2021 IEEE International Conference on Communications (ICC), author's copy

Towards 5G-ready Security Metrics

Lianying Zhao*, Muhammad Shafayat Oshman*, Mengyuan Zhang†,
Fereydoun Farrahi Moghaddam†, Shubham Chander*, Makan Pourzandi†
*Carleton University {lianying.zhao, muhammad.shafayatoshman, shubhamchander}@carleton.ca,
†Ericsson Canada {mengyuan.zhang, fereydoun.farrahi.moghaddam, makan.pourzandi}@ericsson.com

***Abstract*—The fifth-generation (5G) mobile telecom network has been garnering interest in both academia and industry, with better flexibility and higher performance compared to previous generations. Along with functionality improvements, new attack vectors also made way. Network operators and regulatory organizations wish to have a more precise idea about the security posture of 5G environments. Meanwhile, various security metrics for IT environments have been around and attracted the community's attention. However, 5G-specific factors are less taken into consideration.**

This paper considers such 5G-specific factors to identify potential gaps if existing security metrics are to be applied to the 5G environments. In light of the layered nature and multi-ownership, the paper proposes a new approach to the modular computation of security metrics based on cross-layer projection as a means of information sharing between layers. Finally, the proposed approach is evaluated through simulation.

***Index Terms*—Security Metrics, Network Function Virtualization, 5G, Attack Graph, Cloud**

I. Introduction

Security controls, e.g., firewalls and IDS, have been widely adopted by organizations and government agencies to protect their computing infrastructures. In addition to their performance, e.g., CPU consumption and detection rate, the effectiveness of security controls is of utmost importance, to answer the question of "whether or how much a security solution can improve the level of security". In the literature, security metrics [1], [2], [3] are used to calculate an overall risk level of the infrastructure/network that can facilitate decision making as to what can be done to improve security. Especially, in a large system, such as telecom networks, proper security metrics could both serve as guidelines for decision makers to improve security and help users/customers better understand security postures of their deployments.

More specifically, 5G brings a whole new template to the table. On one hand, the 5G environment demonstrates a dynamic nature, thanks to technologies like network slicing [4] and multi-access edge computing (MEC) [5], which allow for on-demand and dynamic allocation of resources. Also, the 5G ecosystem has been turned into a multi-party play, to a greater extent compared to its previous generations, i.e., we have multiple stakeholders sharing the platform and resources, sometimes isolated from each other for privacy and security.

The final version of this article will appear in the proceedings of IEEE International Conference on Communications (ICC'21), Jun. 14-23, 2021. This is the author's copy created on July 10th, 2021.

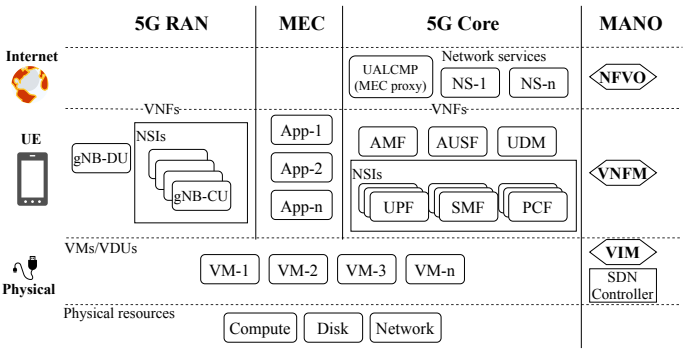


Fig. 1: An overview of the 5G environments

The flexibility and platform sharing of 5G environments are achieved by splitting the application logic from the infrastructure through Network Function Virtualization (NFV) [6] and Software-defined Network (SDN) [7]. 5G with NFV brings the cloud and virtualization to the traditional infrastructure where the network operator owns the whole stack. Nowadays, there is the option for the network operator to deploy virtual network functions (VNFs) on the hardware infrastructure of the cloud service providers (CSPs). Also, virtual network operators can rent the telecom infrastructure from network operators and provide service to end users. These new phenomena potentially pose challenges to the applicability of current security metrics.

Motivation. Typically, a security metric concerns identifying an aspect of the computing environment to measure, e.g., software, process or people [8], and optionally aggregating individual measures or measures of individual systems into an overall metric. From a high-level overview of the 5G environment as shown in Figure 1, one may have a few intuitive questions in mind when trying to apply security metrics: 1) The 5G core, MEC and RAN (radio-access network) might belong to different stakeholders and access to each other's resources is not shared with trust. How is aggregation performed horizontally? 2) The layered structure implies that the relationship with a lower-layer party (e.g., infrastructure provider) may have more complications than with peers (as in a regular cloud environment). How can aggregation be done vertically? 3) Even though one party can locally compute its own metrics independently, how are other parties around it taken into account, as they all affect this party's security? The subsequent discussions aim to shed light on answering these questions and we will revisit this 5G overview.

Contributions.

- We perform a gap analysis by examining existing security metrics in the context of the 5G telecom environments.
- We propose a modular security metric model that specifically addresses the multi-ownership of 5G. In particular, our approaches are discussed in a 5G-specific setting.
- We also evaluate the proposed modular model with simulated resources of varying characteristics and topologies.

II. Security Metrics in the 5G/telecom Context

In this section, we try to identify factors that have been under-considered in the previous work discussed in Section VI and perform a gap analysis in the 5G/telecom context. Gaps will be denoted as G1, G2 and so forth.

A. Assumptions

We base the discussion on the 5G stakeholders defined in the 5G-PPP White Paper on the 5G architecture [9]. We also make further assumptions as follows.

Hardware and software vendors are always *trusted*. Specifically, we exclude supply chain attacks where hardware components are shipped compromised or software is intentionally developed to be buggy/malicious. In this paper, we do not intend to discuss specific attacks, but instead only use potential threats (e.g., zero-days) to analyze/adapt security metrics in the 5G context. As an example, an MEC application may be buggy which allows a connected UE to compromise it. This is possible in reality but we do not refer to a specific vulnerability.

B. G1: A threat model shift

The current studies of security metrics usually consider the owner of the whole infrastructure as trusted [10], e.g., the administrator of the cloud service can faithfully provide data for computation and reliably preserve privacy for data collected from tenants. However, the multi-ownership nature of the 5G environment usually imposes more challenges on such assumptions:

- The Service Customer (SC) can be *malicious*. For example, a small IoT communication platform may misbehave (before getting detected by the administration).
- The Service Provider can be *curious*.[1] For the business reputation, they may ensure a transaction's integrity but not its secrecy, e.g., not disclosing the fact that it is collecting tenant transaction logs or not properly erasing tenant secrets. Curious adversaries have been common [11], which may significantly undermine privacy.
- Any party may unintentionally fail to take adequate security measures. For example, the private key might be distributed in the whole environment, more broadly than necessary. Data sent to or received from such a party will be insecure.

We thus need to re-examine current security metric studies, traditional and cloud-based, with this shifted threat model: the stakeholders of the environment with different interests might be *mutually dis-/semi-trusted*.

[1] In the 5G-PPP architecture, this may include the Network Operator (NOP, providing network services to SPs), the Virtualisation Infrastructure Service Provider (VISP) and the Data Centre Service Provider (DCSP).

C. G2: Input collection barriers

As measuring security requires inputs from all involved parties, data unavailability caused by isolation between parties can pose challenges to security metric computation as well. See Figure 2a.

Isolation has been a key approach to information system security, which is the idea of preventing information from flowing from one party to another. This ensures that unintended parties cannot see or access unauthorized resources. Mechanisms enabling isolation include primitive architectural support such as the x86 protection rings (Ring 0 – Ring 3, isolating userspace and kernelspace) and hypervisor (deemed to be Ring -1), and the more recent trusted computing technologies like Intel SGX. We refer to isolation achieved with such mechanisms as *technical isolation*.

Apart from technical isolation, organizations and individuals also need to comply with business agreements, organizational regulations and law. One example is the service level agreement (SLA) between a tenant and a CSP. We term isolation achieved by such factors as *institutional isolation*.

D. G3: Localized security metrics

In addition to holistic security metrics reflecting the security level of the whole site, each stakeholder might seek to evaluate the security level of their own environment independently.

For instance, a virtual operator may wonder the security of its UDM (unified data management, similar to 4G's HSS) when deployed in a cloud environment and connected to an operator's service. In this case, security metrics concerning resources of the cloud or the operator will not help. What is different here is that the target is within the virtual operator's premises but attack paths leading to it may cross all other parties of the whole site. Therefore, what is needed is a localized security metric incorporating the influences of the whole environment.

E. G4: Dynamicity and flexibility

Compared to previous generations, although the transition is gradual, 5G demonstrates more dynamicity in resource allocation and more flexibility in configuration. This poses challenges to the static computation of security metrics, as input collection and computation both cost. For instance, in the case of network slicing, VM (VNF) instances are spawned dynamically which affects the topology of the statically computed attack paths. If taking into account such dynamicity stemming from different parties (e.g., unplanned or unpredictable), it becomes more difficult for current security metrics to reflect up-to-date situations.

III. Ways Forward to Bridge Gaps

Based on the gaps identified, in this section we explore a few possible directions to address them. As shown in Figure 1, the 5G telecom environments naturally demonstrate a layered structure, from the physical resources, virtualized infrastructure (the NFVI layer), virtual network functions (the VNF layer), to network services (the orchestration layer). This

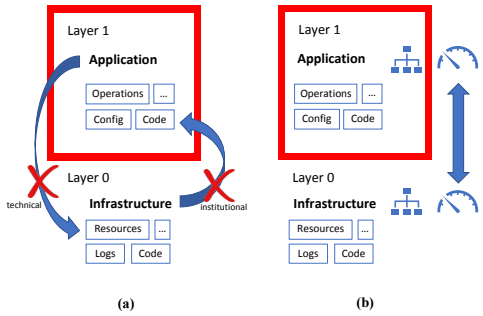


Fig. 2: Isolation and cross-layer projection

layered structure has been one of the considerations in the previous studies (e.g., [12]). We inherit this layered structure as a foundation of our discussion.

A. Trustworthy input collection: G1

The threat model shift makes it hard to trust a potentially malicious or curious third-party with access to one's own resources for input collection. On the other hand, self-reporting can also be unfaithful, e.g., a dishonest provider may just report incorrect software versions in his favor.

In this case, *hardware assistance* plays a more important role when mutual trust is weakened, especially for more privileged parties, such as cloud service providers, assuming that hardware vendors still remain trusted.

There have been certain attempts to protect VNFs' execution and corresponding secrets with hardware security support [13], [14], such as Intel SGX (unprivileged, no access to the system environment). Making use of Intel TXT or AMD SVM/SEV (privileged, capable of accessing the system environment) [15] or TrustZone on ARM servers, we can enforce trustworthy input collection by implementing the collection functions within TXT or SVM, e.g., scanning for software versions.

B. Cross-layer projection: G2

To enable information sharing in the face of isolation, each party can choose to preprocess the information by removing or aggregating details but still preserving the semantics. For instance, an infrastructure provider may expose meaningful semantics for the guest, e.g., VM create, init, start and terminate, instead of the actual operation logs. For attack graphs, a local chunk of nodes and edges may be generated and computed for a metric (e.g., the shorted path), which is then shared instead of the actual attack graph.

As shown in Figure 2b, we define *projection* as such aggregated or preprocessed sharing across layers, preserving semantics for the receiving party. The principle of least exposure, similar to least privilege, can be followed, i.e., depending on how security metrics are computed, only the minimum information enough for the computation is shared.

A key enabler: standardization. To ensure that projection is performed in a consistent manner across layers, it is crucial to specify what information should be collected, what kind of preprocessing is allowed and what is eventually projected. Various industrial standards have covered related topics to some extent. For example, in addition to SCAP (defined in NIST IR 800-117), NIST SP 800-137 also involves a guideline to collect information according to established security metrics.

Another important role that standardization plays is to make security metrics comparable. As the main usefulness of security metrics is to serve as a basis for comparison, it makes no sense if different parties perform incompatible metrics, e.g., a 5 by party A cannot be compared with an 8 by party B. Therefore, standardizing the security metrics is also necessary.

C. Modular security metric computation: G3

To satisfy the requirement for localized security metric computation (G3), at least two goals need to be achieved: 1) A local resource can be selected as the target for which security metrics are computed, as if no other parties existed. 2) External threats affecting the local resource should also be included.

We propose to modularize the computation by following the natural boundaries of different owners. This way, the owner can compute its local metrics independently, which achieves the first goal. For the second, the modules can be further aggregated through the cross-layer projection. See Section IV for detailed discussion.

D. Incremental and iterative computation: G4

The modular model can also be applied to partially address the dynamicity of the 5G environments (G4). When graph modules belong to the same owner, some modules may be more static and some others may be frequently updated (e.g., network slices). *Incremental* computation saves resources and especially reduces response time if dynamic components introduce changes. Only affected modules need to be recomputed.

As a by-product, the modular module also allows iterative computation. This can address a large-scale environment. A huge attack graph can be broken down this way into smaller blocks, each calculated with intermediate values to be aggregated in the next iteration. We leave the detailed discussion as future work.

IV. Modular Security Metrics

In this section, we demonstrate our approaches to modular security metric computation based on the commonly-used attack graph model [16]. Note that as we do not target specific attacks, such attack graph actually becomes a resource graph [17] (relying on zero-day vulnerabilities), which we still refer to as attack graphs.

Attack paths. To facilitate discussion, we first examine how an attacker can take multiple paths to compromise a targeted resource in the 5G environment. These attack paths will form the foundation of subsequent discussions. With the UDM considered as the attack target, three starting points are possible (refer back to Figure 1):

- User equipment (UE). With the advent of edge computing, a smart phone may interact with the access network for application logic other than wireless communication. We assume an MEC-hosted IoT application compromised by the UE, which then further attacks its UALCMP (User

Application LifeCycle Management Proxy) located in the core. If the UALCMP is co-located in the same host with the target VNF, an attack path is formed.

- Physical access. Base stations (gNB for 5G) need to spread geographically for signal coverage. Given the popularity of pico cells and femtocells, an adversary has the chance to physically compromise it [18] without needing to access the operator's protected premises. Taking advantage of the less protected back-haul communication, various attack paths may go to the core VNFs.
- Internet. Certain VNFs also have separate Internet access, and as with any network-facing applications, open ports with vulnerabilities can open the door for attacks.

The multi-ownership situation is demonstrated by the following parties: a virtual operator will be the subject here owning certain 5G core functions. The RAN (with gNB1 and gNB2) belongs to another network operator. Naturally, the MEC platform may be owned by a different party.

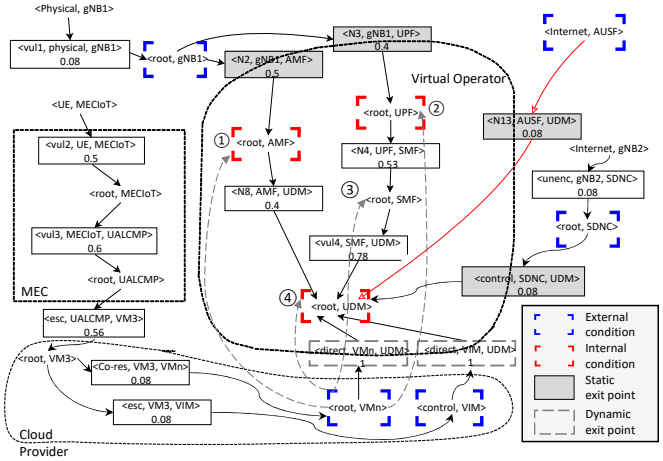


Fig. 3: Modularized attack graph. Surrounded by dashed lines are the modules (MEC, Cloud Provider and Virtual Operator).

Channels. 5G components are connected to each other via various channels. That is also how attack paths can be formed. For instance, the virtual operator's AMF (Access and Mobility Management Function) talks to the gNB via the N2 interface [19] and a gNB communicates with another gNB via the X2 interface [20].

Based on these attack paths, the constructed attack graph is shown in Figure 3. Nodes in point brackets are pre- or post-conditions, e.g., $\langle root, gNB1\rangle$ means having the root privilege on gNB1. Nodes in rectangles are exploits. bridging pre-conditions and post-conditions. For instance, $\boxed{\langle N2, gNB1, AMF\rangle}_{0.5}$ means by exploiting the N2 interface, $\langle root, gNB1\rangle$ can lead to $\langle root, AMF\rangle$ with the probability of $p()$ = 0.5. $p()$ is assigned just to demonstrate calculation.

In addition to $p()$, $P()$ is a cumulative probability of an attacker following the attack paths and executing the exploit. We calculate $P()$ with the Bayesian conditional probability using individual $p()$s, as is done by Frigault et al. [21]. In parallel, we also consider the shortest path approach [22] which reflects the least amount of effort an attacker can expend to achieve the exploit.

Now our task is to enable the virtual operator to compute $P(\langle root, UDM\rangle)$ and the shortest path $Path_{min}(\langle root, UDM\rangle)$ with influences projected from other modules.

The border exploits. As each exploit node always involves two conditions (in many cases on two different components), on the ownership boundary, some exploits have one condition pertaining to one party (e.g., gNB1 with the RAN) and one condition to another (e.g., AMF with the virtual operator). We term such exploits as border exploits.

We define the pre-conditions of the border exploit as *external conditions*, and the post-conditions of the border exploit as *internal conditions*. When the virtual operator is the subject, external conditions are blue-boxed and internal conditions are red-boxed. In the following, we explore how projection can be performed between the two sets of conditions in three steps.

Step 1: Local graph. The virtual operator will first perform the local computation by identifying all its internal conditions on outbound channels (i.e., AMF, UPF and UDM in Figure 3). Around the target UDM, a local attack graph is generated. So the local metric of the UDM can be calculated with the Bayesian network [21] as:

$$P(\langle root, UDM\rangle) = Bayesian(N)$$

where N stands for the local graph as the Bayesian network starting from the internal conditions in red boxes. In this example, $P(\langle root, UDM\rangle) = 0.648$ (conditional probability table omitted here. See [21] for the calculation details). $Path_{min}(\langle root, UDM\rangle) = 1$, from $\langle root, AMF\rangle$. Likewise, each external party can define their border components as the target (in blue boxes) and generate their attack graph.

Step 2: Static exit points. When it comes to projection from external conditions to internal conditions, the channels in between come to attention. A simple situation is static preconfigured channels, such as between constantly-connected components, or any channels both parties would like to hard-code.

We call such external conditions static exit points (in gray boxes). In this case, the external conditions are simply passed along via the static channels. From the RAN owner, $\langle root, gNB1\rangle$ gets $P(\langle root, gNB1\rangle) = p(vul1) = 0.08$, and its shortest path is 1.

Upon receipt of the external conditions, the border exploits can be reconstructed (0.5 and 0.4) as their post-conditions are in the virtual operator's premises. We omit the calculation for the two other border exploits $\langle N13, AUSF, UDM\rangle$ and $\langle control, SDNC, UDM\rangle$. The updated calculation with static exit point projection is: $P_{\text{static}}(\langle root, UDM\rangle) = Bayesian(N_{\text{static}}) = 0.11$. The corresponding $Path_{min}(\langle root, UDM\rangle)$ is still 1, due to the influence by the AUSF.

Step 3: Probing dynamic exit points. In most cases, even when the channels are known, what external components are

connected can be unpredictable or vary. An upstream TCP/IP link can have dynamic hosts via DHCP, similar to network slice instances spawned in real-time. Note that the virtual operator may identify more channels than its actual internal conditions without prior knowledge of other attack graph modules. In such cases, we can dissociate the border exploits from channels. This is demonstrated as dynamic exit points (grayed dotted boxes) at the bottom of Figure 3.

Upon receipt of the external conditions, the virtual operator can enumerate its channels and probe for choices that achieve the optimum for the local attack graph. In this example, it probes its multiple VNFs with $P(\langle root, VMn \rangle)$, e.g., AMF, SMF, UPF and UDM. Note that it will not be able to distinguish the placements ①, ②, ③ and ④, even though ④ is the actual.

Eventually after the 3-step modular computation, the consolidated metric would be:

$$P_{\text{Consolidated}}(\langle root, UDM \rangle) = \max_{i=1,2,3,\ldots,n} (P_{\text{static,dynamic}(i)})$$

where $P_{\text{static,dynamic}(i)}$ is calculated with $Bayesian(N_{\text{static}})$ when combined with each probed placement ($N_{\text{dynamic}(i)}$). For simplicity, we only calculate for VMn (excluding the VIM), and the max of the 16 combinations for placing AMF, UPF, SMF, and UDM on VMn: $P_{\text{Consolidated}}(\langle root, UDM \rangle)$ = 0.122. $Path_{min}(\langle root, UDM \rangle) = 1$.

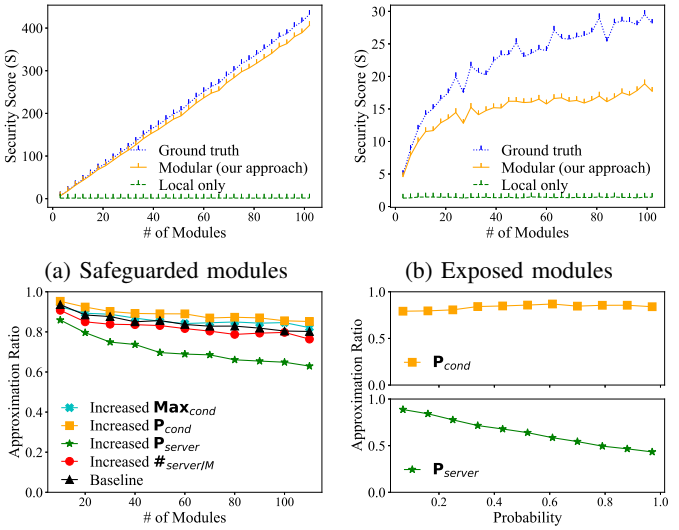
V. Simulation and Discussion

To statistically and visually evaluate the effectiveness of our modular model, we conduct simulations based on attack graphs. We utilize the shortest path approach (denoted as "Security Score (S)") as the security metric. All the simulations have been conducted in a VM equipped with 7 vCPUs and 16GB RAM in a Python environment under Ubuntu 18.04. We repeat each simulation 100 times and take the average.

In the first set of the simulation, we evaluate how the modular model impacts the accuracy of the computed security score with different graph generation parameters. The three data sets we plot are the security scores computed for the same target, based on: 1) The local attack graph generated from resources the target's owner has access to, denoted as "Local only". 2) The full attack graph for the whole environment, assuming no input collection barriers. We refer to this set as the "Ground Truth" for comparison. 3) Our modular computation of the local graph with projected inputs from other attack graphs, referred to as "Modular (our approach)".

The attack graph generation parameters are: *# of Modules* refers to the total number of attack graph modules, one of which is selected as the local graph. P_{server} is the probability of any two servers being connected (we call the physical/virtual machines servers). P_{cond} means the probability of any two conditions being connected. Max_{cond} means the maximum number of conditions a server can have. $\#_{server/M}$ refers to the number of servers per module.

Interpretations of Set I results. In general, compared to the green line at the bottom (the local only computation), the modular enables a more accurate security evaluation, i.e., the modular security score is much closer to the ground truth. Figure 4a shows the experiments based on "safeguarded" modules, i.e., modules being less exposed to the exit points. From the results, our modular computation is very close to the ground truth even when the number of modules reaches 100, each of which may represent a virtual operator (hence we consider 100 as sufficiently large). This is mainly because border exit points cause information loss between modules, thus, less exposure to exit points means less information loss. Figure 4b corresponds to exposed modules, i.e., modules containing more border exploits. For the aforementioned reason, our approach performs less well compared to the safeguarded modules. Nonetheless, we still provide an approximated security evaluation comparing to the local only, which does not reflect the actual security level.



(a) Safeguarded modules (b) Exposed modules

(c) AR vs scale (d) Exploitability and connectivity

Fig. 4: Effects of modular attack graph computation.

In the second set of the simulation, we focus on evaluating the *approximation ratio* = $\frac{S_{\text{Modular}}}{S_{\text{GroundTruth}}}$ when varying graph parameters. The approximation ratio (AR) indicates the deviation from the ground truth; 1 means the same as ground truth and 0 means failure to reflect the actual security level.

Interpretations of Set II results. In Figure 4c the black line (the line with triangular markers) is the baseline setup. We increase each parameter (by ∼%50) to study how AR changes when the scale of the 5G environment increases. Overall, AR changes are very negligible (which is also reflected in Figure 4d-top the line is almost flat) except when we increase P_{server}. This is mainly because P_{server} directly contributes to the connectivity complexity. When the attack graph complexity increases, the aggregated local shortest path in each module fails to accurately represent the shortest path in the entire attack graph. On the other hand, other parameters are less connectivity-related, e.g., P_{cond} and Max_{cond} tend to affect the exploitability (probability). As we are taking the shortest

path (not the Bayesian probability), we are ignoring the exploitability. Figure 4d takes another perspective to show how P_{cond} and P_{server} affect AR by plotting against the probability. Coherent with our observations in Figure 4c, as P_{cond} goes up, AR does not change much, while when P_{server} increases AR goes down significantly.

VI. Related Work

Software vulnerabilities are commonly used as inputs to evaluate the security level of a network/system. For example, Singhal et al. [23] proposed a system where Common Vulnerability Score System (CVSS) scores are one of the inputs taken for security metrics calculation. Contrary to the popular perception that only vulnerabilities with high severity pose danger to the system, the chaining of vulnerabilities, i.e., attack path, can combine several low-severity vulnerabilities to create greater damages. Wang et al. [24] demonstrated a probabilistic security metric that joins all the vulnerabilities in the system and gives a score in the form of probability, outlining which path can be taken by an attacker. Jajodia et al. [25] proposed a topological analysis on how several vulnerabilities can be chained together to mount an exploit.

The studies of security metrics have also touched the cloud. Caron et al. [26] proposed a security metric designed for cloud environments and using this metric, came up with optimized placements of VMs in line with user security requirements. Torkura et al. [27] performed a quantitative analysis on how security metrics could be used to improve cloud security, using OpenStack as a case study. Alhebaishi et al. [10] modeled cross-layer and co-residency attacks in the NFV stack and used optimized VM placement to mitigate such attacks. To evaluate multi-tenancy threats in the cloud, Madi et al. [28] introduced multi-level distance metrics based on compute, physical and network distances, taking into account the level of resource sharing between tenants.

Summary. Different from the aforementioned, our work focuses on 5G-specific environments and proposes a modular model capturing the gaps introduced by the new platform and use cases. Note that this is orthogonal to existing security metrics work as we are proposing a new way of applying them.

VII. Concluding Remarks

We examined what is special about the more recent 5G telecom environments in terms of applying existing security metrics, and were able to identify four gaps at a high level. Among corresponding potential solutions, in particular, we propose a security metric model that allows for modular computation with projected inputs, using 5G-specific examples. The feasibility of our proposal has also been evaluated with simulation. We believe that our exploration can cast light on future research into making security metrics 5G-ready (or for later generations).

References

[1] M. Pendleton, R. Garcia-Lebron, J.-H. Cho, and S. Xu, "A survey on systems security metrics," *ACM Comput. Surv.*, vol. 49, Dec. 2016.

[2] A. Ramos, M. Lazar, R. H. Filho, and J. J. P. C. Rodrigues, "Model-based quantitative network security metrics: A survey," *IEEE COMST'17*, vol. 19, no. 4, pp. 2704–2734.

[3] J.-H. Cho, S. Xu, P. M. Hurley, M. Mackay, T. Benjamin, and M. Beaumont, "Stram: Measuring the trustworthiness of computer-based systems," *ACM CSUR'19*, vol. 51, no. 6, pp. 1–47.

[4] P. Hedman, "Description of network slicing concept," tech. rep., NGMN Alliance, 2016.

[5] S. Shahzadi, M. Iqbal, T. Dagiuklas, and Z. U. Qayyum, "Multi-access edge computing: open issues, challenges and future perspectives," *Journal of Cloud Computing*, vol. 6, no. 1, p. 30, 2017.

[6] R. Mijumbi, J. Serrat, J.-L. Gorricho, N. Bouten, F. De Turck, and R. Boutaba, "Network function virtualization: State-of-the-art and research challenges," *IEEE COMST'15*, vol. 18, no. 1, pp. 236–262.

[7] D. Kreutz, F. M. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, 2014.

[8] Z. Abbadi, "Security metrics: What can we measure," in *Open Web Application Security Project (OWASP)*, vol. 2, 2011.

[9] S. Redana, Ö. Bulakci, A. Zafeiropoulos, A. Gavras, A. Tzanakaki, A. Albanese, A. Kousaridas, A. Weit, B. Sayadi, B. T. Jou, *et al.*, "5G PPP architecture working group: View on 5G architecture," 2019.

[10] N. Alhebaishi, L. Wang, and S. Jajodia, "Modeling and mitigating security threats in network functions virtualization (NFV)," in *DBSec'20*, pp. 3–23, Springer.

[11] Z. A. Kissel and J. Wang, "Verifiable phrase search over encrypted data secure against a semi-honest-but-curious adversary," in *ICDCS'13*, pp. 126–131.

[12] S. Lakshmanan Thirunavukkarasu, M. Zhang, A. Oqaily, G. S. Chawla, L. Wang, M. Pourzandi, and M. Debbabi, "Modeling NFV deployment to identify the cross-level inconsistency vulnerabilities," in *CloudCom'19*, pp. 167–174.

[13] N. Paladi and L. Karlsson, "Safeguarding VNF credentials with Intel SGX," in *SIGCOMM'17*, (Los Angeles, CA, USA), p. 144–146, 2017.

[14] M.-W. Shih, M. Kumar, T. Kim, and A. Gavrilovska, "S-nfv: Securing NFV states by using SGX," in *SDN-NFV Security'16*, pp. 45–48.

[15] Openstack.org, "Trusted compute pools." Available at https://docs.openstack.org/nova/pike/admin/security.html [Accessed Jul. 2, 2020].

[16] O. Sheyner, J. W. Haines, S. Jha, R. Lippmann, and J. M. Wing, "Automated generation and analysis of attack graphs," in *S&P'02*, pp. 273–284, 2002.

[17] M. Zhang, L. Wang, S. Jajodia, A. Singhal, and M. Albanese, "Network diversity: A security metric for evaluating the resilience of networks against zero-day attacks," *IEEE TIFS'16*, vol. 11, no. 5, pp. 1071–1086.

[18] X. Zou, "4G LTE man in the middle attacks with a hacked femtocell." HITB GSEC 2019. https://gsec.hitb.org/sg2019/sessions/4g-lte-man-in-the-middle-attacks-with-a-hacked-femtocell/ [Accessed Jul. 2, 2020].

[19] 3GPP TS 38.413, *NG-RAN; NG Application Protocol (NGAP)*, 2020. Rel. 15.

[20] 3GPP TS 36.423, *Evolved Universal Terrestrial Radio Access Network (E-UTRAN); X2 Application Protocol (X2AP)*, 2020. Rel. 8.

[21] M. Frigault, L. Wang, A. Singhal, and S. Jajodia, "Measuring network security using dynamic bayesian network," in *QoP'08*, pp. 23–30.

[22] N. Idika and B. Bhargava, "Extending attack graph-based security metrics and aggregating their application," *IEEE TDSC'12*, vol. 9, no. 1, pp. 75–85.

[23] A. Singhal and X. Ou, "Techniques for enterprise network security metrics," in *CSIIRW'09*, pp. 1–4.

[24] L. Wang, T. Islam, T. Long, A. Singhal, and S. Jajodia, "An attack graph-based probabilistic security metric," in *DBSec'08*, pp. 283–296, Springer.

[25] S. Jajodia, S. Noel, and B. O'berry, "Topological analysis of network attack vulnerability," in *Managing cyber threats*, pp. 247–266, Springer, 2005.

[26] E. Caron, A. D. Le, A. Lefray, and C. Toinard, "Definition of security metrics for the cloud computing and security-aware virtual machine placement algorithms," in *CyberC'13*, pp. 125–131, IEEE.

[27] K. A. Torkura, F. Cheng, and C. Meinel, "Application of quantitative security metrics in cloud computing," in *ICITST'15*, pp. 256–262, IEEE.

[28] T. Madi, M. Zhang, Y. Jarraya, A. Alimohammadifar, M. Pourzandi, L. Wang, and M. Debbabi, "Quantic: Distance metrics for evaluating multi-tenancy threats in public cloud," in *CloudCom'18*, pp. 163–170.