Citation: G. Amponis, P. Radoglou-Grammatikis, T. Lagkas, S. Ouzounidis, M. Zevgara, I. Moscholios, S. Goudos and P. Sarigiannidis, "Towards Securing Next-Generation Networks: Attacking 5G Core/RAN Testbed," 2022 Panhellenic Conference on Electronics & Telecommunications (PACET), 2022, pp. 1-4, doi: 10.1109/PACET56979.2022.9976365.

Towards Securing Next-Generation Networks: Attacking 5G Core/RAN Testbed

George Amponis[1,2], Panagiotis Radoglou-Grammatikis[1,3], Thomas Lagkas[2],
Savvas Ouzounidis [1], Maria Zevgara [1], Ioannis Moscholios [4], Sotirios Goudos [5], Panagiotis Sarigiannidis[3]

[1] Department of Research and Development, K3Y Ltd., Sofia, Bulgaria.
{gamponis, pradoglou, souzounidis, mzevgara}@k3y.bg
[2] Department of Computer Science, International Hellenic University, Kavala Campus, Greece.
{geaboni, tlagkas}@cs.ihu.gr
[3] Department of Electrical and Computer Engineering, University of Western Macedonia, Kozani, Greece.
{pradoglou, psarigiannidis}@uowm.gr
[4] Department Informatics & Telecommunications, University of Peloponnese, Tripolis, Greece.
{idm}@uop.gr
[5] Department of Physics, Aristotle University of Thessaloniki, Thessaloniki, Greece .
{sgoudo}@physics.auth.gr

***Abstract*—As the networking and communications landscape moves towards 5G and an increasing number of users are already accessing the Internet over 5G systems at an increasing pace, security issues rise and the corresponding vulnerabilities are in need of being addressed. The work presented in this paper constitutes an attempt at addressing the issue of training defenders capable of tackling cyberattacks and detection systems capable of timely notifying of security events. The key contribution of this paper is the proposal of a fully containerized testbed, incorporating a 5G cellular core, a radio access network (RAN), a set of potentially vulnerable hosts, and the appropriate entry points as interfaces. Attackers and defenders alike, can perform attacks or implement defensive measures correspondingly, without needing to exit the established sandbox. The developed testbed and emulation framework is envisaged to pave the path towards facilitating the generation of realistic datasets containing malicious traffic captured over 5G tunnels for enhancing the security of next generation networks.**

***Index Terms*—5G Testbed, Emulation, Cybersecurity Events, Radio Access Networks**

I. Introduction

Training defenders and detection systems capable of handling and detecting cybersecurity events in 5G networks is pivotal for enabling the undisturbed flow of information and minimizing downtime in critical and sensitive applications such as aerial communications [1], remote sensing, and industrial monitoring.

The communications transformation brought by next-generation cellular networks has already redefined existing connectivity and service reliability models so as to include pivotal security-by-design principles. These are deemed an absolute necessity to enable 5G to achieve its promise [2]. It is important to embed security capabilities to both classical networking and 5G-specific attack detection and prevention systems. Moreover, training defenders in the current networking landscape and keeping them up-to-date with the latest developments is proving to be increasingly difficult.

New security requirements have risen and are required to address the needs of different adjacent layers of the 5G architecture, namely physical, network, and application layer. Defensive systems and engineers need to implement new risk management frameworks so as to consider the evolving security threats landscape. Thus, 5G networks should consider additional security requirements, while at the same time, should address the need to maintain some level of modularity and flexibility, to train defenders and improve resilience and reliability [3]. To that end, we aspire to address the issue of lack of defence training baseline by introducing an integrated yet modular, all-in-one testbed capable of modelling cybersecurity events on a spectrum of layers, ranging from the 5G core (i.e., attacks specifically implemented within the 5G core network, against targeted NFs) to the RAN and the applications layer.

The rest of the paper has the following structure. Section II overviews the related work on testbeds for 5G systems. Implementation information about the proposed modular 5G testbed is provided in Section III. The emulated 5G cybersecurity attacks are described in Section IV. Lastly, Section V concludes the paper and provides future directions.

II. Related work

Authors in [4] propose a 5G testbed which allows the user to study performance issues and attacks on network slices. The aim of the authors is to provide a tool for a research study of network slicing; their proposed testbed uses OpenAirInterface for adding core and radio-level 5G elements to the testbed. More specifically, the UE and the AN are

simulated through the OAI Simulator. In this case, instead of representing network functions (NFs), docker containers represent entire network slices. The authors' work can be considered complementary to ours, from a technical point of view, as we deal with individual containerized NFs.

Authors in [5] engage in an in-depth study of the platforms and frameworks available to implement the 5G core network. The authors compare three open-source platforms, namely Magma, Open5GS and Free5GC. The core idea behind this comparative analysis is to address development and computational demands concerning flexibility, scalability and the appropriate use of available resources to deliver performance and various functions. We considered the outputs of this work to implement our own integrated testbed, electing Open5GS as the core network implementation, along with UERANSIM to implement the RAN.

Authors in [6] investigate the process and deployment aspects concerning 5G testbed structures and architectures. The authors also research the usage of an AI-based RAN slicing module and virtual network function (VNF) placement algorithm. The authors work is complementary to ours, as it concerns deployment and orchestration aspects of 5G testbeds leveraging AI.

III. Proposed Containerized 5G Testbed

The testbed proposed and implemented in the context of this paper utilizes a slight modification of Open5GS as the cellular core [7], and UERANSIM as RAN component of the overall synthesis [8]. The architecture of the testbed, allows for a great degree of modularity and extensibility, while its containerized nature supports minimal-overhead instantiation of tests and cybersecurity events, in a scalable, fully automated and near zero-touch manner. Moreover, the testbed's modularity allows for a set of attacks to be implemented simultaneously.

The testbed has been successfully tested with a set of cyberattacks, of both classical networking and 5G-specific nature. This paper focuses on performing a man-in-the-middle (MITM) and a brute force (BF) attack for eavesdropping traffic over a 5G tunnel and obtaining illegitimate access across a next generation cellular network.

In terms of implementing cellular connectivity, the proposed emulation framework is capable of supporting the following NFs:

- Access and Mobility Management Function (AMF)
- Session Management Function (SMF)
- User Plane Function (UPF)
- Network Slice Selection Function (NSSF)
- Network Exposure Function (NEF)
- Network Repository Function (NRF)
- Policy Control Function (PCF)
- Unified Data Management (UDM)
- Application Function (AF)
- Authentication Server Function (AUSF)
- Data Network (DN)
- Radio Access Network (RAN)
- User Equipment (UE)

The proposed work pioneers in the domain of generating malicious datasets, containing traffic corresponding to both classical and 5G-specific attacks. This will allow us to train artificial intelligence (AI) and machine learning (ML) systems to recognise preemptively take action against attacks, in cellular networks. To showcase the applicability of our approach, we validate our work by capturing malicious 5G traffic at both a core and a RAN level. The architecture of the testbed can be seen in 1.

To implement the aforementioned attack scenarios, we have networked three pivotal containers, namely the Server, the Attacker and the Victim. The Server is accessible by the Victim through a 5G connection to the data network (DN), while the Attacker and the Victim are directly networked at the RAN layer, and they constitute user equipment (UE) devices of UERANSIM.

The developed testbed can be instantiated with a single docker-compose command. The overall automated process is as follows. After starting all the appropriate containers implementing the above-mentioned NFs, the UEs are registered. This needs to be done prior to starting the UERANSIM simulation and is implemented so as to allow the proper exchange of handshakes with the emulated cellular antenae. This initial registration can either be done graphically via the included Open5GS webui, or by using a command-line interface (CLI) tool directly at the database handling user registrations. As this method is easier to automate, we chose it as a better alternative. The attributes needed to register a new user are the International mobile subscriber identity (IMSI), the unique Key, and the Operator Code (OPC).

For the purpose of automating the overall setup, we wrote a script to perform the relevant actions. Assuming that the registrations have been successful, the registry holding the relevant user data (in our case the mongodb database) will return:

```
WriteResult({
        "nMatched" : 0,
        "nUpserted" : 1,
        "nModified" : 0,
        "_id" : ObjectId("62711caaf...")
})
```

After registering the UEs using the above-mentioned method, the emulator launches all radio-layer processes. First it launches the gNB emulation using `./nr-gnb -c ./oai-gnb.yaml`, and then the UE emulation(s) using `./nr-ue -c ./oai-ue.yaml`. The individual subscribers can now successfully establish GTP-U tunnels with the third illustrated host entity (Host #3), using the newly created `uesimtun` interface.

IV. 5G Cybersecurity Scenarios

A. Scenario 1: MITM attack

The MITM scenario involves three nodes (Host 1, Host 2 and Host 3), where a malicious user intercepts and eavesdrops traffic exchanged between two legitimate hosts over a 5G tunnel. The participating entities are:

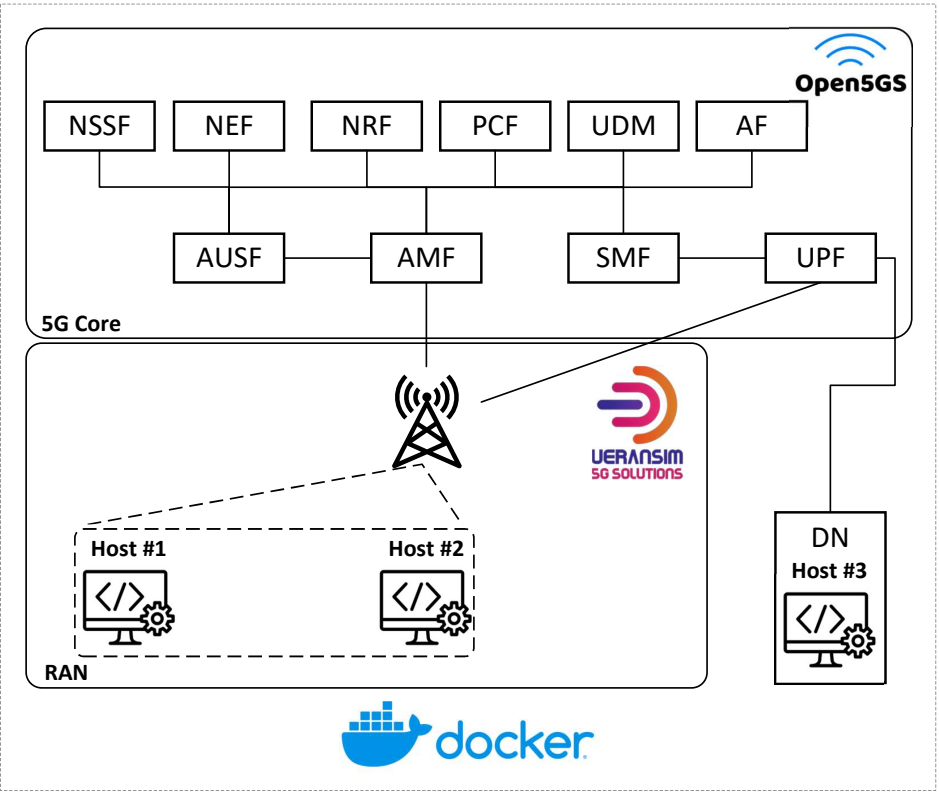

Figure 1. Architecture of the testbed.

- User: this is the ”victim” of the MITM attack. It attempts to load the contents of a web page by accessing a server located at the DN. The container corresponding to this entity has Firefox running on it, so that a user can test connectivity to the web page of interest. The idea behind this configuration is that an actual user can navigate graphically, and access the Server's contents over a virtualized 5G connection. To connect to firefox from the host, the user can visit `localhost:5800`.
- Server: this entity is hosting an http server, serving the files contained in is internal directory. This server uses http instead of https and can thus be used to demonstrate genuine data interception.
- Attacker: this entity is meant to be used as an illegitimate user instantiating the attack. This container comes ready with a set of scripts used to automate the attack process when required.

In this scenario, the hosts shown in Figure 1 interact as described below: Host 1 assumes the role of the legitimate user (i.e., the victim of the attack), while Host 2 assumes the role of the attacker and Host 3 assumes the role of the server (since it is accessible only through the DN). The steps followed to automatically implement the MITM attack are as described below. Firstly, the attacker (Host 2) discovers the IPs of the victim (Host 1) and the server (Host 3). The commands used by the automated script are `dig mitm_victim` and `dig mitm_server`. Secondly, with this information, the attacker will run arpspoof twice, once for each flow direction concurently; the commands used by the script are `arpspoof -t <server_ip> <victim_ip>` and `arpspoof -t <victim_ip> <server_ip>`. Thirdly, the emulator adds a forwarding rule to the attacker entity with `iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -j REDIRECT --to-port 8080`. This will add a rule to iptables to forward every packet with destination port 80 (therefore HTTP traffic) to the proxy. Fourthly, the attacker starts the transparent (i.e., non-blocking) interception process with the command `mitmproxy -m transparent`. Checking the output of the mitm proxy will show the POST and GET requests from and to the server. Alternatively, the malicious host can use any relevant tool to monitor and store the traffic (i.e., `tshark`). The mitigation process of the attack has also been implemented in an automatic manner by the emulator. This process involves utilizing `macchanger` to undo the above-mentioned arp cache poisoning.

B. Scenario 2: BF attack

The BF scenario involves two nodes (Host 1 and Host 3), where a malicious user attempts to gain access to a vulnerable service-component of the ONAP Software Defined Networking Controller (SDNC) Directed Graph (DG) Editor. The participating entities are:

- SDNC DG builder: this entity is the victim of the BF attack. This is the container exposing the ONAP DG service to the end-user. It is located outside of the networking scope of the attacker host, and is only accessible through the emulated 5G tunnel. This is done to ensure that we can obtain flow-level metrics, which can subsequently be utilized to train AI-based intrusion detection and prevention systems (IDPS), using solely 5G core-level datasets.
- SDNC Ansible: An Ansible server enabling the underlying ONAP SDN controller offer the DG service.
- SDNC Controller: this entity is SDNC controller which implements the logic of the targeted service
- Attacker: this entity is meant to be used as an illegitimate user, attempting to take control of the targeted service via a dictionary attack. The target is located at the DN, and as such, all traffic and connection requests will pass through the emulated 5G core network. The exploited vulnerability is that the ONAP SDNC DG Builder service uses basic authentication for its front-end, without any form of login rate-limitations being in place.

For the purpose of this scenario, we will only consider the ONAP SDNC DG builder entity as the target of the attack, for simplicity, mainly since the other networked entities of the ONAP SDNC service are purely supportive and are not directly targeted by the attacker. In this scenario, the hosts shown in Figure 1 interact as described below: Host 1 assumes the role of the attacker and Host 3 assumes the role of the scenario's target, i.e., the SDNC DG builder. The steps followed to automatically implement the BF attack are as described below. Firstly, the attacker (Host 1) performs an nmap to check for open ports at the target using the command `nmap -p- <target-IP>`. Port `3100` should be is open, running the opcon-xps service. A user can access the targeted service by visiting `<target_ip>:3100`. The authentication method is https-get. Secondly, considering the above remarks, the attacker runs hydra using the command `hydra -L <usr_list> -P <pass_list> https-get://target_ip:3100/ -t 4`. If the attack has been successful, the command will return the correct credential combination. The attack can be mitigated by reconfiguring the target entity to drop all packets originating from the attacker by adding the malicious address to a prohibition list.

V. Conclusions

This paper has showcased a containerized and integrated testbed, incorporating all necessary networking elements to achieve connectivity over 5G, while also supporting the implementation of and mitigation against different attacks, as seen in [9]. Using the proposed testbed, we were able to extract useful datasets at various levels, containing 5G traffic corresponding to cyberattacks and nominal traffic alike. The examined scenarios can provide valuable data on the targeted application-layer attacks both for defenders and AI-enabled detection systems. For example, an AI-enabled IDPS can be trained on such datasets, so as to timely terminate the appropriate communication links and potentially secure an otherwise vulnerable host. For the MITM attack, the captured datasets, as expected, contain malicious traffic, accessible through the N3 and N4 5G interfaces. Similarly, in the case of the BF attack, since the attack exclusively concerned traffic exchanged through the 5G tunnel, we were able the extract valuable metrics at a 5G core level. More specifically, by observing the typology and transport-layer characteristics of the GTP-U packets exchanged between the emulated cellular tower and the UPF, we were able to discern and pinpoint the timestamp at which the attack begins. Further modifications and enhancements can be made to the structure and interfaces of the emulator's architecture to facilitate training and attack-modelling on a greater spectrum of attacks and under various different 5G core network and RAN configurations.

Acknowledgement

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 952672 (SANCUS).

References

[1] G. Amponis, T. Lagkas, M. Zevgara, G. Katsikas, T. Xirofotos, I. Moscholios, and P. Sarigiannidis, "Drones in B5G/6G Networks as Flying Base Stations," *Drones*, vol. 6, no. 2, 2022. [Online]. Available: https://www.mdpi.com/2504-446X/6/2/39

[2] A. Dutta and E. Hammad, "5G Security Challenges and Opportunities: A System Approach," in *2020 IEEE 3rd 5G World Forum (5GWF)*, 2020, pp. 109–114.

[3] G. Iashvili, M. Iavich, R. Bocu, R. Odarchenko, and S. Gnatyuk, "Intrusion Detection System for 5G with a Focus on DOS/DDOS Attacks," in *2021 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*, vol. 2, 2021, pp. 861–864.

[4] A. Shorov, "5G Testbed Development for Network Slicing Evaluation," in *2019 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)*, 2019, pp. 39–44.

[5] F. J. De Souza Neto, E. Amatucci, N. A. Nassif, and P. A. Marques Farias, "Analysis for Comparison of Framework for 5G Core Implementation," in *2021 International Conference on Information Science and Communications Technologies (ICISCT)*, 2021, pp. 1–5.

[6] C. V. Nahum, L. De Nóvoa Martins Pinto, V. B. Tavares, P. Batista, S. Lins, N. Linder, and A. Klautau, "Testbed for 5G Connected Artificial Intelligence on Virtualized Networks," *IEEE Access*, vol. 8, pp. 223 202–223 213, 2020.

[7] P. Kiri Taksande, P. Jha, A. Karandikar, and P. Chaporkar, "Open5G: A Software-Defined Networking Protocol for 5G Multi-RAT Wireless Networks," in *2020 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, 2020, pp. 1–6.

[8] K.-L. Lee, C.-N. Lee, and M.-F. Lee, "Realizing 5G Network Slicing Provisioning with Open Source Software," in *2021 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 2021, pp. 1923–1930.

[9] P. R. Grammatikis, P. Sarigiannidis, C. Dalamagkas, Y. Spyridis, T. Lagkas, G. Efstathopoulos, A. Sesis, I. L. Pavon, R. T. Burgos, R. Diaz, A. Sarigiannidis, D. Papamartzivanos, S. A. Menesidou, G. Ledakis, A. Pasias, T. Kotsiopoulos, A. Drosou, O. Mavropoulos, A. C. Subirachs, P. P. Sola, J. L. Domínguez-García, M. Escalante, M. M. Alberto, B. Caracuel, F. Ramos, V. Gkioulos, S. Katsikas, H. C. Bolstad, D.-E. Archer, N. Paunovic, R. Gallart, T. Rokkas, and A. Arce, "SDN-Based Resilient Smart Grid: The SDN-microSENSE Architecture," *Digital*, vol. 1, no. 4, pp. 173–187, 2021. [Online]. Available: https://www.mdpi.com/2673-6470/1/4/13