobieraj@put.poznan.pl

prof. dr hab. inż. Mariusz Głąbowski
mariusz.glabowski@put.poznan.pl

Lecturers

Prerequisites

Basic knowledge of concepts related to computer networks and programming in Python.

Course objective

The aim of the course is to develop knowledge and skills related to network automation and programmability. Students will gain practical experience in creating automation scripts, constructing API calls, using DevOps tools, and implementing network configuration management mechanisms.

Course-related learning outcomes

Knowledge:
1. The student understands the role of automation and programmability in network management. [K1_W07]
2. The student comprehends basic DevOps concepts in network operations and the role of tools like Ansible. [K1_W11]

3. The student knows the differences between XML, JSON, and YAML, as well as their applications in network management.[K1_W11]
4. The student understands the mechanisms of RESTCONF and NETCONF and the role of YANG in network automation. [K1_W11]

Skills:
1. The student can write Python scripts to automate network devices using the requests module.[K1_U02]
2. The student can construct and interpret API calls based on the HTTP protocol for network device management. [K1_U04]
3. The student creates Ansible playbooks and uses Jinja2 templates and YAML data structures to generate network configurations.[K1_U02]
4. The student utilizes DevOps tools to automate network tasks. [K1_U02]

Social competences:
1. The student understands the need for continuous learning in the rapidly evolving field of network automation. [K1_K01]
2. The student collaborates in project teams, effectively communicating ideas and solutions. [K1_K05]

Methods for verifying learning outcomes and assessment criteria

Learning outcomes presented above are verified as follows:

• Written Exam (Knowledge): Theoretical questions on DevOps concepts, YANG, RESTCONF, and NETCONF.
• Laboratories (Skills): Evaluation of Python scripts, correctness of Ansible playbooks, and applied Jinja2 templates.
• Team Project (Social Competencies): Design and implementation of a network automation project using specified tools during lab sessions.
In each form of the course assessment, the grade depends on the number of points the student earns relative to the maximum number of required points. Earning at least 50% of the possible points is a prerequisite for passing. The relationship between the grade and the number of points is defined by the Study Regulations. Additionally, the course completion rules and the exact passing thresholds will be communicated to students at the beginning of the semester through the university’s electronic systems and during the first class meeting (in each form of classes).

Programme content

1. Network Automation and Programmability
2. Python Programming in a Networking Context
3. Configuration Management and DevOps Tools
4. Network Automation Standards and Protocols

Course topics

1: Introduction to Network Automation and Programmability (90 min.)
• Definition and importance of network automation in infrastructure management.
• Benefits of automation: efficiency improvement, error reduction, scalability.
• Overview of programming environments supporting network automation (e.g., Cisco DevNet, AWS, GNS3).
• Introduction to tools such as Terraform, Puppet, and Chef in process automation.
2: Python Programming for Network Automation (Part 1) (90 min.)
• Introduction to Python fundamentals in the context of network automation.
• Creating simple scripts for automating network tasks (e.g., retrieving device configurations).
• Introduction to popular libraries supporting automation, such as paramiko (SSH) and netmiko.
• Demonstration of Python use cases in network management.
3: Python Programming for Network Automation (Part 2) (90 min.)
• requests module in Python: introduction to working with HTTP-based API calls.
• Creating and interpreting GET, POST, PUT, and DELETE requests for network devices.
• Examples of communication with network devices via API (e.g., Cisco REST API).

• Handling and processing API responses in JSON and XML formats.
4: Network Configuration Management with Ansible (Part 1) (90 min.)
• Introduction to Ansible as a network automation tool.
• Creating basic playbooks for configuring network devices.
• Structure of playbooks and use of Ansible modules for networks (e.g., i os_config, nxos_command).
• Introduction to dynamic inventory management for network devices.
5: Network Configuration Management with Ansible (Part 2) (90 min.)
• Creating network configuration templates using Jinja2.
• Using YAML structures in Ansible playbooks.
• Automating device state retrieval and reporting.
• Example projects: VLAN configuration, firmware updates.
6: Network Automation Standards - RESTCONF and NETCONF (90 min.)
• Differences between RESTCONF and NETCONF: applications and functionality.
• Creating RESTCONF calls to network devices using Python (requests).
• Implementing NETCONF using Python libraries (e.g., ncclient).
• Comparison of data transmission formats: JSON, XML, and YAML.
7: Data Models and YANG in Network Automation (90 min.)
• Role of YANG models in defining network data structures.
• Overview of tools supporting YANG models (e.g., Pyang, YANG Explorer).
• Implementation examples of YANG models in network management automation.
• Integration of YANG models with NETCONF and RESTCONF.
8: DevOps in Network Operations (90 min.)
• Introduction to DevOps pipelines: CI/CD (Continuous Integration/Continuous Deployment) in network environments.
• Overview of DevOps tools (Jenkins, Git, Docker) in network operations.
• Examples of automating network infrastructure lifecycle management.
• Course summary: applying knowledge to network automation projects.

Teaching methods

• Theoretical lectures with demonstrations.
• Laboratories with hands-on exercises in a network environment.
• Case study analysis.
• Problem-solving discussions.

Bibliography

Basic:
Omar Farooq, Mastering Python Networking, Packt Publishing, 2018.
Jeff Geerling, Ansible for DevOps: Server and Configuration Management for Humans, Midwestern Mac, LLC, 2015.
Cisco DevNet Documentation: https://developer.cisco.com/

Additional:
Omar Farooq, Mastering Python Networking, Packt Publishing, 2018.
Jeff Geerling, Ansible for DevOps: Server and Configuration Management for Humans, Midwestern Mac, LLC, 2015.
Cisco DevNet Documentation: https://developer.cisco.com/
RESTCONF and NETCONF Documentation: https://datatracker.ietf.org/doc/html/rfc8040
Online resources on YANG models and supporting tools such as Pyang: https://github.com/mbj4668/pyang

Breakdown of average student's workload

	Hours	ECTS
Total workload	118	4,00
Classes requiring direct contact with the teacher	58	2,00
Student's own work (literature studies, preparation for laboratory classes/ tutorials, preparation for tests/exam, project preparation)	60	2,00