СИСТЕМНЫЙ АНАЛИЗ

УДК 004.056:355.233 **DOI: 10.31799/2077-5687-2025-5-3-9**

НАУЧНЫЕ ПОДХОДЫ К МОДЕЛИРОВАНИЮ УГРОЗ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ В ВОЙСКАХ НАЦИОНАЛЬНОЙ ГВАРДИИ С ИСПОЛЬЗОВАНИЕМ ЯЗЫКА ПРОГРАММИРОВАНИЯ PYTHON

П. Н. Афонин
Военная ордена Жукова академия войск национальной гвардии Российской Федерации

В статье рассматривается проблема повышения устойчивости систем управления и информационных инфраструктур войск национальной гвардии (Росгвардии) к кибервоздействиям. Цель работы – разработка научно обоснованной методологии построения динамической модели угроз информационной безопасности (ИБ) с использованием языка программирования Python. Актуальность исследования обусловлена возрастающей сложностью и скоординированностью кибератак на объекты критической информационной инфраструктуры (КИИ), находящиеся в зоне ответственности Росгвардии. В работе предложена архитектура модели, основанная на концепции цифрового двойника защищаемой системы, и детализирован процесс ее параметризации, сочетающий данные мониторинга и формализованные результаты экспертной оценки. Особое внимание уделено анализу чувствительности модели, позволяющему выявить наиболее критические компоненты системы с точки зрения информационной безопасности. Практическая значимость исследования заключается в создании инструментария для проактивного выявления уязвимостей, оценки рисков и оптимизации ресурсов, выделяемых на защиту информации.

Ключевые слова: информационная безопасность, моделирование угроз, войска национальной гвардии, Python, цифровой двойник, анализ чувствительности, экспертные оценки, граф атак.

Для цитирования:
Афонин, П. Н. Научные подходы к моделированию угроз информационной безопасности в войсках национальной гвардии с использованием языка программирования Python / П. Н. Афонин // Системный анализ и логистика. – 2025. – № 5(48). – с. 3-9. DOI: 10.31799/2077-5687-2025-5-3-9.

SCIENTIFIC APPROACHES TO MODELING INFORMATION SECURITY THREATS IN THE NATIONAL GUARD TROOPS USING THE PYTHON PROGRAMMING LANGUAGE

P. N. Afonin
Military Order of Zhukov Academy of the National Guard of the Russian Federation

The article discusses the problem of increasing the resilience of control systems and information infrastructures of the National Guard troops (Rosgvardiya) to cyber attacks. The purpose of the work is to develop a scientifically based methodology for building a dynamic threat model for information security using the Python programming language. The relevance of the study is due to the increasing complexity and coordination of cyber attacks on critical information infrastructure facilities located in the Rosgvardiya area of responsibility. The paper proposes a model architecture based on the concept of a digital twin of the protected system, and details the process of its parameterization, combining monitoring data and formalized expert evaluation results. Special attention is paid to the sensitivity analysis of the model, which makes it possible to identify the most critical components of the system from the point of view of information security. The practical significance of the research lies in the creation of tools for proactive vulnerability identification, risk assessment and optimization of resources allocated to information protection.

Keywords: information security, threat modeling, National Guard troops, Python, digital twin, sensitivity analysis, expert assessments, attack graph.

For citation:
Afonin, P. N. Scientific approaches to modeling information security threats in the national guard troops using the Python programming language / P. N. Afonin // System analysis and logistics. – 2025. – № 5(48). – p. 3-9. DOI: 10.31799/2077-5687-2025-5-3-9.

Введение

Эволюция киберугроз характеризуется переходом от спорадических атак к сложным, многоэтапным кампаниям, нацеленным на дестабилизацию деятельности государственных

институтов. Войска национальной гвардии (далее – ВНГ) Российской Федерации, выполняющие задачи по обеспечению общественной безопасности, защите объектов КИИ и обороноспособности страны, являются высокоприоритетной мишенью для подобных воздействий. Традиционные, статические подходы к оценке рисков ИБ, основанные на «чек-листах» и периодическом аудите, не успевают адаптироваться к динамично меняющейся тактике противника.

В связи с этим возникает научная и практическая задача создания адаптивной, формализованной и воспроизводимой системы моделирования угроз [1]. Такой подход должен не только идентифицировать известные уязвимости, но и прогнозировать потенциальные векторы атак на основе структуры и свойств моделируемой системы. Язык программирования Python, обладающий богатым набором библиотек для Data Science, сетевого анализа и математического моделирования, представляет собой идеальную платформу для реализации данной задачи.

1. Обзор современных научных подходов к моделированию угроз информационной безопасности:

Моделирование угроз информационной безопасности представляет собой процесс идентификации, анализа и оценки потенциальных угроз для информационных активов организации. Существует множество различных подходов к моделированию угроз, каждый из которых имеет свои преимущества и недостатки [2, 3].

1.1. Методология STRIDE основана на классификации угроз по категориям: Spoofing (подмена), Tampering (изменение), Repudiation (отказ), Information Disclosure (разглашение информации), Denial of Service (отказ в обслуживании), Elevation of Privilege (повышение привилегий) и позволяет систематизировать угрозы и определить меры защиты для каждой категории. Вместе с тем, STRIDE ограничена в плане учета сложных, комбинированных атак.

1.2. Методология DREAD используется для оценки рисков, связанных с выявленными угрозами. Оценка проводится по пяти параметрам: Damage potential (потенциальный ущерб), Reproducibility (воспроизводимость), Exploitability (простота эксплуатации), Affected users (количество затронутых пользователей), Discoverability (простота обнаружения). DREAD позволяет приоритизировать угрозы и выбирать наиболее эффективные меры защиты, однако, она субъективна и зависит от квалификации экспертов.

1.3. Моделирование на основе атак (Attack trees) представляет собой графическое отображение возможных путей реализации угрозы, позволяет выявить слабые места в системе защиты и определить наиболее вероятные векторы атак, однако, требует глубокого знания архитектуры системы и возможных способов ее эксплуатации.

1.4. Агентное моделирование (Agent-based modeling) имитирует поведение различных акторов (злоумышленников, пользователей, системных администраторов) и их взаимодействие в информационной системе, позволяет оценить влияние различных факторов на безопасность системы и выявить эмерджентные свойства, которые не могут быть обнаружены с помощью других методов, однако, требует значительных вычислительных ресурсов и сложной реализации.

1.5. Использование искусственного интеллекта (ИИ) и машинного обучения (МО) может использоваться для автоматизации процессов идентификации, анализа и оценки угроз. При этом алгоритмы МО могут обучаться на исторических данных об атаках и выявлять аномалии, которые могут указывать на подготовку к новым атакам. ИИ может использоваться для создания самообучающихся систем защиты, способных адаптироваться к изменяющимся угрозам, однако, данный подход требует больших объемов данных и квалифицированных специалистов в области ИИ.

2. Использование языка программирования Python для моделирования угроз информационной безопасности в ВНГ:

Python является одним из самых популярных языков программирования в мире, благодаря своей простоте, гибкости и большому количеству библиотек для работы с данными, машинным обучением и сетевыми технологиями. Использование Python для моделирования угроз ИБ в ВНГ имеет ряд преимуществ:

2.1. Простота и скорость разработки: Python имеет простой и понятный синтаксис, что позволяет быстро разрабатывать и прототипировать модели угроз.

2.2. Широкий выбор библиотек: Python предоставляет широкий выбор библиотек для работы с данными (NumPy, Pandas), машинным обучением (Scikit-learn, TensorFlow, PyTorch), сетевыми технологиями (Scapy, Nmap) и другими областями, необходимыми для моделирования угроз ИБ.

2.3. Возможность интеграции с другими системами: Python легко интегрируется с другими системами и приложениями, что позволяет использовать разработанные модели в реальных условиях.

2.4. Активное сообщество разработчиков: Python имеет большое и активное сообщество разработчиков, которое предоставляет поддержку и разрабатывает новые библиотеки и инструменты.

2.5. Поддержка моделирования различных видов угроз: Python позволяет моделировать различные виды угроз, такие как:

- сетевые атаки (DoS, DDoS, MITM).
- вредоносное программное обеспечение (вирусы, трояны, черви).
- социальную инженерию (фишинг, претекстинг).
- внутренние угрозы (нарушение политики безопасности, утечка данных).

2.6. Возможность автоматической генерации отчетов и визуализации данных: Python предоставляет инструменты для автоматической генерации отчетов о результатах моделирования и визуализации данных, что облегчает анализ и принятие решений.

Примеры практического применения Python для моделирования угроз ИБ в ВНГ:

- моделирование сетевых атак: с использованием библиотеки Scapy можно создавать пакеты различных типов и имитировать сетевые атаки, такие как DoS, DDoS, MITM. Это позволяет тестировать устойчивость сетевой инфраструктуры ВНГ к различным видам атак;
- анализ вредоносного программного обеспечения: с использованием библиотек VirusTotal API и YARA можно анализировать вредоносное программное обеспечение и выявлять его характеристики, а также разрабатывать сигнатуры для обнаружения вредоносных программ;
- моделирование социальной инженерии: с использованием библиотек Requests и Beautiful Soup можно автоматизировать сбор информации о сотрудниках ВНГ из открытых источников (социальные сети, сайты и т.д.) и моделировать фишинговые атаки, что позволяет оценить уровень осведомленности сотрудников о правилах ИБ и эффективность проводимых мероприятий по их обучению;
- разработка систем обнаружения вторжений (IDS): с использованием библиотек Scikit-learn и TensorFlow можно разрабатывать системы обнаружения вторжений, которые обучаются на данных о нормальной и аномальной сетевой активности и автоматически выявляют подозрительные действия.

3. Авторская методология построения модели угроз

Моделирование угроз предлагается осуществлять в рамках многоуровневой архитектуры, ядром которой является «цифровой двойник» информационной инфраструктуры подразделения [4]. Цифровой двойник – это не статичная схема, а динамическая программная модель, отражающая топологию системы, свойства ее компонентов (активов) и бизнес-процессы (потоки данных), которые она поддерживает [5].

Формальная постановка задачи:

Пусть $A = \{a_1, a_2, ..., a_n\}$ – множество активов системы (серверы, рабочие станции, маршрутизаторы, базы данных, приложения).

Каждому активу a_i ставится в соответствие вектор атрибутов:

$$\overrightarrow{Q_i} = \left(C_i, I_i, A_i, \overrightarrow{D_i}, \overrightarrow{S_i}\right)$$

где C_i – ценность конфиденциальности актива;

I_i – ценность целостности

A_i – ценность доступности;

$\overrightarrow{D_i}$ – вектор зависимостей (какие активы требуются для функционирования a_i);

$\overrightarrow{S_i}$ – вектор соединений (с какими активами a_i взаимодействует).

Множество угроз $T = \{t_1, t_2, ..., t_m\}$ формализуется через тактики и техники фреймворка MITRE ATT&CK [6]. Каждая угроза t_j описывается кортежем (Pre_j, $Post_j$, P_j), где: Pre_j – предварительные условия для реализации угрозы (например, наличие уязвимости, уровень доступа);

$Post_j$ – пост-условия (результат успешной атаки);

P_j – вероятность реализации угрозы при выполнении условий Pre_j.

Взаимосвязь элементов моделируется с помощью ориентированного графа атак $G = (V, E)$, где вершины V представляют состояния системы (доступ к активам), а ребра E – возможные переходы между состояниями, осуществляемые посредством реализации угроз t_j.

4. Параметризация модели: интеграция данных и экспертных оценок

Ключевой проблемой является наполнение модели достоверными данными. Предлагается использовать комбинированный подход.

4.1. Источники объективных данных:

- базы конфигураций (CMDB): автоматизированный импорт списка активов и их взаимосвязей;
- базы уязвимостей (NVD, CVE): интеграция с национальными и международными базами для получения актуальных данных об уязвимостях.
- системы мониторинга (SIEM): данные о реальных событиях ИБ для калибровки вероятностей P_j.

4.2. Метод экспертных оценок для формализации нечисловых параметров:

Для определения таких параметров, как C_i, I_i, A_i, а также вероятности P_j для угроз, не имеющих достаточной статистики, применяется метод парных сравнений Т. Саати [7] в модификации для коллективного принятия решений. На рис. 1 представлен программный код на языке Python, описывающий представленный концепт.

```python
import numpy as np
from typing import List

class ExpertAssessment:
    def __init__(self, experts_weights: List[float]):
        self.experts_weights = np.array(experts_weights)
        self.experts_weights /= self.experts_weights.sum()  # Нормализация весов

    def calculate_priority_vector(self, comparisons_matrix: np.ndarray) -> np.ndarray:
        """Вычисляет вектор приоритетов на основе матрицы парных сравнений."""
        # Метод собственного вектора
        eigenvalues, eigenvectors = np.linalg.eig(comparisons_matrix)
        max_index = np.argmax(eigenvalues.real)
        priority_vector = eigenvectors[:, max_index].real
        priority_vector = priority_vector / priority_vector.sum()  # Нормализация
        return priority_vector

    def aggregate_expert_judgments(self, expert_matrices: List[np.ndarray]) ->
np.ndarray:
        """Агрегирует оценки нескольких экспертов в одну матрицу."""
        aggregated_matrix = np.zeros_like(expert_matrices[0])
        for weight, matrix in zip(self.experts_weights, expert_matrices):
            aggregated_matrix += weight * matrix
        return aggregated_matrix

# Пример: Оценка конфиденциальности активов тремя экспертами
# Шкала: 1 - равноважность, 3 - умеренное превосходство, 5 - сильное превосходство
expert1_matrix = np.array([[1, 3, 5], [1/3, 1, 2], [1/5, 1/2, 1]])  # Сравнение Актив1,
Актив2, Актив3
expert2_matrix = np.array([[1, 2, 4], [1/2, 1, 3], [1/4, 1/3, 1]])
expert3_matrix = np.array([[1, 4, 6], [1/4, 1, 2], [1/6, 1/2, 1]])

assessor = ExpertAssessment(experts_weights=[0.5, 0.3, 0.2])  # Веса экспертов
aggregated_matrix = assessor.aggregate_expert_judgments([expert1_matrix,
expert2_matrix, expert3_matrix])
confidentiality_scores = assessor.calculate_priority_vector(aggregated_matrix)

print("Вектор ценности конфиденциальности активов:", confidentiality_scores)
# Вывод: [0.633 0.260 0.106] -> Актив1 наиболее критичен.
```

Рис. 1. Программный код на языке Python, описывающий параметризацию модели с учетом возможности интеграции данных и экспертных оценок

5. Анализ чувствительности цифрового двойника

Для оценки устойчивости модели и выявления ключевых точек приложения защитных мер проводится анализ чувствительности. Его цель – определить, как изменения входных параметров модели (например, вероятность успеха отдельной атаки или ценность актива) влияют на итоговый агрегированный риск системы.

Математический аппарат основан на использовании метода Монте-Карло [8]. Параметры модели P_j, C_i варьируются в заданных диапазонах (например, ±20%) согласно заданным законам распределения (нормальное, равномерное). Для каждой сгенерированной комбинации параметров пересчитывается общий риск. На рис. 2 представлен программный код на языке Python, описывающий возможность анализа чувствительности цифрового двойника.

```python
python
import pandas as pd
import numpy as np
from SALib.sample import saltelli
from SALib.analyze import sobol

def risk_model(P_j, C_i):
    # Упрощенная функция расчета общего риска
    # В реальности здесь происходит полный проход по графу атак
    return P_j * C_i

# Определение параметров для анализа
problem = {
    'num_vars': 2,
    'names': ['P_j', 'C_i'],
    'bounds': [[0.1, 0.9],  # Диапазон для вероятности атаки
               [0.5, 1.5]]  # Диапазон для ценности актива (отн. ед.)
}

# Генерация выборки по схеме Соболя/Сальтелли
param_values = saltelli.sample(problem, 1000)

# Вычисление отклика модели для каждой точки
Y = np.array([risk_model(*params) for params in param_values])

# Анализ индексов Соболя
Si = sobol.analyze(problem, Y)
print("Индексы первого порядка S1:", Si['S1'])
print("Индексы общего влияния ST:", Si['ST'])
```

Рис. 2. Программный код на языке Python, описывающий возможность анализа чувствительности цифрового двойника.

Интерпретация результатов: высокое значение индекса Соболя первого порядка S_1 для параметра P_j будет указывать на то, что общий риск системы наиболее чувствителен к изменению вероятности данной атаки. Следовательно, меры по снижению именно этой вероятности (например, установка сигнатур в IPS) дадут максимальный эффект.

Заключение

Разработанная методика позволяет перейти от качественных к количественным оценкам в области управления рисками ИБ в подразделениях войск национальной гвардии России. Использование Python обеспечивает необходимую гибкость и масштабируемость модели, при этом существенную научную новизну представленного подхода составляют:

- синтез методологии цифрового двойника и аппарата анализа чувствительности для задач ИБ;
- формализованная процедура интеграции объективных данных и коллективных экспертных оценок для параметризации модели.

Внедрение подобной системы моделирования позволит специалистам по ИБ Войск национальной гвардии России обоснованно принимать решения о распределении ресурсов, оптимизировать конфигурацию систем защиты и отрабатывать на цифровом двойнике сценарии реагирования на инциденты до их возникновения в реальной системе.

Перспективы дальнейших исследований видятся в развитии модели в сторону агент-ориентированного подхода для имитации поведения злоумышленников и пользователей, а также в интеграции с системами оперативного управления для создания замкнутого цикла адаптивной безопасности.

СПИСОК ЛИТЕРАТУРЫ

1. *Афонин П. Н.* Информационное обеспечение в таможенных органах / П. Н. Афонин, И. А. Сальников. – Санкт-Петербург: Российская таможенная академия, 2006. – 392 с.
2. *Афонин П. Н.* Управление стеком технологических инноваций в войсках национальной гвардии / П. Н. Афонин // Системный анализ и логистика. – 2025. – № 3(46). – С. 66-71. – DOI 10.31799/2077-5687-2025-3-66-71.
3. *Афонин П. Н.* Формально-лингвистическая модель семематической структуры информации / П. Н. Афонин, Н. С. Яснова, П. А. Фалеев // Известия СПбГЭТУ ЛЭТИ. – 2023. – Т. 16, № 4. – С. 54-60. – DOI 10.32603/2071-8985-2023-16-4-54-60.
4. *Афонин П. Н.* Применение цифровых двойников в таможенном контроле / П. Н. Афонин, Е. В. Лобас, Н. А. Шемякин // Инновации. – 2021. – № 10(276). – С. 9-13. – DOI 10.26310/2071-3010.2021.276.10.002.
5. *Марков А. С.* Систематика уязвимостей и дефектов безопасности программных ресурсов / А. С. Марков, А. А. Фадин // Защита информации. Инсайд. – 2013. – № 3(51). – С. 56-61.
6. Моделирование таможенных информационных систем / П. Н. Афонин [и др.]. – СПб: Российская таможенная академия, 2020. – 128 с.
7. *Саати Т.* Принятие решений. Метод анализа иерархий / Т. Саати. – М.: Радио и связь, 1993. – 278 с.
8. MITRE ATT&CK® Framework. [Электронный ресурс]. – URL: https://attack.mitre.org/ (дата обращения: 26.10.2025).

ИНФОРМАЦИЯ ОБ АВТОРЕ

Афонин Петр Николаевич
Профессор, доктор. техн. наук, доцент
ФГКОУ ВО «Военная ордена Жукова академия войск национальной гвардии Российской Федерации»
198206, Россия, Санкт-Петербург, ул. Л. Пилютова, д. 1
E-mail: pnafonin@yandex.ru

INFORMATION ABOUT THE AUTHOR

Afonin Petr Nikolaevich
Dr. tech. Sciences, associate Professor
National Guard Troops Academy
1, L.Pilyutova str., Saint-Petersburg, 198206, Russia
E-mail: pnafonin@yandex.ru

Дата поступления: 16.09.2025
Дата принятия: 29.09.2025