

ются разделение профессий на «мужские» и «женские», а также высокая семейная нагрузка, которая мешает женщинам строить карьеру.

Для решения этих проблем в рамках государственных программ важен комплексный подход. Государству необходимо развивать переобучение граждан под конкретные рабочие места, выдавать денежные субсидии на открытие своего дела и помогать выпускникам находить первую работу. При этом лучшим способом помочь женщинам является развитие дистанционного труда. Такой формат позволяет успешно совмещать декрет или уход за детьми с работой, сохраняя доход. Внедрение этих мер поможет создать равные условия для всех, снизить напряженность в обществе и эффективнее использовать трудовые ресурсы страны.

СПИСОК ЛИТЕРАТУРЫ

1 О занятости населения Республики Беларусь : Закон Респ. Беларусь, 15 июня 2006 г., № 125-З : в ред. от 11 нояб. 2024 г. // КонсультантПлюс. Беларусь : справ. правовая система (дата обращения: 17.05.2026).

2 Трудовой кодекс Республики Беларусь : [принят Палатой представителей 8 июня 1999 г. : одобр. Советом Респ. 24 июня 1999 г.] : в ред. от 20 июля 2024 г. // ЭТАЛОН : информ.-поисковая система (дата обращения: 17.05.2026).

3 Динамика численности рабочей силы // Национальный статистический комитет Республики Беларусь. – URL: www.belstat.gov.by (дата обращения: 17.05.2026).

4 **Медведева, И.** Разница в оплате труда мужчин и женщин в Беларуси составляет 5,8 % / И. Медведева // БелТА. – URL: www.belta.by (дата обращения: 17.05.2026).

Получено 02.06.2026

ISSN 2227-1155. Сборник студенческих научных работ.
Вып. 31. Гомель, 2026

УДК 004.43:004.056

В. Н. ЕРМАКОВА (ЭМ-21), Д. В. МИНИНА (ГИ-31), Т. А. ПЕТРОВА (ЭМ-31)
Научный руководитель – ст. преп. *В. Е. МИНИН*

ИСПОЛЬЗОВАНИЕ PYTHON В КИБЕРБЕЗОПАСНОСТИ

В современную эпоху цифровых технологий кибербезопасность становится одним из важнейших аспектов защиты информации и инфраструктуры. В условиях постоянного роста количества угроз, необходимости быстрого реагирования и автоматизации процессов выбор эффективных инструментов и языков программирования приобретает особое значение. Python, благодаря своей универсальности, просто-

те и богатству библиотек, занимает лидирующие позиции среди языков, используемых в сфере кибербезопасности.

Python – это интерпретируемый, высокоуровневый язык программирования, который отличается понятным синтаксисом и широкой экосистемой. Эти свойства позволяют специалистам быстро разрабатывать новые инструменты, автоматизировать рутинные задачи, проводить анализ данных и создавать сложные системы защиты. Его используют в самых разных областях кибербезопасности: от анализа угроз и проведения разведки до разработки систем обнаружения вторжений и криптографических решений [1]. Области применения Python в кибербезопасности:

1 Анализ угроз и разведка. Python позволяет быстро собирать информацию о потенциальных атаках, анализировать логи, парсить веб-страницы и получать инсайты о новых уязвимостях. Используя библиотеки, такие как Beautiful Soup и Requests, можно автоматизировать сбор данных, а также интегрировать их в системы SIEM и системы автоматического реагирования. Важным аспектом является использование машинного обучения для классификации угроз, кластеризации активности и определения новых типов атак на основе анализа данных.

2 Форензика и расследование инцидентов. Фреймворк Volatility – ключевой инструмент для анализа памяти компьютеров и серверов. Он позволяет извлекать артефакты, находить криптографические ключи, восстанавливать процессы и следы вредоносных действий. Использование Python в этом контексте обеспечивает гибкость и возможность автоматизации сложных процедур. Современные исследования включают использование GPU-ускорения для анализа больших объемов данных и ускорения обработки сложных сценариев.

3 Криптография. Python широко используется для реализации криптографических алгоритмов: симметричного шифрования (AES, DES) и асимметричного (RSA, ECC). Библиотеки, такие как PyCryptodome и cryptography, позволяют быстро создавать прототипы систем шифрования, реализовывать цифровые подписи и управлять ключами, внедрять криптографические протоколы в IoT-устройствах, мобильных приложениях и облачных системах.

Пример использования AES в Python:

```
from Crypto.Cipher import AES
from Crypto.Random import get_random_bytes

key = get_random_bytes(16)
cipher = AES.new(key, AES.MODE_GCM)
plaintext = b"Sensitive data"
ciphertext, tag = cipher.encrypt_and_digest(plaintext)

# Декодирование
cipher_dec = AES.new(key, AES.MODE_GCM, nonce=cipher.nonce)
```

```
decrypted_data = cipher_dec.decrypt_and_verify(ciphertext, tag)
```

4 Создание собственных инструментов и автоматизация атак. Python используется при разработке сканеров уязвимостей, автоматических эксплойтов, ботов-краулеров, инструментов для проведения тестов на проникновение (Pentest Frameworks). Обеспечивает взаимодействие с сетевыми протоколами, автоматизацию эксплуатации уязвимостей с помощью Pymetasploit, Nmap, Scapy. В последние годы активно развивается автоматизация атак с помощью AI и машинного обучения, что позволяет создавать более сложные и адаптивные злоумышленнические инструменты.

5 Облачные платформы, автоматизация и DevSecOps. Python используется для автоматизации развертывания, мониторинга и защиты облачных инфраструктур, при интеграции с API AWS, Azure, GCP для автоматического обновления правил, настройки брандмауэров, сбора метрик и автоматического реагирования. Участвует в разработке CI/CD процессов с учетом безопасности. Обеспечивает использование Infrastructure as Code (IaC) и автоматизированных тестов безопасности для выявления уязвимостей еще на этапе разработки.

6 Обучение и применение машинного обучения, искусственного интеллекта. Используется при создании моделей для обнаружения аномалий, классификации вредоносных программ, предсказания атак и автоматического реагирования. Позволяет использовать TensorFlow, Keras, Scikit-learn, XGBoost, внедрять системы обнаружения угроз на базе нейронных сетей, проводить анализ поведения пользователей и машинного обучения для предиктивной безопасности.

Python-экосистема насчитывает более 500 000 пакетов. Рассмотрим некоторые библиотеки языка, которыми пользуются профессионалы в области кибербезопасности:

1 Библиотеки сетевой безопасности, такие как nmap (Network Mapper) и twisted, часто используются для сканирования сетевого уровня, идентификации портов и идентификации различных протоколов приложений.

2 BeautifulSoup используется для сбора скрап-данных с веб-сайтов в виде XML (Extensible Markup Language) и HTML (Hyper Text Markup Language).

3 YARA используется для выявления и классификации вредоносного ПО.

4 Mechanize используется для взаимодействия с данными. Он также применяется в XSS (Cross Side Scripting) и SQL Injection. Он может использоваться для сбора данных на сайтах.

5 Pymetasploit3 обеспечивает возможность использования фреймворка Metasploit этой библиотеки для поиска различных ошибок на различных средах или платформах, например мобильных, Linux или Windows. Metasploit содержит большое количество полезных нагрузок.

6 Scapy создает и обрабатывает любые сетевые пакеты, также поддерживает декодирование и анализ пакетов (небольшой сегмент большого сообщения).

7 Криптографические библиотеки (cryptography, hashlib, PyNaCl, python-npurg и другие) используются для защиты или шифрования любого файла с помощью криптографических методов.

На сегодня Python является де-факто стандартом для задач анализа данных, автоматизации рутинных операций (обработка логов, парсинг пакетов) и создания прототипов эксплойтов. Это было достигнуто благодаря трем факторам: низкий порог входа, обширная библиотечная база (Scapy, Impacket, Cryptodome) и кроссплатформенность. Однако узким местом остается производительность и управление памятью, что делает Python неприменимым для задач, требующих максимального быстродействия или выполнения низкоуровневых операций.

Несмотря на указанные недостатки, перспективы использования Python в сфере кибербезопасности остаются высокими, так как этот язык адаптируется под самые современные вызовы ИТ-индустрии. Ключевые направления:

1 Интеграция с системами искусственного интеллекта. В кибербезопасности это будет выражено в создании адаптивных детекторов аномалий, которые не просто сигнатурно сравнивают, а семантически анализируют поведенческие паттерны.

2 Автоматизация реагирования на инциденты. Python станет связующим звеном между SIEM-системами, базами данных угроз и API облачных провайдеров и позволит сократить время автоматического реагирования на инциденты.

3 Работа в гибридных различных системах. Ядро таких систем разрабатывается на Rust или Go, а настройка для логики и отчетности – на Python (например, проект PyO3). В кибербезопасности такой подход позволит создавать полноценные песочницы и эмуляторы сетей, где Python управляет сценариями атак, а низкоуровневый код обеспечивает изоляцию и производительность.

4 Реверс-инжиниринг протоколов IoT и SCAD. С увеличением роста числа проприетарных промышленных протоколов растет и потребность в исследовании и быстром написании парсеров и фаззеров. Python с его динамической типизацией и REPL-средой останется безальтернативным выбором для исследования неизвестных бинарных форматов.

5 Безопасность самого кода на Python. Востребованы статические анализаторы нового поколения, использующие машинное обучение для выявления уязвимостей логики приложений, написанных на том же самом Python.

Практическая ценность Python в области защиты информационных систем обусловлена не столько его вычислительными характеристиками, сколько зрелой экосистемой и лаконичностью синтаксиса, позволяющей минимизировать время между возникновением задачи и реализацией рабочего прототипа. Данное свойство делает язык востребованным на всех этапах обеспечения безопас-

ности: от аудита конфигураций до оркестрации средств реагирования, что подтверждает его статус стандартного инструмента в стеке инженерных ролей, связанных с администрированием и защитой инфраструктуры.

В долгосрочной перспективе эволюция Python в кибербезопасности будет определяться смещением его функциональной нагрузки от исполнительного ядра к управляющему слою. Язык не будет конкурировать с системными компилируемыми средствами в области низкоуровневой обработки данных, однако сохранит безусловное преимущество как интеграционная прослойка, обеспечивающая связанность гетерогенных защитных модулей. Пока будет сохраняться потребность в извлечении информации из сырых данных и формализации процедур реагирования, Python будет выполнять роль не столько инструментальной утилиты, сколько операционной среды для выработки и верификации стратегий безопасности.

СПИСОК ЛИТЕРАТУРЫ

1 Как Python помогает с задачами безопасности? // Слёрм. – URL: <https://slurm.io/blog/tpost/a64b11e861-kak-python-pomogaet-s-zadachami-bezopasn/> (дата обращения: 22.03.2026).

2 Бухарев, Р. С. Хакинг на Python / Р. С. Бухарев. – М. : Наука и Техника, 2025. – 368 с.

3 Зейтц, Д. Black Hat Python: программирование для хакеров и пентестеров / Д. Зейтц, Т. Арнольд ; пер. с англ. А. Павлова. – 2-е изд. – СПб. : Питер, 2022. – 224 с.

Получено 28.05.2026

ISSN 2227-1155. Сборник студенческих научных работ.
Вып. 31. Гомель, 2026

УДК 656.072

М. В. ЖИГАЛОВ (УД-41)

Научный руководитель – ст. преп. *И. М. ЛИТВИНОВА*

АНАЛИЗ И ПЕРСПЕКТИВЫ РАЗВИТИЯ ЖЕЛЕЗНОДОРОЖНЫХ ПАССАЖИРСКИХ ПЕРЕВОЗОК ПО МАРШРУТУ МИНСК – НАЦИОНАЛЬНЫЙ АЭРОПОРТ МИНСК

Железнодорожный транспорт занимает важное место в системе перевозок Республики Беларусь. Маршрут Минск – Национальный аэропорт Минск является важным звеном транспортной инфраструктуры столицы Беларуси, соединяя деловые и туристические потоки, способствуя развитию экономики и улучшая качество обслуживания пассажиров.

Развитие пассажирского железнодорожного сообщения имеет важную социально-экономическую значимость. Обеспечение транспортной доступ-