



KARTA OPISU PRZEDMIOTU - SYLABUS

Nazwa przedmiotu

Automatyzacja konfiguracji, utrzymania i testowania sieci systemów teleinformatycznych
[S1Cybez1>AKUiTSST]

Przedmiot

Kierunek studiów

Cyberbezpieczeństwo

Rok/Semestr

4/7

Studia w zakresie (specjalność)

–

Profil studiów

ogólnoakademicki

Poziom studiów

pierwszego stopnia

Język oferowanego przedmiotu

polski

Forma studiów

stacjonarne

Wymagalność

obieralny

Liczba godzin

Wykład

16

Laboratorium

30

Inne

0

Ćwiczenia

0

Projekty/seminaria

12

Liczba punktów ECTS

4,00

Koordynatorzy

dr hab. inż. Maciej Sobieraj

maciej.sobieraj@put.poznan.pl

prof. dr hab. inż. Mariusz Głabowski

mariusz.glabowski@put.poznan.pl

Wykładowcy

Wymagania wstępne

Znajomość podstawowych pojęć związanych z sieciami komputerowymi i programowaniem w Pythonie.

Cel przedmiotu

Celem przedmiotu jest rozwinięcie wiedzy i umiejętności związanych z automatyzacją i programowalnością sieci. Studenci zdobędą praktyczne doświadczenie w tworzeniu skryptów automatyzujących, konstruowaniu wywołań API, korzystaniu z narzędzi DevOps oraz implementowaniu mechanizmów zarządzania konfiguracjami sieciowymi.

Przedmiotowe efekty uczenia się

Wiedza:

1. Student/ka zna rolę automatyzacji i programowalności w zarządzaniu sieciami. [K1_W07]
2. Student/ka rozumie podstawowe koncepcje DevOps w operacjach sieciowych oraz rolę narzędzi

takich jak Ansible. [K1_W11]

3. Student/ka zna różnice między XML, JSON, YAML i przypadki ich zastosowania w zarządzaniu sieciami. [K1_W11]

4. Student/ka rozumie mechanizmy RESTCONF i NETCONF oraz rolę YANG w automatyzacji sieci. [K1_W11]

Umiejętności:

1. Student/ka potrafi tworzyć skrypty w Pythonie do automatyzacji urządzeń sieciowych z użyciem modułu requests. [K1_U02]

2. Student/ka potrafi konstruować i interpretować wywołania API oparte na protokole HTTP do zarządzania urządzeniami sieciowymi. [K1_U04]

3. Student/ka tworzy playbooks Ansible i korzysta z szablonów Jinja2 oraz struktur danych YAML w celu generowania konfiguracji sieciowych. [K1_U02]

4. Student/ka wykorzystuje narzędzia DevOps w celu automatyzacji zadań sieciowych. [K1_U02]

Kompetencje społeczne:

1. Student/ka rozumie potrzebę ciągłego aktualizowania wiedzy w dynamicznie rozwijającym się obszarze automatyzacji sieci. [K1_K01]

2. Student/ka współpracuje w zespołach projektowych, efektywnie komunikując swoje pomysły i rozwiązania. [K1_K05]

Metody weryfikacji efektów uczenia się i kryteria oceny

Efekty uczenia się przedstawione wyżej weryfikowane są w następujący sposób:

- Egzamin pisemny (wiedza): Pytania teoretyczne dotyczące koncepcji DevOps, YANG, RESTCONF i NETCONF.

- Laboratoria (umiejętności): Ocena skryptów Pythona, poprawność playbooków Ansible i zastosowanych szablonów Jinja2.

- Projekt zespołowy (kompetencje społeczne): Przygotowanie i implementacja projektu automatyzacji sieci z wykorzystaniem wskazanych narzędzi w trakcie zajęć laboratoryjnych.

W każdej formie zaliczenia przedmiotu ocena zależy od liczby zdobytych przez studenta punktów w stosunku do maksymalnej liczby punktów obowiązkowych. Warunkiem pozytywnego zaliczenia jest otrzymanie co najmniej 50% punktów możliwych do zdobycia. Zależność oceny od liczby punktów definiuje Regulamin Studiów. Dodatkowo zasady zaliczania przedmiotu i dokładne progi zaliczeniowe zostaną przekazane studentom na początku semestru z wykorzystaniem uczelnianych systemów elektronicznych oraz na pierwszych zajęciach (w każdej formie zajęć).

Treści programowe

1. Automatyzacja sieci i programowalność
2. Programowanie w Pythonie w kontekście sieci
3. Zarządzanie konfiguracją i narzędzia DevOps
4. Standardy i protokoły automatyzacji sieci

Tematyka zajęć

1: Wprowadzenie do automatyzacji sieci i programowalności (90 min.)

- Definicja i znaczenie automatyzacji sieci w zarządzaniu infrastrukturą sieciową.

- Korzyści z automatyzacji: poprawa efektywności, redukcja błędów, skalowalność.

- Przegląd środowisk programistycznych wspierających automatyzację sieci (np. Cisco DevNet, AWS, GNS3).

- Wprowadzenie do roli narzędzi takich jak Terraform, Puppet, Chef w automatyzacji procesów.

2: Programowanie w Pythonie dla automatyzacji sieci (cz. 1) (90 min.)

- Omówienie podstaw Pythona w kontekście automatyzacji sieci.

- Tworzenie prostych skryptów automatyzujących zadania sieciowe (np. pobieranie konfiguracji urządzeń).

- Wprowadzenie do popularnych bibliotek wspierających automatyzację, takich jak `paramiko` (SSH) i `netmiko`.

- Prezentacja przykładowych scenariuszy użycia Pythona w zarządzaniu siecią.

3: Programowanie w Pythonie dla automatyzacji sieci (cz. 2) (90 min.)

- Moduł `requests` w Pythonie: wprowadzenie do pracy z wywołaniami API opartymi na protokole HTTP.
 - Tworzenie i interpretacja wywołań GET, POST, PUT i DELETE dla urządzeń sieciowych.
 - Przykłady komunikacji z urządzeniami sieciowymi za pomocą API (np. Cisco REST API).
 - Obsługa i przetwarzanie odpowiedzi API w formatach JSON i XML.
- 4: Zarządzanie konfiguracją sieci z wykorzystaniem Ansible (cz. 1) (90 min.)
- Wprowadzenie do Ansible jako narzędzia automatyzacji sieci.
 - Tworzenie podstawowych playbooków do konfiguracji urządzeń sieciowych.
 - Struktura playbooków i wykorzystanie modułów Ansible dedykowanych dla sieci (np. `ios\_config`, `nxos\_command`).
 - Wprowadzenie do dynamicznych inwentaryzacji urządzeń sieciowych.
- 5: Zarządzanie konfiguracją sieci z wykorzystaniem Ansible (cz. 2) (90 min.)
- Tworzenie szablonów konfiguracji sieciowych przy użyciu Jinja2.
 - Wykorzystanie struktur YAML w playbookach Ansible.
 - Automatyzacja pobierania danych o stanie urządzeń i ich raportowanie.
 - Przykładowe projekty: konfiguracja VLAN, aktualizacja oprogramowania urządzeń.
- 6: Standardy automatyzacji sieci - RESTCONF i NETCONF (90 min.)
- Różnice między RESTCONF i NETCONF: zastosowania i funkcjonalność.
 - Tworzenie wywołań RESTCONF do urządzeń sieciowych za pomocą Pythona (`requests`).
 - Implementacja NETCONF z wykorzystaniem bibliotek Pythona (np. `ncclient`).
 - Prezentacja różnic w transmisji danych między JSON, XML, a YAML.
- 7: Modele danych i YANG w automatyzacji sieci (90 min.)
- Rola modeli YANG w definiowaniu struktury danych sieciowych.
 - Przegląd narzędzi wspierających pracę z modelami YANG (np. Pyang, YANG Explorer).
 - Przykłady implementacji modeli YANG w automatyzacji zarządzania sieciami.
 - Integracja modeli YANG z NETCONF i RESTCONF.
- 8: DevOps w operacjach sieciowych (90 min.)
- Wprowadzenie do potoków DevOps: CI/CD (Continuous Integration/Continuous Deployment) w kontekście sieci.
 - Omówienie narzędzi DevOps (Jenkins, Git, Docker) w operacjach sieciowych.
 - Przykłady automatyzacji cyklu życia infrastruktury sieciowej.
 - Podsumowanie kursu: zastosowanie wiedzy w projektach automatyzacji.

Metody dydaktyczne

- Wykłady teoretyczne z elementami demonstracji.
- Laboratoria z ćwiczeniami praktycznymi w środowisku sieciowym.
- Analiza case studies.
- Dyskusje problemowe.

Literatura

Podstawowa:

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

Uzupełniająca:

- Omar Farooq, Mastering Python Networking, Packt Publishing, 2018.
- Jeff Geerling, Ansible for DevOps: Server and Configuration Management for Humans, Midwestern Mac, LLC, 2015.
- Dokumentacja Cisco DevNet: <https://developer.cisco.com/>
- Dokumentacja RESTCONF i NETCONF: <https://datatracker.ietf.org/doc/html/rfc8040>
- Materiały online na temat modeli YANG i narzędzi wspierających, takich jak Pyang: <https://github.com/mbj4668/pyang>

Bilans nakładu pracy przeciętnego studenta

	Godzin	ECTS
Łączny nakład pracy	118	4,00
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	58	2,00
Praca własna studenta (studia literaturowe, przygotowanie do zajęć laboratoryjnych/ćwiczeń, przygotowanie do kolokwium/egzaminu, wykonanie projektu)	60	2,00