ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

UDC 004.056:004.7:004.891:621.396.49 doi: 10.32620/reks.2025.1.17

Roman ODARCHENKO[1], Maksim IAVICH[2], Alla PINCHUK[1]

[1] *National Aviation University, Kyiv, Ukraine*
[2] *Caucasus University, Tbilisi, Georgia*

DEVELOPMENT OF A METHOD FOR AUTOMATED 5G AND BEYOND NETWORK SLICES PENETRATION TESTING

The subject *of this article is penetration testing methodologies for 5G networks and beyond.* ***The aim*** *of this paper is to develop a methodology and software for automated penetration testing of the network infrastructure of next-generation cellular networks with a layered architecture.* ***The tasks*** *to be solved are as follows: 1) to analyze existing penetration testing methods of 5G and beyond networks and research in this area; 2) to develop a new method for automated 5G and beyond network slices penetration testing; 3) to design and implement the methodology in the form of software for virtualize environments; 4) to develop a 5G test network architecture based on open-source solutions and methodology of experiments conducting; 5) to test and validate the solution effectiveness in detecting vulnerabilities and simulating realistic attack scenarios in the 5G test network environment. The following* ***results*** *were obtained: 1) the new method for automated 5G and beyond network slices penetration testing was developed, leveraging Genetic Algorithms to optimize attack strategies; 2) a software tool for automating penetration testing was implemented, enabling efficient detection of critical and high-severity vulnerabilities and simulating attacks in a complex 5G network environment; 3) a test network architecture was created for experimentation, enabling a controlled evaluation of the methodology; 4) the experimental results demonstrated the effectiveness and operability of the proposed method.* ***Conclusions.*** *The primary contribution of this research is the development of a methodology, which is implemented in software, to enhance and automate the penetration testing process. The results prove the operability and effectiveness of the proposed solutions, demonstrating improved vulnerability detection, optimized attack strategy generation, and a higher success rate of penetration tests in a complex network environment.*

Keywords: *5G and beyond; penetration testing; automate penetration testing; automated testing framework; 5G security.*

1. Introduction

A large number of 5th-generation cellular networks are currently deployed worldwide. All developed countries already have commercial 5G networks in operation, many others are still running test networks or have commercial cells in large locations. However, the world of telecommunications is developing rapidly, and new generations of cellular communication appear on average every 10 years; thus, we are already on the threshold of the deployment of 6th-generation cellular networks. Figure 1 presents a roadmap for developing cellular communication networks. Ongoing efforts are being made to enhance the current generation as we progress toward the next.

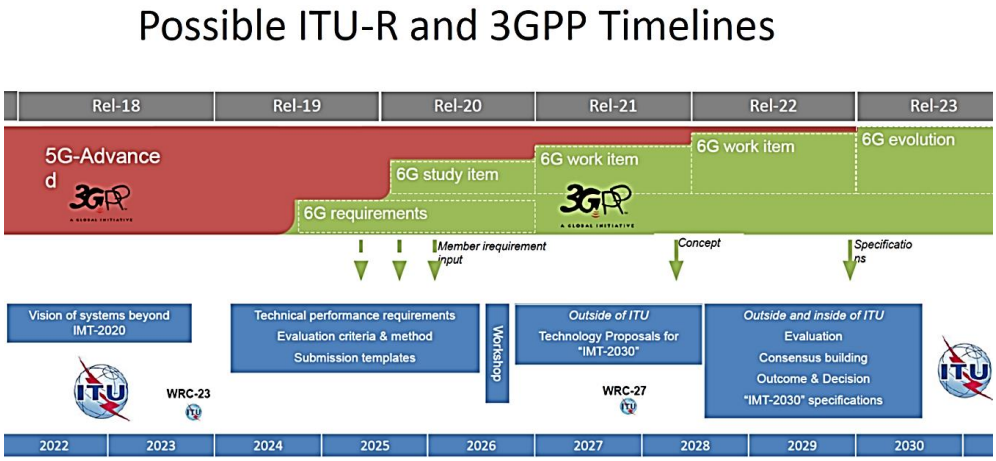

Fig. 1. Timeline of the development and standardization of mobile networks [1]

It is expected that 6G cellular networks will have significantly improved characteristics in terms of their efficiency both for communication operators and considerably improved characteristics in terms of subscriber service (QoE and QoS).

To achieve these extremely high-performance levels, several new technologies have been developed and continue to evolve. However, the best solutions that have been tested for a long time will be borrowed from previous generations. In this article, the focus is on the concept of "network slicing". 5G networks use this concept, which means that the 5G network infrastructure can be logically cut into “network layers” — “slices” for different business applications and for different radio access technologies (RAT) (fig. 2).

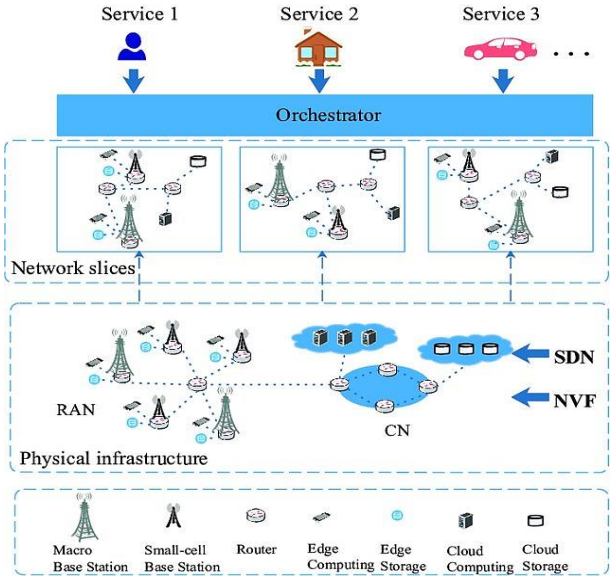

Fig. 2. Network slices concept [2]

These networks can be optimized separately for different data transfer rate requirements for different RATs. For example, a 4K video application requires high speed and is not critical to packet delay, whereas an NB-IoT application, on the contrary, is not demanding in terms of speed but, in some cases, requires fairly fast information delivery. The “Tactile Internet” application almost always requires the lowest possible delay. Future networks will enhance network slicing capabilities to support even more diverse and demanding applications [3].

At the same time, in addition to characteristics directly related to the quality of user service, cyber security tools, techniques, and mechanisms are also subject to optimization. Therefore, a large amount of modern research is devoted to analyzing the security systems of modern cellular networks, determining the weakest points, and, accordingly, eliminating identified shortcomings.

One of the key stages of ensuring the required level of cyber security is the detection of vulnerabilities in networks as a whole, their constituent parts, and penetration testing. Penetration testing is usually performed using manual or automated technologies to systematically disrupt network elements (switches, routers, etc.), IT infrastructure (servers, endpoints, web applications), and mobile devices and other potential points of influence. After successfully exploiting vulnerabilities in a particular system, testers may attempt to exploit the affected system for further attacks on other internal resources to achieve higher levels of access to electronic assets and information via privilege escalation.

Information about vulnerabilities that have been successfully exploited during penetration testing is usually collected and provided for use by information systems and network management. This allows them to draw strategic conclusions and prioritize appropriate efforts to eliminate problems. Therefore, the primary purpose of penetration testing is to assess the possibility of compromising systems or end-users and to assess the consequences of such incidents on the resources or operations involved.

However, for the large and complex architectures of modern networks managed by mobile operators, conducting penetration testing manually is challenging. Therefore, it is essential to leverage all available automation methods, incorporating the latest advancements in the field.

2. Literature Review

Recent research highlights the increasing complexity of 5G systems, which necessitates sophisticated automation techniques to identify vulnerabilities. Advances in machine learning (ML) and artificial intelligence (AI) have shown promise in automating various stages of security testing, including vulnerability detection and attack simulation, which are particularly relevant for dynamic and scalable environments like 5G slices [4, 5].

Network slicing introduces unique challenges, as each slice represents a logically isolated network tailored to specific use cases, requiring tailored security approaches. Hermosilla et al. [4] demonstrated how ML-driven methods can enhance lifecycle security by dynamically adapting to the evolving threats associated with each slice. Similarly, Kumar and Sharma [6] proposed frameworks for automating attack path discovery in dynamic network environments, underlining the importance of identifying and mitigating threats specific to 5G slices.

Automation tools for penetration testing in traditional networks often fall short when applied to 5G networks, primarily due to their decentralized

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

architecture, ultralow latency requirements, and diverse deployment scenarios [7, 8]. The introduction of intelligent agents and reinforcement learning-based models has addressed some of these limitations by enabling real-time decision-making and adaptability during penetration testing [9, 10]. These approaches facilitate the identification of critical vulnerabilities, such as those arising from misconfigured network functions or insecure APIs [11, 12].

Incorporating software-defined networking (SDN) and network function virtualization (NFV) into 5G networks has further complicated the security landscape. Recent studies have emphasized the role of automated testing frameworks in evaluating the security implications of SDN and NFV, particularly in maintaining isolation between slices and preventing lateral movement during attacks [13, 14]. The integration of simulation environments and digital twins has been proposed to provide a safe and efficient testing ground for assessing potential threats without compromising live systems [8, 15].

Furthermore, the need for comprehensive and scalable security solutions has driven interest in tools capable of orchestrating and executing multivector attacks across different slices to uncover hidden vulnerabilities [6]. Such tools leverage graph-based analysis and behavioral modeling to predict potential attack paths, providing a holistic view of network security [9]. The integration of these tools with automated testing frameworks can streamline penetration testing and reduce the time required to secure complex 5G infrastructures.

Existing literature consistently emphasizes the importance of addressing the unique security challenges posed by 5G networks and beyond. Automated penetration testing methods tailored to these environments are indispensable to safeguard the reliability and integrity of network slices, thereby enabling the secure deployment of diverse applications, ranging from critical infrastructure to consumer services [9, 13].

Reviewed existing solutions for 5G penetration testing and identified different gaps that should be filled. While advances in ML and AI techniques show promise in automating vulnerability detection and attack simulation, these methods often rely on historical data and struggle to detect novel threats or adapt to the rapidly changing configurations of 5G slices. Traditional automation tools designed for static and centralized networks are ill-suited to decentralized architectures, ultralow latency requirements, and complex interactions between IoT devices and edge computing in 5G environments. Solutions leveraging SDN and NFV, while addressing some challenges, introduce additional attack surfaces, such as insecure APIs and lateral movement within slices, and often lack real-time responsiveness and scalability for large infrastructures. Furthermore, tools that employ graph-based analysis and behavioral modeling, although offering a holistic view of attack paths, are resource-intensive and insufficiently integrated with testing frameworks, limiting their practicality for dynamic 5G scenarios.

Thus, the development of automated penetration testing methods for "5G and beyond" network slices that combine real-time automation, scalable designs, and comprehensive threat modeling is critical for ensuring the security and resilience of next-generation communication networks.

Since the field of penetration testing is not a new field at all, there are already many well-tested methods (OSSTMM (Open Source Security Testing Methodology Manual); ISSAF (Information Systems Security Assessment Framework); PTES (Penetration Testing Execution Standard); NIST SP800-115 (National Institute of Standards and Technology Special Publication 800-115); OWASP (Open Web Application Security Project), etc.)) that have proven themselves well when used for different types of networks and systems of different scales and purposes. However, cellular communication systems have specific characteristics, especially given the wide variety of technologies that they use.

The penetration testing of 5G cellular networks has several features that distinguish it from previous generations (2G, 3G, 4G). This is due to the new architectural and technological solutions implemented in 5G, as well as to a greater degree of network interaction with other technologies, such as IoT (Internet of Things), cloud services, and low-latency networks.

Compared to previous generations, 5G places a much greater emphasis on distributed computing resources and network functions. During penetration testing, it is crucial to assess all possible entry points, including cloud computing platforms, decentralized data processing nodes, and edge components.

Because network function virtualization is a key aspect of 5G, penetration testing must also consider potential vulnerabilities in virtualization and containerization systems.

Because 5G offers much lower latency for critical applications (for example, in the field of medicine or autonomous transport), it is important to check whether attacks on the network or individual nodes can affect these parameters.

The growing number of IoT devices connected to 5G requires testing the protection capabilities against DDoS attacks, data interception, and other threats that can arise from interactions between different types of devices.

5G networks use higher frequencies (up to 100 GHz), which opens up new attack vectors. It is necessary

to test whether communication mechanisms are protected from the interception of signals at these frequencies.

5G uses beamforming technology for directional signal transmission. Testing should include checking for possible attacks, such as spoofing and beam blocking.

Due to the use of virtualization and distributed architecture, it is necessary to conduct testing for the presence of possible vulnerabilities between different network segments, including the network core (Core Network) and Radio Access Network (RAN).

Many IoT devices will have access to 5G networks. Most of these devices may have weak built-in security mechanisms; thus, penetration testing should include checking the protection of these devices against compromise and possible attacks through the IoT segment.

5G provides smoother transitions between different networks (Wi-Fi, 4G, 5G). Testing should include attacks related to roaming, as well as interception or modification of data when passing between different networks.

Because mobile devices can move between different base stations, testing must consider the possibility of attacks at these stations, particularly in remote or unprotected areas.

Thus, a multi-criteria analysis of these methods was performed to determine their potential use specifically for cellular communication networks (Table 1).

The comparison of penetration testing methodologies reveals their strengths and limitations when applied to 5G cellular networks. Each methodology has a unique focus, with OSSTMM and PTES providing comprehensive frameworks that can be easily adapted to the specific challenges caused by 5G, such as virtualization and the integration of IoT devices.

ISSAF offers important insights for organizational risk assessments, which are obligatory to understand the broader context of 5G security. At this time, NIST SP800-115 emphasizes risk management, an important aspect given the critical nature of 5G applications.

OWASP is useful when web components are assessed. These components are integral to 5G services; however, they are not sufficient for addressing the unique vulnerabilities introduced by new technologies and higher frequencies.

3. Objectives and the Approach

This study aims to develop a methodology and software to ensure reliable and automated penetration testing of network infrastructures in 5G and beyond networks with layered architecture. The primary goal is to address the critical cybersecurity challenges posed by the dynamic and complex structure of modern cellular networks. Specifically, this includes identifying vulnerabilities, optimizing attack strategies, and enhancing security measures. It is essential to design and implement a method that leverages Genetic Algorithms (GAs) to automate and optimize penetration testing to realize efficient detection of vulnerabilities and simulation of attack scenarios within diverse network slices. During the development process, limitations and assumptions regarding the virtualization and distributed architecture of 5G networks must be considered. The methodology should also be tested and validated through

Table 1

Comparison of penetration testing methodologies

Methodology	Description	Strengths for 5G	Limitations for 5G
OSSTMM (Open Source Security Testing Methodology Manual)	Comprehensive framework for security testing.	Thorough analysis of all attack vectors.	Does not provide explicit guidance on advanced 5G-specific features (e.g., network slicing, virtualization, etc.).
ISSAF (Information Systems Security Assessment Framework)	Framework for security assessments.	Useful for assessing organizational risks in 5G.	Focuses more on traditional systems and limited applicability to dynamic and distributed 5G architectures.
PTES (Penetration Testing Execution Standard)	Standardized approach to penetration testing.	Clear guidelines can be adapted for 5G environments.	Lacks coverage of high-frequency vulnerabilities and IoT-specific threats common in 5G networks.
NIST SP800-115	Guidelines for conducting penetration testing.	Strong focus on risk management applicable to 5G.	Its web-centric scope does not encompass network-level threats, higher frequencies, or the specialized attack surfaces arising from 5G's distributed and virtualized infrastructure.
OWASP	Focused on web application security.	Valuable for testing web components of 5G networks.	Its web-centric scope does not encompass network-level threats, higher frequencies, or the specialized attack surfaces arising from 5G's distributed and virtualized infrastructure.

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

experiments based on the efficiency, accuracy, and practicality of GA optimization to identify attack strategies. Furthermore, the proposed framework should be enhanced and analyzed to identify practical advantages and limitations in terms of ensuring the cybersecurity of next-generation networks.

The main objectives and stages of this research are as follows:

– *stage 1*: defining the problem of penetration testing in 5G and beyond networks, identifying unique challenges, such as virtualization, network slicing, and distributed architecture, and analyzing existing methodologies for limitations and opportunities (Sections 2, 4);

– *stage 2*: developing a novel method for automated penetration testing by integrating Genetic Algorithms (Section 4);

– *stage 3*: designing and implementing the method, considering technologies, programming languages, approach to data collection, building the attack representation, and implementing GA (Section 5);

– *stage 4*: validating the methodology's effectiveness through controlled experiments, analyzing key performance metrics, and discussing practical recommendations for improving its application in real-world scenarios (Sections 6, 7);

– *stage 5*: concluding the research by summarizing the results, identifying key findings, and outlining further directions for enhancing automated penetration testing in 5G and beyond networks (Section 8).

4. Analysis

4.1. Suggested methodology

To address the security challenges caused by 5G networks, we propose an automated penetration testing framework that uses Genetic Algorithms (GAs) to optimize attack strategies. This framework aligns with described methodologies such as OSSTMM, ISSAF, PTES, NIST SP800-115, and OWASP, and offers enhancements required due to the complexities of modern network environments.

The proposed approach involves systematic data collection, gathering information on network devices using APIs and tools like Shodan. This process aligns with OSSTMM and NIST SP800-115 principles and emphasizes the importance of having a good understanding of network topology and vulnerabilities. The core of the proposed framework employs a Genetic Algorithm module that constructs an attack representation through an attack tree. Then, this representation is processed by the GA to check and optimize different attack strategies using a defined fitness function, which is built on the structured methodologies of PTES and ISSAF.

Then, an integration layer improves the offered framework through various penetration testing tools, supports OWASP's focus on web application security, and improves its excessively broader 5G context.

Using leveraging GAs, the proposed framework accelerates the penetration testing process and improves the identification of effective attack vectors by addressing specific vulnerabilities related to 5G technologies. The proposed method builds on existing standards and adapts them to the rapidly evolving cybersecurity challenges.

Thus, the proposed methodology can form the basis of automated testing for the penetration of cellular communication networks. The generalized methodology of this testing is illustrated in Figure 3.

4.2. Data Collection and Network Analysis

The proposed methodology's main stages are data gathering, GA model usage, and usage of advanced penetration testing tools. These stages are considered in more detail below.

The first step in obtaining data for the developed GA model is to use Shodan. This search-engine tool for Internet-connected devices gathers information about publicly accessible devices and services in the network. It can identify exposed endpoints, vulnerabilities (e.g., CVE), and devices relevant to the network slices. Thus, we can extract detailed information about the network slice topology, which allows us to model a realistic network environment for penetration testing.

The mulVAL (Multihost, multistage Vulnerability Analysis) framework is an open-source tool for analyzing and generating potential attack paths (trees) within a network corresponding to its topology [16]. Thus, it can simulate adversarial behavior to identify security vulnerabilities in the topology and generate a simplified attack graph to visualize possible exploit chains across multiple network components.

By using a simplified matrix, it is possible to process collected data in a suitable form for use in the GA model. The received matrix was used as input data for the GA model. The matrix also contains additional data about the actions required to achieve the goal (e.g., executing a command, accessing a file, etc.). This simplified format allows Genetic Algorithms to analyze input data efficiently.

4.3 GAs and Attack Strategy Optimization

The GAs analyze all possible attack paths and determine the optimal path that leads to achieving the goal (for example, access to critical data or resources) the fastest. The model considers the risks, efforts to execute attacks, and possible outcomes of each step.

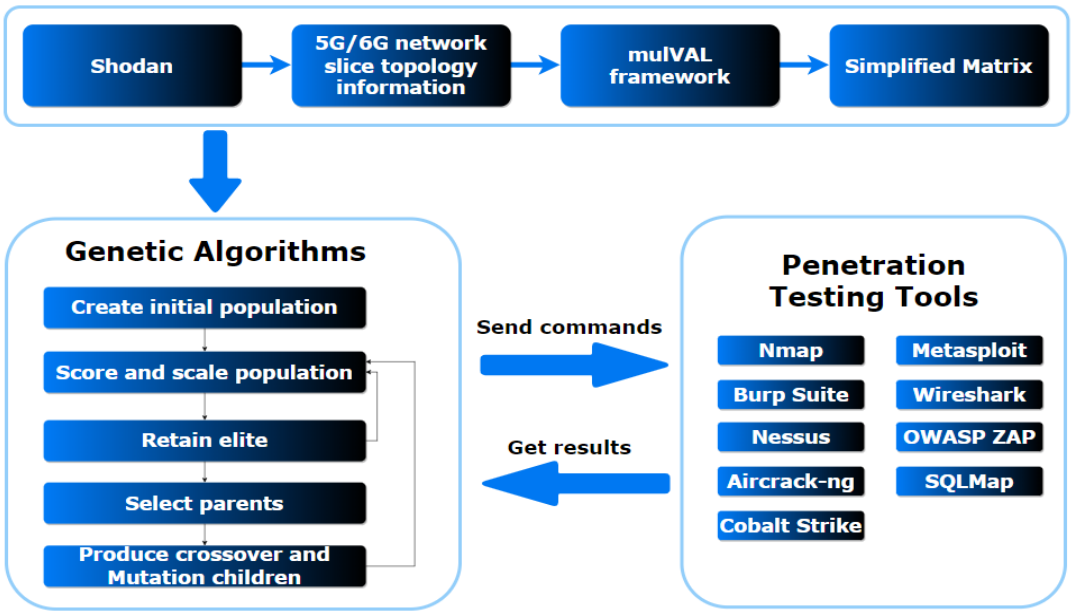

Fig. 3. Generalized 5G and beyond network slices penetration testing methodology

After analysis, the model generates commands to execute attacks using testing tools.

The use of GAs allows the method to dynamically optimize attack strategies. This optimization process balances multiple objectives, including success probability, impact, and resource cost. In addition, the GA model provides adaptability and scalability to the framework. It processes extensive datasets and adjusts to evolving network configurations, ensuring relevance even as the network topology changes. This capability is essential for handling the distributed and layered architectures characteristic of modern networks. By iteratively refining attack strategies based on real-time feedback, the proposed model enhances the accuracy of vulnerability detection and the efficiency of resource utilization. Without such advanced algorithms, the testing framework would struggle to meet the demands of next-generation networks, particularly in terms of speed, scalability, and the ability to uncover innovative attack vectors.

4.4. Integration with Penetration Testing Tools

The proposed methodology is integrated with existing penetration testing tools, such as Wireshark, Metasploit, Nessus, Netcat, and Hydra. In addition, it includes command execution, vulnerability exploiting, and security assessments.

The methodology is focused on creating a software interface that allows the GA model to send commands to those tools; after that, collected feedback is analyzed by the system. This allows decision-making on how to proceed with further steps based on the obtained data about the effectiveness of the attacks. The simplified process is illustrated in Fig. 4.

This diagram illustrates the relationship between the key components of cybersecurity, including attackers, organizations, and risk assessment tools. An attacker attacks an Enterprise network, which triggers a Risk Assessment.

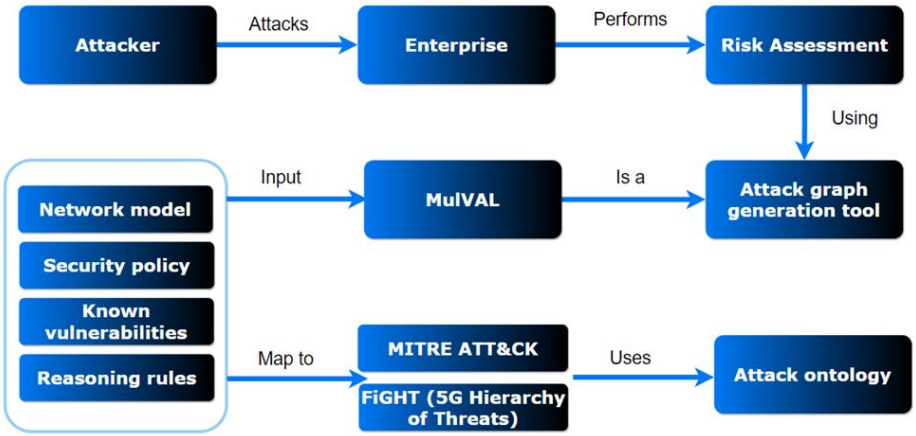

Fig. 4 Flow [17]

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

In this process, the MulVAL tool is used to generate Attack Graphs. MulVAL takes inputs from network models, security policies, known vulnerabilities, and interaction rules. The collected data is also mapped to the MITRE ATT&CK, a knowledge base of attack tactics, techniques, and procedures (TTPs). This allows us to create a full-fledged Attack Ontology for further analysis [17]. In the context of threats to 5G technologies, this structure is complemented by the FiGHT (5G Hierarchy of Threats) concept, which is integrated into the overall analysis to identify specific risks in new network environments.

This methodology also suggests the use of the following tools that are widely used for network penetration testing (Tabl. 2).

4.4. Key Performance Indicators (KPIs) for Algorithm Selection

The most appropriate AI model was the Genetic Algorithm. This was achieved using the next approach.

The KPIs for comparing algorithms in the context of 5G penetration testing were chosen to address the specific challenges and demands of this complex and evolving environment (Tabl. 3).

– *Complexity Management* ensures that the algorithm can handle the intricate nature of 5G networks. Include diverse services, interconnected devices, and virtualize infrastructures.

– *Scalability* evaluates the algorithm's ability to remain effective as network size and complexity grow, which is a critical factor as 5G networks continue to expand.

– *Adaptability* ensures that the algorithm can adjust to the ever-changing cybersecurity landscape, including emerging threats and vulnerabilities.

– *Multi-Objective Balancing* measures how well the algorithm can optimize competing goals, such as maximizing success rates while minimizing resource use and detection risks.

– *The Integration Potential* considers the algorithm's compatibility with other AI techniques and tools, thereby enabling enhanced analysis and a more robust security framework.

– *Robustness assesses* the algorithm's ability to discover innovative and unconventional attack strategies, which are critical for overcoming advanced security measures.

– *The Convergence Speed* focuses on the time or number of iterations required to reach a solution, ensuring practicality in real-world, time-sensitive scenarios.

– *Solution Quality* ensures that the algorithm delivers high-quality, actionable results that meet real-world requirements and effectively enhance network security.

Table 2

Penetration testing tools

Penetration Testing Tool	Description
Metasploit	A big framework for developing and executing exploits, facilitating the testing of vulnerabilities identified during data collection. Tools support the optimization of attack strategies using integration with the GA module.
Nmap	A powerful network scanning tool that helps to map network topology and find services, needed to understand potential attack surfaces in 5G networks.
Burp Suite	An integrated platform for web application security testing, very useful for identifying vulnerabilities in web components, aligning with OWASP principles.
Wireshark	A network protocol analyzer that captures packet data, is useful for analyzing network traffic, to identify anomalies and inform the attack representation.
Nessus	A vulnerability scanner that identifies known vulnerabilities, supports the automated vulnerability assessment process in the framework.
OWASP ZAP	An open-source scanner focused on web application security, it helps to find vulnerabilities during the testing phase, aligning with OWASP's best practices.
Aircrack-ng	A suite for assessing Wi-Fi network security, relevant for penetration testing in environments where wireless communication integrates with 5G.
SQLMap	An automated tool for detecting and exploiting SQL injection vulnerabilities, it is important for testing web applications, which were identified during the data collection phase.
Cobalt Strike	A tool for advanced threat emulation, it supports the GA module by simulating real-world attack scenarios and refines attack strategies based on fitness evaluations.

Table 3

Comparison of KPIs

KPI	GA	DE	PSO	ES	ABC	Tabu search
Complexity Management	Excellent	Moderate	Moderate	High	Moderate	Moderate
Scalability	High	High	High	High	Moderate	Moderate
Adaptability	High	Moderate	Moderate	High	Moderate	Low
Multi-Objective Balancing	High	Moderate	Moderate	Moderate	Moderate	Low
Integration Potential	High	Moderate	Moderate	Moderate	Moderate	Low
Robustness	High	High	Moderate	High	Moderate	High
Convergence Speed	Moderate	Fast	Fast	Moderate	Moderate	Moderate
Solution Quality	High	High	High	High	High	High

Explanation of Ratings:

– *High:* the algorithm excels in this KPI;

– *Moderate:* the algorithm performs adequately but may have limitations;

– *Low:* the algorithm underperforms compared to others in this KPI;

– *Very High:* the proposed algorithm significantly outperforms the compared algorithms.

In the context of 5G network slice penetration testing context, where complexity, adaptability, and robustness are paramount, Genetic Algorithms stand out as the optimal choice. They offer a superior balance of critical capabilities that address the challenges posed by the complex and rapidly evolving 5G environment. By excelling at the most crucial KPIs, GAs provide a comprehensive and effective solution to enhance network security via advanced penetration testing strategies.

The choice of employing Genetic Algorithms for our penetration testing framework, particularly in the 5G network context, offers several significant advantages:

1. *Complexity Management.* 5G networks are characterized by their complexity and feature numerous interconnected devices and diverse services. GAs excel in navigating this complexity by exploring a vast solution space, which helps identify effective attack strategies that may be overlooked using more deterministic methods.

2. *Adaptability.* The cybersecurity landscape is evolving continuously, and new vulnerabilities and attack vectors are emerging frequently. GAs are inherently adaptive; they can evolve strategies based on performance feedback, making them well-suited for dynamic environments like 5G networks.

3. *Multi-Objective Optimization.* Penetration testing often requires balancing multiple objectives, such as maximizing the chances of success while minimizing resource use and risk of detection. GAs are adept at optimizing multiple objectives simultaneously; thus, they are ideal for this task.

4. *Robustness.* GAs maintain genetic diversity within the population of strategies, thereby preventing premature convergence on suboptimal solutions. This is particularly important in 5G networks, where innovative and novel attack strategies are required to bypass advanced security measures.

5. *Integration Potential.* GAs can be effectively combined with other AI techniques, such as machine learning models. This integration enhances the overall effectiveness of the penetration testing framework, which allows for more sophisticated analysis and strategy development.

6. *Scalability.* As 5G networks grow in both scale and complexity, GAs can handle larger populations of potential solutions efficiently. This scalability ensures that the framework remains effective even when the attack surface expands.

5. Design and Implementation

This section presents the design of our novel automated penetration testing framework, which uses Genetic Algorithms (GAs) to optimize attack strategies in complex network environments, particularly in 5G networks. The proposed framework comprises three main components: the data collection module, the Genetic Algorithm module, and an integration layer for various penetration testing tools. To implement the framework we used Python language with the required libraries.

The data collection module is designed to gather the required information about the network environment, including device details, vulnerabilities, and network topology. The proposed method uses various APIs and scanning tools to collect relevant data.

The architecture of this module includes a backend service that interfaces with data sources, in our case we used Shodan, the National Vulnerability Database (NVD), and the Common Vulnerabilities and Exposures (CVE) system. In addition, we have a frontend dashboard for visualizing the collected data, and a database manages and retrieves the information efficiently.

The technologies used in this module include the following Python libraries: Requests (for API calls), BeautifulSoup (for web scraping), SQLAlchemy (for

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

database interaction), and Pandas (for data analysis).

The Genetic Algorithm module implements the GA to optimize attack strategies by evaluating potential attacks using a fitness function. This function considers parameters such as effectiveness, impact, and cost.

The architecture of this module includes components for initialization, fitness evaluation, selection, crossover, mutation, and termination criteria. The GA iterates over several generations, evolves attack strategies, and reports the results for further analysis.

The technologies used in this module include the following Python libraries: NumPy (for numerical operations), DEAP (for evolutionary algorithms), and Matplotlib (for visualizing results).

The integration layer connects the GA module to different penetration testing tools, and it facilitates the execution of optimized attack strategies. The layer manages tool configurations and result reporting.

The architecture of the integration layer is used as middleware, enabling communication between the GA module and the tools. It handles API requests and responses, tool execution, and result aggregation.

The technologies used in this module include the following Python libraries: Subprocess (for executing shell commands), Flask (for creating a lightweight API), and JSON (for data interchange).

5.1 Steps Involved in Implementing the Automated Penetration Testing Method

The implementation of the automated penetration testing framework follows these main steps:

1. *Setup Development Environment.* Install Python and the required libraries using pip and set up a version control system to manage code.

2. *Data Collection.* Implement API calls to gather data from network devices (e.g., using Shodan) and store data in a database. Automated vulnerability assessments are performed by scanning tools and adding them to the vulnerability database.

3. *The Genetic Algorithm Module is constructed.* The fitness function is defined, and efficiency, impact, and cost parameters. Create functions to initialize, select, crossover, and mutation of possible solutions. Implement a loop to evolve strategies until termination criteria are met.

4. *Integrate Penetration Testing Tools.* Select tools for integration (e.g., Metasploit, Nmap) and implement wrappers for their APIs. Create functions to execute these tools using the parameters obtained from the optimized attack strategies and gather the results for analysis.

5. *Testing and Validation.* Conduct unit tests on individual components (data collection, GA module, integration layer) and perform integration tests to ensure smooth communication between components. Validate the framework by executing penetration tests in the test environment.

6. *Documentation and Reporting.* Prepare user documentation with detailed setup, configuration, and usage of the proposed framework. Develop a reporting mechanism to present findings and recommendations from penetration tests.

7. *Deployment.* Deploy the application in a secure environment and ensure compliance with legal and ethical guidelines when conducting penetration testing.

5.2. Technologies and Programming Languages

The programming language used in the development of the proposed framework is Python, which we selected because of its simplicity and extensive libraries that are required for network interactions, data processing, and machine learning. The proposed framework uses different libraries to enhance functionality, including the following:

– Django for web API development;

– NumPy and Pandas for data manipulation and analysis;

– DEAP for implementing genetic algorithms;

– Matplotlib to visualize results.

Based on the research conducted in the previous section, a scheme for implementing the proposed methodology into the existing architecture of 5G cellular networks was proposed (Fig. 5).

The general infrastructure includes the following key components: Cloud RAN (Radio Access Network), 5G Core, resource orchestrator, and automated penetration testing tool. The architecture provides for the full integration of automated processes to assess the security of network slices, which are virtualize network segments adapted to the specific needs of users or services.

Cloud RAN with base stations and 5G Core are the main components of 5G networks. The orchestrator acts as a control unit that manages and assigns network resources dynamically and plays a pivotal role in maintaining the Quality of Service (QoS) across various slices. Network slices (Slice 1, Slice 2…, Slice N) are logical partitions of the 5G network, and they are customized for specific use cases or customers. Each slice represents a virtualize segment tailored to the different service requirements. The penetration testing tool is located in a virtual machine (VM) and connected to the cloud infrastructure. This component performs automated attacks on network segments to identify vulnerabilities. It uses AI/ML models to analyze data and determine the best attack paths.

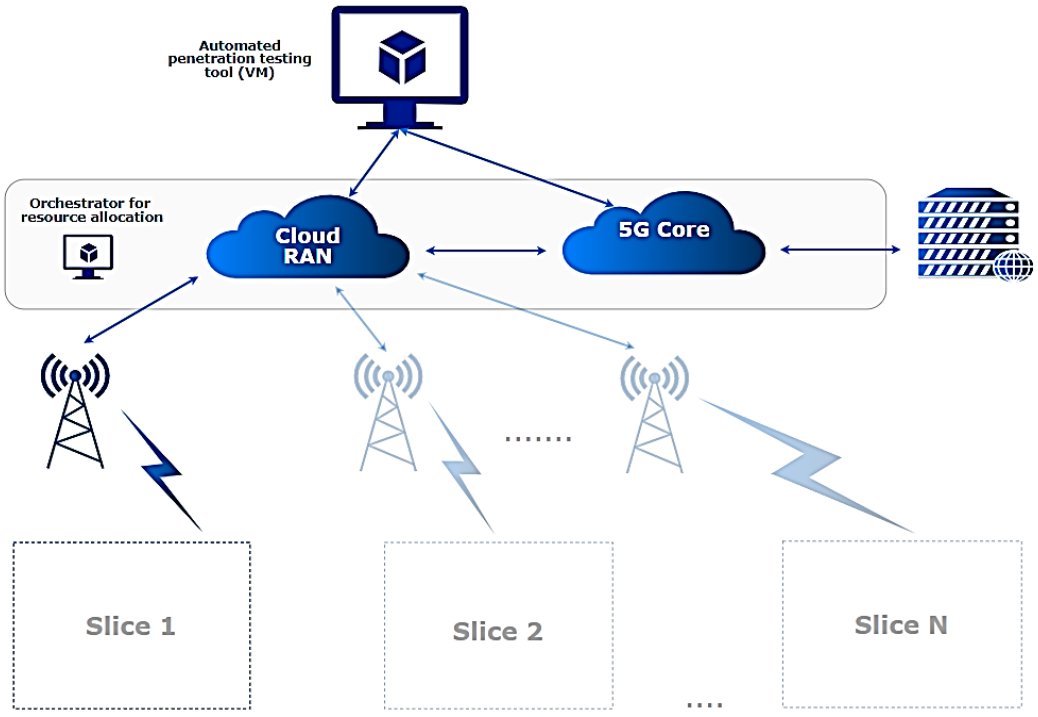

Fig. 5 System design

5.3. Automated Penetration Testing Framework Using Genetic Algorithms

The proposed framework comprises three important components: data collection, the Genetic Algorithm module, and an integration layer for various penetration testing tools.

Genetic Algorithms (GAs) are optimization techniques inspired by natural selection. They solve complex problems by evolving a population of potential solutions over several generations.

Genetic algorithms can be implemented as follows:

1) *Initialization:* the algorithm begins with a randomly generated population of candidate solutions.

2) *Fitness Evaluation:* each candidate solution is assessed using a fitness function that measures how well it solves the problem. The fitness function can be expressed as follows:

– $F(x) = E(x) - P(x)$, (1)

– $F(x)$ is the fitness score of solution x,

– $E(x)$ is the effectiveness or benefit of solution x,

– $P(x)$ is the cost or penalty associated with solution x.

3) *Selection:* solutions are selected for reproduction based on their fitness scores. The selection probability of solution s is calculated as follows:

– $P(s) = F(s)/\sum F(i)$, (2)

– $P(s)$ is the probability of selecting solution s,

– $F(s)$ is the fitness score of solution s,

– $\sum F(i)$ is the total fitness score of all solutions in the population.

4) *Crossover:* selected solutions combine parts of two parent solutions to create new offspring, allowing for the sharing of successful traits.

5) *Mutation:* random changes are introduced to some offspring to maintain population diversity. This prevents the algorithm from getting stuck in local optima.

6) *Replacement:* the new generation of solutions replaces some or all of the previous population. This cycle repeats until specific termination criteria are met, such as reaching a desired fitness level or a set number of generations.

GAs are widely used in various fields because they can effectively navigate complex search spaces. Some common applications of the proposed method include the following:

– Engineering Design: designing structures or mechanical components;

– Scheduling: solving scheduling problems in industries such as transportation and manufacturing;

– Machine Learning: tuning hyperparameters for better model performance;

– Financial Modeling: assisting in portfolio optimization and risk management.

– Game Development: creating adaptive behaviors for non-player characters (NPCs).

Genetic Algorithms are a powerful approach to solving complex optimization problems across many domains. Their adaptability and ability to evolve solutions make them valuable tools in both theory and practice.

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

5.4. Data Collection

The efficacy of any automated testing framework is strongly contingent on the quality and relevance of the data it uses. Our data collection process is systematic and multi-faceted, focusing on both network topology and vulnerabilities.

Initially, we gathered comprehensive information about network devices using various APIs and network scanning tools. The primary source of this data is Shodan, which provides access to a wealth of information about internet-connected devices. For example, a typical JSON output from Shodan to a web server might look like this:

```
json
Copy code
{
     "ip_str": "203.0.113.5",
     "port": 443,
     "hostnames": ["example.com"],
     "org": "Example Organization",
     "data": "HTTP/1.1 200 OK",
     "vulns": ["CVE-2021-34527"],
     "location": {
        "country_name":"United States",
        "city": "San Francisco",
        "latitude": 37.7749,
        "longitude": -122.4194
      },
     "product": "nginx",
     "version": "1.19.10",
     "transport": "tcp"
}
```

In this output:

– **ip_str:** the public IP address of the device;

– **port:** the port on which the service is running;

– **hostnames:** domain names associated with the IP address;

– **org:** the organization hosting the service;

– **data:** initial HTTP response data from the server;

– **vulns:** known vulnerabilities identified by CVE IDs;

– **location:** geographic location details of the device;

– **product:** the identified software product;

– **version:** version of the software running.

This comprehensive dataset allows accurate modeling of the network environment, thereby providing critical insights into potential attack surfaces. Each unique service running on a device contributes to the construction of the attack representation, including its associated vulnerabilities.

After collecting the host data, we perform automated vulnerability assessments using established tools. These tools scan the identified devices for known vulnerabilities and map them against recognized databases like the National Vulnerability Database (NVD) and the Common Vulnerabilities and Exposures (CVE) system. The result is a comprehensive vulnerability dataset that catalogs known exploits, their severity scores, and various other metrics.

– After collecting the host data, we perform automated vulnerability assessments using established tools. These tools scan the identified devices for well-known vulnerabilities by mapping them against recognized databases like the National Vulnerability Database (NVD) and the Common Vulnerabilities and Exposures (CVE) system. As a result, the outcome is a working vulnerability dataset containing known exploits, severity scores, and various other metrics.

5.5. Building the Attack Representation

Once the necessary data are collected, the next step involves constructing an attack representation. This is achieved by formulating an attack tree that visually maps out potential attack vectors. Each node in the tree represents a state or a specific action that can be taken against the network, and the edges indicate transitions or possible pathways between these states.

To facilitate the Genetic Algorithm, this attack tree must be converted into a structured format. We develop a matrix that captures all possible transitions between attack states. This matrix not only serves as a foundation for GA processing and integrates key performance metrics, such as the likelihood of success and potential impact of each attack pathway.

The fitness function is the core of the Genetic Algorithm, and it evaluates the performance of various attack strategies. We define the fitness function as follows:

$$F(x) = w_1 * S(x) + w_2 * I(x) - w_3 * C(x). \quad (3)$$

In this equation:

– $F(x)$ is the fitness score of strategy x;

– $S(x)$ is the success probability of executing the attack;

– $I(x)$ is the impact score of the attack;

– $C(x)$ is the cost associated with the attack;

– $w_1, w_2,$ and w_3 are weights used to balance the factors.

The effective design of the fitness function is important because it directly influences the GA's ability to identify optimal attack strategies. Adjusting the weights allows fine-tuning based on specific objectives, such as prioritizing stealth over impact on certain scenarios.

5.6. Implementing the Genetic Algorithm

The implementation of the Genetic Algorithm involves a series of continuous steps:

1. Population initialization: this process begins by

generating an initial population of attack strategies, which are sampled randomly from the attack tree. This diverse initial set ensures a broad exploration of potential solutions.

2. Fitness evaluation: each strategy was evaluated using a predefined fitness function. This step quantifies the performance of each attack strategy based on its likelihood of success, potential impact, and associated costs.

3. Selection: the best performing strategies are selected for reproduction. This can be achieved using methods such as tournament selection or roulette wheel selection, where strategies with higher fitness scores have a greater chance of being selected. The selection probability can then be calculated as follows:

$$P(s) = F(s) / \sum F(i) \text{ for } i = 1 \text{ to } N, \qquad (4)$$

where $P(s)$ is the probability of selecting strategy s; N is the total number of strategies in the population.

4. Crossover and Mutation: new strategies are generated by combining elements of the selected strategies through crossover, where parts of two or more strategies are merged. Mutation introduces random changes to some strategies, thus ensuring genetic diversity and preventing convergence on suboptimal solutions.

5. Termination Criteria: The GA process continues until specific termination criteria are met, such as reaching a predetermined number of generations or achieving a fitness score that meets or exceeds a specified threshold. This allows selection of the most effective attack strategies identified during evolutionary processes.

6. Result Interpretation and Reporting: After convergence of the algorithm, the final set of strategies is analyzed. This analysis includes assessing the potential effectiveness of each strategy in a real-world testing environment and preparing a report detailing the findings, recommendations, and potential mitigation.

The Genetic Algorithm identifies the most efficient and effective attack pathways for penetration testing in the complex 5G network landscape. Using leveraging GAs, our approach not only increases the speed and efficacy of penetration testing and adapts to the vectors of modern cybersecurity challenges.

6. Testing and Validation

6.1. Testing Environment Overview

Testing of the proposed software occurred on the network of the National Aviation University, which was built following open specifications (Fig. 6).

To deploy a 5G test network based on open-source solutions, an end-to-end project from OpenAirInterface5G is used: OAI CN 5G (network core) and OAI gNB (base station, RAN), which is a 5G SA (Stand-Alone) network. The OAI provides projects with all standardized 3GPP functions. The 5G CN was deployed on the virtual machine, and the gNB was deployed on PC Intel Core, both of which were operating on Ubuntu Server OS. The Ettus USRP B210 software-defined radio was used for radio access.

All network components are allocated to the same subnet and are connected to the same virtual switch.

To test the developed methodology and the corresponding software, a test network was used (Fig. 7). A virtual machine with the developed software was installed on the network. After establishing the network, the software was launched to scan the network status. The real-time scan results are recorded in the connected database.

6.2. Testing and Results Validation

Testing the proposed automated penetration testing framework was conducted in a controlled environment on a 5G network using open-source solutions from OpenAirInterface5G. The main objective of the testing goal was to validate the efficiency, accuracy, and practicality of the Genetic Algorithm (GA) optimization in terms of identifying attack strategies.

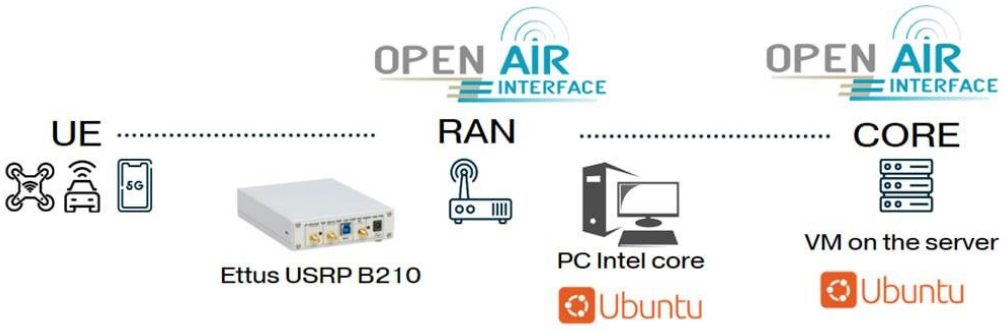

Fig. 6 Testing environment

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

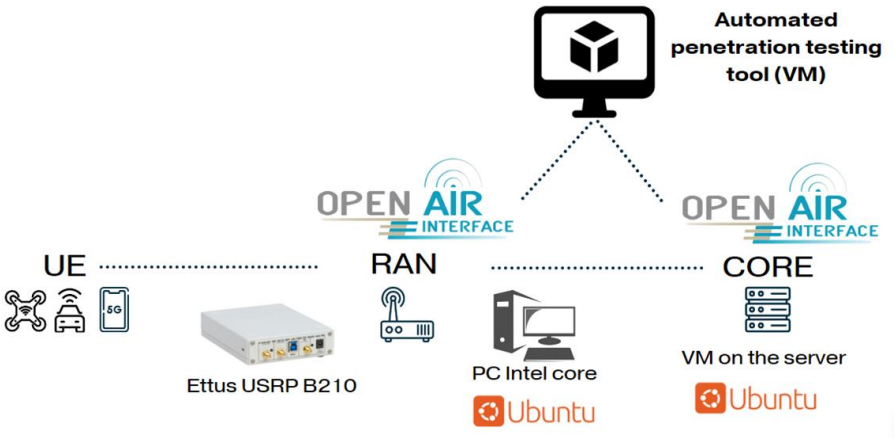

Fig. 7 Deployment of testing environment

During the initial network scan, the framework detected 25 vulnerabilities across 15 different network devices. These vulnerabilities were matched against the CVE database, and the following results:

– Critical vulnerabilities: 5,

– High-severity vulnerabilities: 10,

– Medium-severity vulnerabilities: 7,

– Low-severity vulnerabilities: 3.

The total detection time for scanning 15 devices was 12 min, which demonstrates the framework's efficiency in gathering and processing vulnerability data in a real-world scenario.

The GA successfully generated and optimized a set of 5 attack strategies over 50 generations. For each strategy, the fitness function was calculated, factoring in:

– Success Probability (S(x)): ranged from 70% to 95%;

– Impact Score (I(x)): ranged from 50 to 100, indicating the potential damage from the attack;

– Cost (C(x)): ranged from 10 to 30, measured in resource cost, time, and effort to launch the attack.

The strategy with the highest fitness score was represented as follows:

– Success Probability (S): 92%,

– Impact (I): 85,

– Cost (C): 20,

– Fitness Score (F): 0.75, optimized for a balance between impact and cost.

When simulating attacks in the test environment, the success rate of the GA-generated attack strategies was measured as follows:

– *Strategy 1 (Highest Fitness):* 92% success rate in compromising the targeted base station.

– *Strategy 2:* 85% success rate in exploiting vulnerability in network router

– *Strategy 3:* 78% success rate in breaching the firewall of a network device.

The success rate varied depending on the complexity of the network environment and the attack vector. The impact and cost analysis of the attack strategies demonstrated that the GA successfully balanced the trade-off between attack effectiveness and resource utilization:

– *Strategy 1 (Highest Fitness):* high impact with moderate cost,

– *Strategy 2:* medium impact with low cost, optimized for stealth.

– *Strategy 3:* high cost but very high impact, designed for disruptive attacks.

The weighting factors w_1, w_2, and w_3 were adjusted to prioritize stealth (lower cost) or high impact based on the test scenario.

Table 4

Test Results Summary

Test Case	Result
Vulnerability Detection Efficiency	25 vulnerabilities were detected across 15 devices. Critical vulnerabilities identified: 5.
Attack Strategy Optimization	5 optimized attack strategies were generated. The best strategy had a 92% success rate with an impact score of 85.
Success Rate of Identified Strategies	Strategy 1: 92%, Strategy 2: 85%, Strategy 3: 78%.
Impact and Cost Analysis	GA successfully balanced cost and impact, with strategies ranging from stealth to high impact.

7. Discussion

The main results obtained during testing the proposed methodology are summarized as follows:

1. The proposed framework identified 25 vulnerabilities across 15 devices in 12 minutes, including 5 critical and 10 high-severity vulnerabilities.
2. The Genetic Algorithm optimized 5 attack strategies over 50 generations, and the best strategy achieved a 92% success rate and fitness score of 0.75.
3. The success rates of the top three strategies were: 92% for compromising the base station, 85% for exploiting a router vulnerability, and 78% for breaching a firewall.
4. The GA effectively balanced the attack impact and resource cost, optimizing strategies for high impact or stealth based on the scenario.

However, there are also certain limitations in the application of the developed methodology:

1) Complexity of Real-World Networks.

Although the framework performs well in controlled environments, its effectiveness may decrease in more complex, dynamic, or highly secure real-world networks with unpredictable configurations and defenses.

2) Dependence on Vulnerability Databases.

The accuracy and completeness of vulnerability detection are reliant on the quality and up-to-date status of external vulnerability databases (e.g., CVE, NVD), which may not capture all emerging or zero-day vulnerabilities.

3) Computational Resource Requirements.

The optimization process of a Genetic Algorithm can be resource-intensive, especially when dealing with large, complex networks, potentially requiring significant computational power and time for extensive scanning and analysis.

4) Adaptability to New Attack Vectors.

Although the framework is designed to optimize known attack strategies, it may struggle to adapt quickly to entirely new, unknown attack vectors or advanced persistent threats (APTs) that have not been modeled in the system.

5) False Positives and False Negatives.

As with any automated penetration testing tool, there may be instances of false positives (incorrectly identifying vulnerabilities) or false negatives (failing to identify certain vulnerabilities), particularly in environments with frequent changes or custom configurations.

It can be seen that while the developed methodology demonstrates promising results, it has certain limitations. The framework's effectiveness may decrease in highly complex and dynamic real-world networks, where unpredictable configurations and advanced defenses can occur. In addition, the accuracy of vulnerability detection relies heavily on the completeness and timeliness of external vulnerability databases, which may not capture all emerging or zero-day vulnerabilities. The computational demands of the Genetic Algorithm's optimization process can also be resource-intensive, potentially requiring significant time and power for larger networks. In addition, the system may face challenges in adapting to entirely new attack vectors or advanced persistent threats (APTs) that have not been modeled. Finally, as with many automated penetration testing tools, there may be false positives and false negatives, particularly in networks with custom configurations. Therefore, further scientific research is planned to enhance the framework's adaptability to complex, real-world networks by integrating machine learning to make dynamic adjustments to attack strategies. In addition, efforts will be made to incorporate more real-time and comprehensive vulnerability databases to improve detection accuracy. The optimization of computational efficiency will also be a priority, focusing on reducing resource consumption through parallel processing or more efficient algorithms. Expanding the range of modeled attack vectors, including APTs and insider threats, strengthens the framework's ability to address unknown attack scenarios. Finally, refining the scanning algorithms to minimize false positives and negatives will further improve the reliability of the system.

8. Conclusion

Cellular communication networks are evolving toward the sixth generation, which will retain the advantages of previous generations while significantly enhancing the subscriber capabilities, including reduced latency, increased bandwidth, and higher connection density. These improvements are driven by advancements in radio interfaces, network cores, transport channels, and other infrastructure components. However, it is crucial to note that future cellular networks will continue to rely on layered structures and virtual resource sharing.

Despite remarkable advancements in cellular networks, ensuring high-level cybersecurity remains a critical challenge. To assess the mechanisms and level of protection in these networks, penetration testing techniques are invaluable. Therefore, the purpose of this study was to develop a methodology for penetration testing within the network infrastructure of next-generation cellular networks, particularly those with a layered architecture. This methodology leverages best practices from existing penetration testing methods and tools, combined with artificial intelligence algorithms, to orchestrate the process and optimize decision-making during testing. Among the various algorithms analyzed, the Genetic Algorithm was selected as the most suitable, considering its application specifics.

The proposed methodology was implemented as software installed on a dedicated virtual machine

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

connected to the network. To test the developed methodology and associated software, a test network architecture was created using open-source solutions, and a methodology for conducting experimental research was devised accordingly. The results prove the operability and effectiveness of the proposed solutions, demonstrating improved vulnerability detection, optimized attack strategy generation, and a higher penetration test success rate in a complex network environment.

Thus, the main areas of scientific research were formulated, which are aimed at: enhancing the adaptability of penetration testing methodologies for next-generation networks, further integrating artificial intelligence for real-time decision-making, developing more efficient algorithms for resource optimization during testing, and expanding the scope of attack vectors to account for evolving cybersecurity threats. These research directions will enhance 5G security and beyond by enabling more effective and adaptive penetration testing approaches.

Contributions of authors: conceptualization, methodology, formulation of tasks – **Maxim Iavich, Roman Odarchenko;** design and implementation of proposed method **– Maxim Iavich, Roman Odarchenko, Alla Pinchuk;** testing and validation **– Maxim Iavich, Alla Pinchuk;** writing – analysis of results, visualization, original draft preparation – **Maxim Iavich**, **Roman Odarchenko**, writing – review and editing **– Roman Odarchenko**, **Alla Pinchuk.**

Conflict of Interest

The authors declare that they have no conflict of interest concerning this research, whether financial, personal, authorship or otherwise, that could affect the research and its results presented in this paper.

Financing

This study was conducted without financial support.

Data Availability

The manuscript contains no associated data.

Use of Artificial Intelligence

The authors confirm that they did not use artificial intelligence technologies when creating the current work.

Acknowledgments

This work was supported by the Shota Rustaveli National Foundation of Georgia (SRNSFG) (NFR-22-14060).

All the authors have read and agreed to the published version of this manuscript.

References

1. *6G: Hype versus reality*. STL Partners, 2022. Available at: https://stlpartners.com/research/6g-hype-versus-reality/. (accessed 11.10.2024)

2. Hussain, M. A., Begum, A. U., Syed, Z., & Al-Ansari, D. M. S. Dynamic slicing optimization in 5G networks using a recursive LSTM mechanism with grey wolf optimization. *Journal of Theoretical and Applied Information Technology*, 2024, vol. 102, iss. 2, pp. 569-583.

3. *What is network slicing for 5G and beyond-networks*. Available at: https://telcomatraining.com/what-is-network-slicing-for-5g-and-beyond-networks (accessed 11.10.2024).

4. Hermosilla, A., Gallego-Madrid, J., Martinez-Julia, P., Ortiz, J., Kafle, V. P., & Skarmeta, A. Advancing 5G Network Applications Lifecycle Security: An ML-Driven Approach. *CMES-Computer Modeling in Engineering & Sciences*, 2024, vol. 141, iss. 2, pp. 1447-1471.

5. Kholidy, H. A., Karam, A., Reed, J. H., & Elazzazi, Y. An Experimental 5G Testbed for Secure Network Slicing Evaluation. *2022 IEEE Future Networks World Forum (FNWF)*, Montreal, QC, Canada, 2022, pp. 131-138. DOI: 10.1109/FNWF55208.2022.00032.

6. Kumar, P., & Sharma, R. Attack path discovery in dynamic network environments for automated penetration testing over 5G networks. *In Networks Attack Detection on 5G Networks using Data Mining Techniques*, CRC Press, 2024, pp. 143-163.

7. Benzaid, C., Taleb, T., & Song, J. AI-based Autonomic & Scalable Security Management Architecture for Secure Network Slicing in B5G. *IEEE Network*, 2022, pp. 1–9. DOI: 10.1109/mnet.104.2100495.

8. Cunha, V. A., & et al., 5 Growth: Secure and Reliable Network Slicing for Verticals. *2021 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit)*, Porto, Portugal, 2021, pp. 347-352. DOI: 10.1109/EuCNC/6GSummit51104.2021.9482536.

9. El Rajab, M., Yang, L., & Shami, A. Zero-touch networks: Towards next-generation network automation. *Computer Networks*, 2024, vol. 243, article no. 110294. DOI: 10.1016/j.comnet.2024.110294.

10. Liyanage, M., Pham, Q. V., Dev, K., Bhattacharya, S., Maddikunta, P. K. R., Gadekallu, T. R., & Yenduri, G. A survey on Zero touch network and Service Management (ZSM) for 5G and beyond networks. *Journal of Network and Computer Applications*, 2022, vol. 203, article no. 103362. DOI: 10.1016/j.jnca.2022.103362.

11. Lekidis, A. Towards 5G Advanced network slice assurance through isolation mechanisms. *In Proceedings of the 19th International Conference on*

Availability, Reliability and Security, 2024, article no. 136, pp. 1-7. DOI: 10.1145/3664476.3669923.

12. Chouman, A., Manias, D. M., & Shami, A. A Modular, End-to-End Next-Generation Network Testbed: Towards a Fully Automated Network Management Platform. *arXiv preprint*, 2024, arXiv:2403.15376.

13. Saad, S. B. Security architectures for network slice management for 5G and beyond. Doctoral dissertation, Sorbonne Université, 2023. 132 p.

14. Alwis, C. D., Porambage, P., Dev, K., Gadekallu, T. R., & Liyanage, M. A Survey on Network Slicing Security: Attacks, Challenges, Solutions and Research Directions. *IEEE Communications Surveys & Tutorials*, 2023, vol. 1. DOI: 10.1109/comst.2023.3312349.

15. Iavich, M., & Odarchenko, R. *Automated Penetration Testing in 5G Networks*. In: Nechyporuk, M., Pavlikov, V., Krytskyi, D. (eds) Integrated Computer Technologies in Mechanical Engineering - 2023. ICTM 2023. Lecture Notes in Networks and Systems, vol. 996. Springer, Cham, 2024. DOI: 10.1007/978-3-031-60549-9_33

16. Ou, X., Govindavajhala, S., & Appel, A. W. MulVAL: A logic-based network security analyzer. *In USENIX security symposium*, 2005, vol. 8, pp. 113-128.

17. Tayouri, D., Baum, N., Shabtai, A., & Puzis, R. A Survey of MulVAL Extensions and Their Attack Scenarios Coverage, 2022. DOI: 10.48550/arXiv.2208.05750.

Received 29.12.2024, Accepted 17.02.2025

РОЗРОБКА МЕТОДУ АВТОМАТИЗОВАНОГО ТЕСТУВАННЯ ПРОНИКНЕННЯ СЕГМЕНТІВ МЕРЕЖ 5G І НАСТУПНИХ ПОКОЛІНЬ

Р. Одарченко, М. Явіч, А. Пінчук

Предметом дослідження є методології тестування на проникнення для мереж 5G і наступних поколінь. **Метою** дослідження є розробка методології та програмного забезпечення для автоматизованого тестування на проникнення мережевої інфраструктури стільникових мереж наступного покоління з багаторівневою архітектурою. **Завдання:** 1) аналіз існуючих методів тестування на проникнення до мереж 5G і beyond та досліджень у цій галузі; 2) розробка нового методу автоматизованого тестування на проникнення до зрізів мереж 5G і beyond; 3) проєктування та реалізація методики у вигляді програмного забезпечення для віртуалізованих середовищ; 4) розробка архітектури тестової мережі 5G на основі рішень з відкритим вихідним кодом та методики проведення експериментів; 5) тестування та перевірка ефективності рішення при виявленні вразливостей та моделюванні реалістичних сценаріїв атак в середовищі тестової мережі 5G. У ході дослідження були отримані наступні **результати**: 1) розроблено новий метод автоматизованого тестування на проникнення в мережі 5G і наступних поколінь, який використовує генетичні алгоритми (GA) для оптимізації стратегій атак; 2) реалізовано програмний інструмент для автоматизації тестування на проникнення, що дозволяє ефективно виявляти критичні та високосерйозні вразливості та моделювати атаки в складному середовищі мережі 5G; 3) створено архітектуру тестової мережі для проведення експериментів, що дозволило провести контрольовану оцінку методології; 4) результати експериментів продемонстрували ефективність та працездатність запропонованого методу. **Висновки.** Основним внеском даного дослідження є розробка методології, реалізованої в програмному забезпеченні, для вдосконалення та автоматизації процесу тестування на проникнення. Отримані результати підтвердили працездатність та ефективність запропонованих рішень, зокрема продемонстрували покращене виявлення вразливостей, оптимізовану генерацію стратегій атаки та вищий відсоток успішності тестів на проникнення в складному мережевому середовищі.

Ключові слова: 5G і далі; тестування на проникнення; автоматизація тестування на проникнення; фреймворк для автоматизованого тестування; безпека 5G.

Одарченко Роман – д-р техн. наук, проф., декан Факультету аеронавігації, електроніки та телекомунікацій, Національний авіаційний університет, Київ, Україна.

Явіч Максим – д-р філос., проф., директор Центру кібербезпеки, Кавказький університет, Тбілісі, Грузія.

Пінчук Алла – студ. магістратури каф. телекомунікаційних та радіоелектронних систем, Національний авіаційний університет, Київ, Україна.

Roman Odarchenko – Doctor of Technical Sciences, Professor, Dean of the Faculty of Air Navigation, Electronics and Telecommunications, National Aviation University, Kyiv, Ukraine,
e-mail: roman.odarchenko@npp.kai.edu.ua, ORCID: 0000-0002-7130-1375, Scopus Author ID: 57188708598.

Maksim Iavich – PhD, Professor, Director of the Cyber Security Center, Caucasus University, Tbilisi, Georgia,
e-mail: miavich@cu.edu.ge, ORCID: 0000-0002-3109-7971, Scopus Author ID: 57194794201.

Alla Pinchuk – Master Student of the Department of Telecommunication and Radioelectronic Systems, National Aviation University, Kyiv, Ukraine,
e-mail: alla.pinchuk@kai.edu.ua, ORCID: 0000-0003-3567-0445, Scopus Author ID: 58490473900.