



(51) International Patent Classification:

G06F 8/70 (2018.01) H04L 29/06 (2006.01)
G06F 9/54 (2006.01) H04L 29/08 (2006.01)
G06F 17/00 (2019.01)

(21) International Application Number:

PCT/US2021/058421

(22) International Filing Date:

08 November 2021 (08.11.2021)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

63/111,306 09 November 2020 (09.11.2020) US

(71) Applicant: **THE TRUSTEES OF PRINCETON UNIVERSITY** [US/US]; 87 Prospect Avenue, Princeton, NJ 08544 (US).

(72) Inventors: **BROWN, Jacob**; 3-400 Walmer Road, Toronto, ON M5P2X7 (CA). **SAHA, Tanujay**; 5569 First Campus Center, Princeton, NJ 08544 (US). **JHA, Niraj, K.**; 20 Norbridge Drive, Princeton, NJ 08540 (US). **AARAJ, Najwa**; Apt 3012 The Address Downtown, Dubai Downtown, Dubai (AE).

(74) Agent: **DRACHTMAN, Craig, M.**; Meagher Emanuel Laks Goldberg & Liao, LLP., One Palmer Square, Suite 325, Princeton, NJ 08542 (US).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO,

(54) Title: SYSTEM AND METHOD FOR MACHINE LEARNING ASSISTED SECURITY ANALYSIS OF 5G NETWORK CONNECTED SYSTEMS

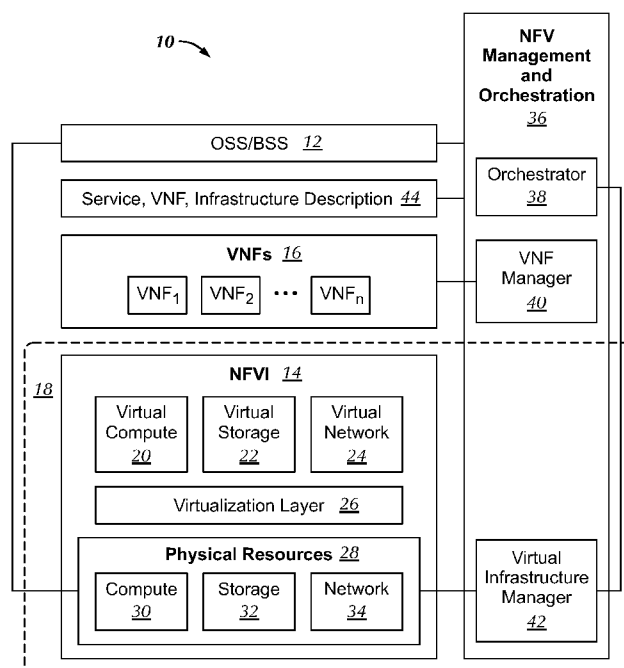


FIG. 1

(57) Abstract: According to various embodiments, a method for detecting security vulnerabilities in a fifth generation core network (5GCN) is disclosed. The method includes constructing an attack graph from a plurality of regular expressions. Each regular expression corresponds to a sequence of system level operations for a known 5GCN attack. The method further includes performing a linear search on the attack graph to determine unexploited 5GCN attack vectors where path in the attack graph that does not represent a known 5GCN attack vector represents an unexploited 5GCN attack vector. The method also includes applying a trained machine learning module to the attack graph to predict new 5GCN attacks. The trained machine learning module is configured to determine a feasibility of linking unconnected nodes in the attack graph to create a new branch representing a new 5GCN vulnerability exploit.

DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, IT, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

(84) Designated States (*unless otherwise indicated, for every kind of regional protection available*): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

- *with international search report (Art. 21(3))*
- *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))*

SYSTEM AND METHOD FOR MACHINE LEARNING ASSISTED SECURITY ANALYSIS OF 5G NETWORK CONNECTED SYSTEMS

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application claims priority to provisional application 63/111,306, filed November 9, 2020, which is herein incorporated by reference in its entirety.

STATEMENT REGARDING FEDERALLY SPONSORED RESEARCH OR DEVELOPMENT

[0002] This invention was made with government support under Grant No. CNS-1617628 awarded by the National Science Foundation. The government has certain rights in the invention.

FIELD OF THE INVENTION

[0003] The present invention relates generally to security analysis of 5G networks and more particularly to systems and methods for a machine learning based framework for discovering vulnerabilities in a 5GCN by creating and analyzing attack graphs.

BACKGROUND OF THE INVENTION

[0004] Fifth generation (5G) networks hold promise for realizing the vision of universal connectivity. They enable various verticals like Internet-of-Things (IoT), autonomous vehicles, smart cities, and telemedicine. These applications require high-bandwidth, robust, flexible, dynamic, and fault-tolerant network architectures.

[0005] 5G networks represent a huge leap, both qualitatively and quantitatively, from previous-generation telecommunication networks. The network core architecture has undergone a paradigm shift from its predecessor, the Evolved Packet Core. Previously, network functions were implemented on commodity hardware. In 5G networks, the network functions are mostly implemented in software. Moreover, with the advent of cloud computing, many network operations are now virtualized. This allows multiple operators to use the same underlying hardware resources to provide network services. This technology is broadly known as network function virtualization (NFV). 5G networks also separate the communication on the data plane from that on the control plane. This involves the use of a controller that observes the entire network before making routing decisions. This technology, broadly referred to as software-defined networking (SDN), has been shown to reduce both the

operational expenditure (OPEX) and capital expenditure (CAPEX) of the network. Many of these transitions have become possible due to the utilization of an mm-wave technology in 5G. Incorporation of these new technologies results in significant improvements in efficiency, reliability, and flexibility of wireless networks.

[0006] The confluence of the new technologies makes the 5G core network (5GCN) an intricate system including software-defined networking (SDN), network function virtualization (NFV), distributed systems, and cloud computing. The 5GCN has a service-based architecture that dynamically modifies itself according to the requirements of the operators and users. However, introduction of new technologies into the 5GCN also expands its attack surface, as it now inherits the vulnerabilities of all these individual technologies.

[0007] Security and privacy of users are of prime importance in 5G networks. The Third Generation Partnership Project (3GPP) has been working continuously to define the security standards of 5G communication systems. Multiple versions of security standards have been published to date. Recent surveys and articles list the potential vulnerabilities of various 5G-enabling technologies like cloud radio access networks, SDN, NFV, network slicing, cloud computing, and multi-edge computing.

[0008] There are many vulnerabilities that exist in the SDN ecosystem. Multiple implementation vulnerabilities exist in various open-source SDN controllers and network operating systems (NOSs) like OpenFlow, POX, and Open-Daylight. Similarly, network slicing and NFV have their own vulnerabilities. NFV inherits many of its vulnerabilities from traditional virtualization technologies. However, prior work does not report on the specific attack vectors that can exploit these vulnerabilities in the 5G framework and lacks detailed analyses of the impact of these vulnerabilities on the end-user. No prior work appears to explore interactions among vulnerabilities of different technologies, like SDN and NFV, to generate complex attack vectors.

[0009] Attack graphs can be used to analyze 5GCN security. Attack graphs have found extensive use in network security, software, and electronic systems. Various vulnerability assessment tools have been developed to analyze the security of software systems and networks using attack graphs. Some of the popular ones are MulVal and A2G2V. However, these tools do not address discovery of unique vulnerability exploits in a software-defined and virtualized network.

[0010] The 5G ecosystem includes multiple protocols executing at different layers. Many vulnerabilities have been detected in various 5G protocols like cellular paging protocols, multiple control layer protocols, and cellular access network protocols. The 5G-AKA

protocol claims to provide higher security than its predecessors because it provides enhanced user identity protection, more sophisticated key derivation, and an increased influence of the home network in authentication. However, the increased complexity of the 5G-AKA protocol leads to new vulnerabilities.

SUMMARY OF THE INVENTION

[0011] According to various embodiments, a method for detecting security vulnerabilities in a fifth generation core network (5GCN) is disclosed. The method includes constructing an attack graph from a plurality of regular expressions. Each regular expression corresponds to a sequence of system level operations for a known 5GCN attack. The attack graph includes a plurality of nodes and a plurality of paths. Each node represents a system-level operation of the 5GCN and each path represents a 5GCN attack vector. The method further includes performing a linear search on the attack graph to determine unexploited 5GCN attack vectors where path in the attack graph that does not represent a known 5GCN attack vector represents an unexploited 5GCN attack vector. The method also includes applying a trained machine learning module to the attack graph to predict new 5GCN attacks. The trained machine learning module is configured to determine a feasibility of linking unconnected nodes in the attack graph to create a new branch representing a new 5GCN vulnerability exploit.

[0012] According to various embodiments, a system for detecting security vulnerabilities in a fifth generation core network (5GCN) is disclosed. The system includes one or more processors configured to construct an attack graph from a plurality of regular expressions. Each regular expression corresponds to a sequence of system level operations for a known 5GCN attack. The attack graph includes a plurality of nodes and a plurality of paths. Each node represents a system-level operation of the 5GCN and each path represents a 5GCN attack vector. The processors are further configured to perform a linear search on the attack graph determine unexploited 5GCN attack vectors, where a path in the attack graph that does not represent a known 5GCN attack vector represents an unexploited 5GCN attack vector. The processors are also configured to apply a trained machine learning module to the attack graph to predict new 5GCN attacks. The trained machine learning module is configured to determine a feasibility of linking unconnected nodes in the attack graph to create a new branch representing a new 5GCN vulnerability exploit.

[0013] According to various embodiments, a non-transitory computer-readable medium having stored thereon a computer program for execution by a processor configured to

perform a method for detecting security vulnerabilities in a fifth generation core network (5GCN) is disclosed. The method includes constructing an attack graph from a plurality of regular expressions. Each regular expression corresponds to a sequence of system level operations for a known 5GCN attack. The attack graph includes a plurality of nodes and a plurality of paths. Each node represents a system-level operation of the 5GCN and each path represents a 5GCN attack vector. The method further includes performing a linear search on the attack graph to determine unexploited 5GCN attack vectors where path in the attack graph that does not represent a known 5GCN attack vector represents an unexploited 5GCN attack vector. The method also includes applying a trained machine learning module to the attack graph to predict new 5GCN attacks. The trained machine learning module is configured to determine a feasibility of linking unconnected nodes in the attack graph to create a new branch representing a new 5GCN vulnerability exploit.

[0014] Various other features and advantages will be made apparent from the following detailed description and the drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

[0015] In order for the advantages of the invention to be readily understood, a more particular description of the invention briefly described above will be rendered by reference to specific embodiments that are illustrated in the appended drawings. Understanding that these drawings depict only exemplary embodiments of the invention and are not, therefore, to be considered to be limiting its scope, the invention will be described and explained with additional specificity and detail through the use of the accompanying drawings, in which:

[0016] Figure 1 depicts a NFV reference architecture according to an embodiment of the present invention;

[0017] Figure 2 depicts a SDN architecture according to an embodiment of the present invention;

[0018] Figure 3 depicts network slicing for 5G networks with SDN/NFV and its attack surface according to an embodiment of the present invention;

[0019] Figure 4 depicts a table of basic operations in regular expressions according to an embodiment of the present invention;

[0020] Figure 5 depicts a graph of tuning a regular expression into an execution graph according to an embodiment of the present invention;

[0021] Figures 6(a)-(c) depict a table of regular expressions for SDN attacks according to an embodiment of the present invention;

[0022] Figures 7(a)-(c) depict an aggregated attack graph of SDN control plane vulnerabilities according to an embodiment of the present invention;

[0023] Figures 8(a)-(c) depict an aggregated attack graph of SDN data plane vulnerabilities according to an embodiment of the present invention;

[0024] Figures 9(a)-(c) depict a table of regular expressions of NFV threat vectors mentioned in standards document ETSI_GS_NFV-SEC_001_v1.1.1 according to an embodiment of the present invention;

[0025] Figures 10(a)-(c) depict an aggregated attack graph of NFV vulnerabilities according to an embodiment of the present invention;

[0026] Figures 11(a)-(b) depict an aggregated attack graph of malicious peripheral based attacks according to an embodiment of the present invention;

[0027] Figure 12 depicts a table of summary of attacks in the attack graphs according to an embodiment of the present invention;

[0028] Figure 13 depicts a table of category-wise examples of novel possible exploits according to an embodiment of the present invention;

[0029] Figure 14 depicts a table of a number of novel possible exploits per category according to an embodiment of the present invention;

[0030] Figure 15 depicts a table of node features according to an embodiment of the present invention;

[0031] Figure 16 depicts a graph of constructing a feature vector and label for a plausible branch according to an embodiment of the present invention;

[0032] Figure 17 depicts a table of number of datapoints per graph according to an embodiment of the present invention;

[0033] Figure 18 depicts a table of NPV (in %) of ML/CSP models according to an embodiment of the present invention;

[0034] Figure 19 depicts a table of the final ensemble model according to an embodiment of the present invention;

[0035] Figure 20 depicts a table of a confusion matrix of the final ensemble model according to an embodiment of the present invention;

[0036] Figure 21 depicts a table of performance metrics of the final ensemble model on the test set according to an embodiment of the present invention;

[0037] Figure 22 depicts a 5G-AKA protocol flow where dashed lines represent secure channels according to an embodiment of the present invention;

[0038] Figure 23 depicts a 5G-AKA session confusion attack flow according to an embodiment of the present invention; and

[0039] Figure 24 depicts a table of property satisfaction under compromised channels and components according to an embodiment of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

[0040] The core network architecture of telecommunication systems has undergone a paradigm shift in fifth generation (5G) networks. 5G networks have transitioned to software-defined infrastructures, thereby reducing their dependence on hardware-based network functions. New technologies, like network function virtualization and software-defined networking, have been incorporated in the 5G core network (5GCN) architecture to enable this transition. This has resulted in significant improvements in efficiency, performance, and robustness of the networks. However, this has also made the core network more vulnerable, as software systems are generally easier to compromise than hardware systems.

[0041] Generally disclosed herein are embodiments for a comprehensive security analysis framework for the 5GCN. A novelty of this approach lies in the creation and analysis of attack graphs of the software-defined and virtualized 5GCN through machine learning. This analysis was able to determine 119 novel possible exploits in the 5GCN. These possible exploits of 5GCN vulnerabilities can generate five novel attacks on the 5G Authentication and Key Agreement protocol. The attacks at the network, protocol, and the application layers are thus combined to generate complex attack vectors. In a case study, these attack vectors are used to find four novel security loopholes in WhatsApp running on a 5G network.

[0042] Prior work in 5G security has referred to broad categories of attacks to which the 5GCN may be vulnerable. Disclosed herein are far-reaching implications of these threats and how they may interact with each other to give rise to complex attacks that were infeasible in previous generations of telecommunication networks. The sequences of operations that are executed to implement an attack, also referred to as attack vectors, can be combined into an attack graph for a concise representation. The various attack vectors pertaining to SDN, NFV, and 5G protocols are combined into attack graphs. These graphs are analyzed to generate 119 novel possible exploits that are exclusive to 5G networks. They are possible exploits in a specific system. The numerous vulnerabilities arising due to implementation errors are generally system specific. It is shown how these possible exploits can compromise the 5G Authentication and Key Agreement (AKA) protocol. Five new attack vectors are discovered in the 5G-AKA protocol that can be triggered by 5GCN vulnerabilities. It is demonstrated

how various attacks at the network and protocol levels can be combined to remotely hack targeted end-user applications.

[0043] In a case study, the hacking of the WhatsApp account of an end user is demonstrated. WhatsApp was chosen as the target application because it is the most widely used instant messaging (IM) platform and possesses some of the most advanced security features. Four security loopholes were discovered that may be triggered in WhatsApp in the absence of appropriate 5GCN security measures.

[0044] It is shown how the framework generally disclosed herein can scale to larger infrastructures through the use of machine learning (ML) and a constraint satisfaction problem (CSP) formulation. ML and CSP formulation are used at the system level to predict possible vulnerability exploits when a new node is added to the attack graphs. A new node may be added when a new vulnerability is discovered or when a new vulnerable component is introduced in the 5GCN. Utilization of ML at the system level is inspired by the SHARKS framework, where ML was used to discover novel possible exploits in an IoT system. SHARKS is an acronym for Smart Hacking Approaches for Risk Scanning and is generally described in PCT Publication No. WO 2020/219157, which is hereby incorporated by reference in its entirety. Although SHARKS was originally targeted at IoT and cyberphysical systems, it is also applicable to the 5GCN architecture, to be described further below.

[0045] **System Overview**

[0046] The vulnerabilities of various disruptive technologies like NFV, SDN, and network slicing are analyzed herein. This section provides a brief description of these concepts and introduces some of the techniques used to analyze system security.

[0047] **NFV:**

[0048] A network comprises various network functions (NFs) like gateways, load balancers, and firewalls. In traditional networks, these NFs are implemented on proprietary hardware systems. Such systems are not flexible and incur high maintenance costs because they are vendor proprietary. Moreover, they often remain underutilized. These issues prevent network operators from improving their average revenue per user (ARPU). NFV provides a way to increase ARPU by reducing network CAPEX and OPEX.

[0049] NFV abstracts out lower-level NF details by implementing NFs on virtual machines (VMs). This facilitates easier adoption of NFs by various applications. In addition, the virtual network functions (VNFs) provide higher flexibility and higher resource utilization. Any network function (such as but not limited to routers, domain name service, firewalls, caching, and network address translation) when implemented through software is

referred to as a VNF. Some nonlimiting categorical examples of VNFs are switching (e.g., BNG, CG-NAT, routers), tunnelling gateway elements (e.g., IPSec/SSL VPN gateways), traffic analysis (e.g., DPI, QoE measurement), signaling (e.g., SBCs, IMS), application-level optimization (e.g., CDNs, load balancers), home routers and set top boxes, mobile network nodes (e.g., HLR/HSS, MME, SGSN, GGSN/PDN-GW, RNC), network-wide functions (e.g., AAA servers policy control, charging platforms), and security functions (e.g., firewalls, intrusion detection systems, virus scanners, spam protection).

[0050] An NFV architecture 10 is shown in Figure 1. In this figure, every layer interacts only with the layers directly above and below it. The rest of the infrastructure is abstracted out. For example, the VNFs interact only with OSS/BSS above and virtual resources below. They do not need to interact directly with any other layer.

[0051] The various components of the NFV architecture 10 are as follows:

[0052] Operations Support System (OSS)/Business Support System (BSS) 12: This is responsible for various network management and operations functions like service provisioning and fault tolerance.

[0053] Network functions virtualization infrastructure (NFVI) 14: This is a distributed system of resources designed to provide a common platform to the VNFs 16. As shown in Fig. 1, the NFVI can be categorized into three classes: virtual resources 18 (including virtual computing 20, virtual storage 22, and virtual networking 24), a virtualization layer 26, and physical resources 28 (including physical computing 30, physical storage 32, and physical networking 34). The virtualization layer 26 acts as a bridge between the physical resources 28 and the virtual resources 18 by primarily managing allocation of physical resources 28 to virtual resources 18.

[0054] NFV Management and Network Orchestration (MANO) unit 36: This includes an NFV orchestrator 38, a VNF manager 40, and a virtual infrastructure manager (VIM) 42. The NFV orchestrator 38 plays an important role in instantiating the network. The VNF manager 40 is responsible for instantiating the VNFs 16. It manages various attributes of the VNFs 16 like their creation, migration, resource allocation, and termination. The VIM 42 is responsible for management and virtualization of the physical compute 30, storage 32, and network resources 34.

[0055] Service, VNF, Infrastructure Description 44: This dataset provides information regarding the VNF deployment template, VNF forwarding graph, service-related information, and NFV infrastructure information models. These templates/descriptors are used internally within the applicable functional block of NFV MANO 36, as needed.

[0056] All the components described above are provided by third-party vendors, unlike pre-5G networks where all components are proprietary. This makes these components inherently untrustworthy. Moreover, third-party software systems cannot be protected by hardware-based fingerprinting mechanisms like hardware root-of-trust and physical unclonable functions.

[0057] SDN:

[0058] Typically, network devices have their functionalities hard coded into the devices. This hinders flexibility and innovation in networks. SDNs ameliorate these issues, make virtualization of networks easier, and have the potential to increase the ARPU of network operators. The primary objective of software-defined networking is decoupling the control and data planes. SDNs have centralized controllers that make forwarding decisions for the switches. The controllers have a broad overview of the entire network, so they can make better decisions than localized switches.

[0059] An SDN architecture 100 is shown in Figure 2. The logically centralized controllers 102 in the control plane 104 receive application requirements of SDN applications 106 (such as operator services 108, management apps 110, monitoring apps 112, and vendor apps 114) through the northbound interface 116. The controllers 102 are responsible for translating the application requirements into efficient flow rules. These rules are relayed to devices 118 of the data plane 120 via the southbound interface 122. The data plane 120 mainly includes forwarding devices like routers and switches. The data plane devices 118 communicate periodically with the controllers 102, updating the controllers 102 with the current situation in the data plane 120. This gives the controllers 102 a global view of the network, thus enabling the controllers 102 to make efficient forwarding decisions.

[0060] Various components of the SDN architecture 100, namely the control plane 104, data plane 120, northbound interface 116, and southbound interface 122, are prone to vulnerabilities.

[0061] Network Slicing for 5G Networks with SDN/NFV:

[0062] Network slicing is an approach for sharing virtual network resources among multiple verticals. A network slice refers to an independent, end-to-end network composed of virtual resources. Network slicing enables the network operators to meet their ambitious goals, like scalability and low latency, by providing better network isolation and increased statistical multiplexing. A network slicing architecture 200 for the 5GCN is depicted in Figure 3.

[0063] There are two kinds of resources available for sharing: NFs and the physical infrastructure. The NFs are provided to the operators by the tenants and the infrastructure by the infrastructure provider (InP). Virtualization and SDN are utilized at both the tenant and InP levels. The SDN controller at the tenant and InP levels are referred to as the tenant controller (TC) 202 and the infrastructure controller (IC) 204, respectively. A simplified example of the implementation of network slicing is depicted in Fig. 3. Every network slice has a network services orchestrator (NSO) 206 that communicates with the resource orchestrator (RO) 208 of the tenant. A tenant provides multiple slices to the operators. In the simplified example depicted in Fig. 3, the tenant is dependent on a single InP for its resources. In reality, the tenant may be dependent on multiple InPs. The network slicing architecture demonstrates how SDN and NFV are used together in the 5GCN.

[0064] Regular Expression:

[0065] Regular expression is a concise representation of a set of strings. Regular expressions are used to represent an attack vector. The set of all permissible characters in a regular expression is called its alphabet, denoted by Σ . The operations in regular expressions that are used herein are described in the table in Figure 4. Regular expressions are generally used to denote system-level operations that are incomprehensible to humans.

[0066] The characters of the regular expression are defined herein at a higher granularity for the sake of generality. The alphabet (Σ) of the regular expressions here includes human-understandable system-level operations. For example, by $\Sigma = \{\text{'Install malicious switch,' 'Insert malware in hypervisor,'} \dots\}$. This is done to ensure that application of the approach herein is independent of the application, OS, or the compiler employed by the 5GCN.

[0067] Methodology

[0068] This section describes the methodology and its impact for embodiments of the comprehensive security analysis framework for the 5GCN. The security of the software-defined and virtualized 5GCN are analyzed using ML and CSP formulation.

[0069] Attack Surface:

[0070] An attack surface of a system refers to the set of various entry points that can be exploited. The various components that compose the attack surface of the 5GCN are depicted in Figure 3. They are as follows: 1) User applications; 2) Northbound interface of SDN controller; 3) SDN controller; 4) Control channel of SDN; 5) VNFs; 6) Tenant; 7) Network slice; 8) NFV MANO unit; 9) Management network between tenant and InP; 10) Hypervisor;

11) InP peripheral attacks, attacks on physical infrastructure. The attack vectors for exploiting vulnerabilities of these components are discussed in detail in the subsequent sections.

[0071] Attack Vector Representation:

[0072] Regular expressions and attack graphs are used to represent various attacks on the 5G system. Regular expressions are used because they allow for representing the sequence of exploits in an exploit chain. Attack graphs are used because they enable efficient modeling of the interactions between different threats. This section describes the process of constructing the attack graphs from various attacks. First, every attack is decomposed into a sequence of system level_operations. This sequence is represented using a regular expression. Then, this regular expression is converted into an attack graph. For example, consider an attack in which a target switch is disconnected from its SDN controller by poisoning the Address Resolution Protocol (ARP). This attack can be executed by the following sequence of system level operations: 1) Install a malicious VM in the system; 2) Launch an ARP poisoning attack to alter the MAC address of the controller on the target switch; 3) In the target switch memory, replace the MAC address of the original controller with that of the malicious VM; 4) The target switch is now disconnected from the controller; and 5) Send malicious flow rules to the switch from the malicious VM. This disrupts network functionalities.

[0073] Let ch denote a character from the alphabet Σ of the regular expressions. Then, the regular expression of the attack vector described above can be represented as: $ch_i(\text{Install malicious VM})$; $ch_j(\text{ARP poisoning})$; $ch_k(\text{Impersonate controller in switch})$; $ch_l(\text{Disconnect switch from controller})$; $ch_m(\text{Crash network})$. This regular expression can be converted into an execution graph, as shown in Figure 5. The execution graphs of multiple attacks are combined to obtain the aggregated attack graphs.

[0074] 5GCN Vulnerability Analysis:

[0075] This section describes the vulnerabilities of NFV, SDN, and peripheral devices, and the threats that arise from them. For each of these domains, regular expressions and attack graphs are used to conduct a complete security analysis of the system.

[0076] SDN Vulnerability Analysis:

[0077] SDN is one of the most disruptive technologies that is deployed in 5G systems. SDN implementation contains multiple vulnerabilities and is prone to exploits of varying complexity, including topology poisoning attacks, controller hijacking attacks, man-in-the-middle (MiTM) attacks, and denial-of-service (DoS) attacks, as nonlimiting examples. Moreover, popular open-source NOSs for the SDN controller, namely OpenFlow, OpenDaylight, and POX, have been shown to be vulnerable to multiple attacks.

[0078] SDN vulnerabilities can be broadly divided into two categories: control plane and data plane attacks. Control plane attacks involve compromising the NOS or the control channel that is used to send the control messages to the data plane devices. The communication in the control channel is generally unencrypted to enhance performance. This is a potential security loophole. An adversary with access to the control channel can possibly eavesdrop on the control messages to infer the network topology. Knowledge of the network topology can lead to a variety of attacks. Moreover, an adversary can compromise the integrity of the control messages without being detected. This can cause malicious network reconfiguration and DoS attacks. The data plane is also vulnerable to various attacks. The data plane attacks generally target individual switches and forwarding devices.

[0079] All SDN attack vectors are represented as regular expressions and then they are converted into attack graphs. The regular expressions of various SDN attacks are shown in the table in Figures 6(a)-(c). These attack vectors in the SDN control plane (SDN-CP) and the SDN data plane (SDN-DP) are then concisely represented as the attack graphs shown in Figures 7(a)-(c) and Figures 8(a)-(c), respectively. In an attack graph, every path from a head node to a tail node is a unique attack vector. The graph in Figures 7(a)-(c) has 14 unique SDN-CP attack vectors and the graph in Figures 8(a)-(c) has 25 unique SDN-DP attack vectors.

[0080] NFV Vulnerability Analysis:

[0081] NFV provides a dynamic and loosely coupled infrastructure that caters to a large diversity of user requirements. However, NFV inherits multiple implementation vulnerabilities and exploits thereof. Prior to NFV, when a proprietary function was introduced in the network, there existed an established trust between the developer and the operator. This trust is absent in an NFV-enabled network architecture because third-party VNFs are usually susceptible to a variety of threats, such as: 1) Generic networking threats; 2) Generic virtualization threats; and 3) Emerging threats due to a combination of networking and virtualization.

[0082] Due to multi-tenancy and Infrastructure-as-a-Service paradigms of virtualization, access to the core network is easier than before. This makes the 5GCN vulnerable to different kinds of attackers, some of whom may be end customers of retail networks, retail network operators, wholesale network operators, hypervisor operators, infrastructure sharers and operators, and facility managers. Hence, security monitoring should be an integral part of the 5GCN ecosystem.

[0083] Regular expressions of the NFV threats and vulnerabilities are described in the table in Figures 9(a)-(c) and are concisely represented in the attack graph shown in Figures 10(a)-(c). The attack graph has 25 unique NFV attack vectors. These attack vectors have been constructed from the ETSI NFV security problem statement.

[0084] Malicious Peripherals:

[0085] The 5GCN is vulnerable to malicious peripheral devices that can potentially compromise the virtualization infrastructure. Input/Output (IO) attacks involve malicious peripherals that make root-level read and write accesses to the DRAM or to the memory embedded in other peripherals. Various attacks involve corrupting the Peripheral Component Interconnect (PCI) to install rootkits, exploiting Message Signal Interrupts (MSI) and VGA driver vulnerabilities for privilege escalation on hypervisors, and overwriting root-table entries to gain kernel privileges. A concise representation of these attacks is shown in the attack graph in Figures 11(a)-(b). This attack graph includes 49 attack vectors introduced by malicious peripherals and attacks on physical infrastructure of InP.

[0086] The vulnerabilities mentioned in this section require physical access to the infrastructure. Hence, they are less likely to be exploited than NFV and SDN vulnerabilities. However, such attacks are quite common, and their impact is often catastrophic. Thus, it is necessary to take precautions against such attacks while designing the system.

[0087] Graph Analysis:

[0088] The attack vectors in the graphs are constructed from SDN vulnerabilities, NFV vulnerabilities, and IO vulnerabilities. 113 attack vectors were found total in the four aggregated attack graphs. This is summarized in the table in Figure 12.

[0089] It is observed that many of the unconnected nodes in these graphs can be linked together to generate new possible exploits. This section analyzes the feasibility of connections among the unconnected nodes. A link or a branch is deemed to be feasible if the control/data flow represented by that branch is feasible in a real-world system. For example, nodes '*Exploit test backdoors*' and '*Access sensitive information*' can be connected because sensitive credentials of a resource can be accessed through backdoors. On the other hand, nodes '*Compromise hypervisor*' and '*Flood management ports*' cannot be connected because there is a lack of a direct causal relationship between the two.

[0090] Connecting a pair of nodes leads to a new directed branch in the graph. A new branch is interpreted as a novel possible exploit of an existing vulnerability. There are two categories of novel possible exploits in this analysis:

[0091] Intra-graph: These possible exploits are restricted to one of the four domains, namely SDN-CP, SDN-DP, NFV, and malicious peripherals. For example, when two nodes are connected in Figures 7(a)-(c), a novel possible exploit is generated in the SDN-CP.

[0092] Inter-graph: These possible exploits involve the combination of vulnerabilities of multiple attack graphs. For example, when a node in Figures 8(a)-(c) is connected to a node in Figures 10(a)-(c), it leads to a novel possible exploit that combines vulnerabilities of the SDN-DP with that of the NFV infrastructure.

[0093] Some novel possible exploits are demonstrated in the table in Figure 13. The number of novel possible exploits per category are stated in the table in Figure 14.

[0094] ML Analysis:

[0095] When the number of components in the 5GCN increases, the size of the attack graphs increases significantly due to the quadratic dependence of the size of the search space on the number of nodes in the graph. To add a new node to these graphs, every possible connection between the new node and the existing nodes has to be analyzed, which significantly hinders scalability of this framework. To overcome this obstacle, ML and CSP formulation are employed to predict the possible connections of a new node in the graphs.

[0096] Feature Engineering:

[0097] Feature engineering is a necessary pre-processing step for using an ML or CSP model. Every possible branch in the graphs has to be represented by a feature vector for it to be processed by the ML or CSP model. The feature vectors of a branch are generated by implementing the following sequence of steps: 1) Assign feature values for individual nodes; and 2) Combine the feature vectors of the constituent nodes of a branch. Various attributes (features) are assigned to the nodes of the attack graph(s) depending on the layer(s) at which it is executed, the type of impact the attack would have on the system and network, and its position in the graph(s).

[0098] The set of features that are used herein includes the following: application layer, controller, application controller interface, VNF, network infrastructure, management layer, hypervisor, flooding (DoS), access control, data plane, side-channel analysis (SCA), control channel, sensitive information, SDN-CP, SDN-DP, NFV, malicious peripheral, head, and tail. 1 is assigned to the features that are related to the node and 0 is assigned to the others. For example, the features of nodes '*Install malicious apps*' and '*Assign low memory to VM*' are demonstrated in the table in Figure 15. It can be observed that the feature vectors of these two nodes are $\{1,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,1,0\}$ and $\{0,0,0,1,1,1,1,0,0,0,0,0,0,0,0,1,0,0\}$.

[0099] A branch of the graph is represented with an ordered pair of the source and destination nodes, i.e., (source, destination). The feature vector of a branch is obtained by the ordered concatenation of the feature vectors of the source and destination nodes, as shown in Figure 16. This feature vector constitutes a datapoint for the disclosed ML/CSP model. A positive label (equal to 1) or a negative label (equal to -1) is assigned to this datapoint if the branch is feasible or infeasible, respectively.

[0100] All plausible branches are classified into positive and negative examples. The positive examples also include the existing branches in the attack graphs. The dataset for each graph is split into a training set and a test set. The training set is used to train the model and the test set is used to evaluate it. The training set has 85% of the data while the test set has the remaining. The table in Figure 17 shows the number of instances in the training and test sets for each of the graphs.

[0101] Analysis with a CSP Formulation:

[0102] A CSP formulation requires creating a set of constraints on the features of the data instances, such that any feature that satisfies all the constraints represents a feasible exploit. To obtain a CSP formulation based on the dataset herein, a set S is generated that contains the feature vectors of all the positive examples in the training set. For prediction, the feature vector of the test instance is checked for whether belongs to S . If it does, a positive label is assigned to it; else, a negative one.

[0103] ML Models:

[0104] Multiple ML models are trained on the data herein and the best performing ones are chosen for the final ensemble model. The performance of these models is reported in the next section. In this section, the various ML models are briefly described.

[0105] Naive Bayes: Naive Bayes is a probabilistic ML algorithm based on Bayes theorem. The Naive Bayes model assumes that features are independent of each other, given the label. Let the class label be denoted by y and the input features $\{x_1, x_2, \dots, x_n\}$. Assuming feature independence, the probability of label y can be calculated as:

$$[0106] \quad P(y|x_1, x_2, \dots, x_n) = \frac{P(y) \prod_{i=1}^n P(x_i|y)}{\prod_{i=1}^n P(x_i)} \quad (1)$$

[0107] The class label with the highest conditional probability is assigned to a test instance, i.e., $prediction = \operatorname{argmax} P(y|x_1, x_2, \dots, x_n)$. The probabilities $P(x_i|y)$, $P(x_i)$, and $P(y)$ can be obtained by constructing a frequency table of the features from the training data. Herein, the Gaussian Naïve Bayes classifier was used, where the likelihood of the features is assumed to be a Gaussian distribution.

[0108] Decision Tree: A decision tree classifier uses a decision tree to assign class labels. A decision tree can be expressed as a logical expression composed of ‘AND’ and ‘OR’ boolean operators. The leaf nodes of the tree represent the class labels. The other nodes represent conditional tests on the data attributes. Edges between two nodes represent control flow transition that depends on the outcome of the conditional test at the source node.

[0109] k-Nearest Neighbors (k-NN): The k-NN algorithm assigns a datapoint to the most popular class label among its k ($k \geq 1$) nearest neighbors. $k = \{1,2,3,4,5,6,7\}$ was used herein. It was observed that the performance on the data initially increased with an increase in k until $k = 3$. Then, it either stopped increasing or started decreasing. Hence, $k = 3$ was chosen.

[0110] Support Vector Machine (SVM): For an n -dimensional dataset, SVM constructs an $(n - 1)$ -dimensional separating hyperplane that serves as the decision boundary. SVM can generate nonlinear decision boundaries with the help of kernel transformations based on a quadratic optimization algorithm. Various parameters of the SVM model were adjusted and it was observed that the most effective kernel for the data was the radial basis function. The class imbalance effects are mitigated through data preprocessing.

[0111] Artificial Neural Network (ANN): ANNs are loosely modeled after the biological neurons in the brain. An ANN variant called the multi-layer perceptron (MLP) is used. The neurons in a MLP model are arranged in multiple layers. Every neuron receives signals from all the neurons in its previous layer. All these signals are weighted by their corresponding edge weights and their weighted sum is passed through a nonlinear activation function. This output is then propagated to all the neurons in the next layer. The training process involves updating the edge weights so that the prediction error is minimized. Here, two-layer MLPs with the ReLU activation function yield the best results.

[0112] Performance Results:

[0113] This section compares the performance of various ML algorithms on the data. The Negative Predictive Value (NPV) is used to evaluate the ML models. NPV is defined as the fraction of correct negative predictions, as shown in Equation (2). The models are combined with highest NPVs to obtain the final ensemble model. The ensemble model is evaluated with additional metrics like precision, recall, F1 score, Matthew’s correlation coefficient (MCC), and classification accuracy.

$$[0114] \quad NPV = \frac{\text{True Negative}}{\text{True Negative} + \text{False Negative}} \quad (2)$$

[0115] The framework herein is configured in such a way that a security analyst, who uses the framework, can trust the negative predictions of the model with high confidence to be infeasible exploits. Then, the analyst only needs to manually examine the positive predictions for possible exploits. This significantly reduces the amount of manual effort needed. The NPVs of the models are shown in the table in Figure 18.

[0116] A stratified three-fold cross validation is used for evaluation of the models. Stratified cross validation ensures that each fold has an equal ratio of positive and negative labels.

[0117] The dataset is quite imbalanced with a much higher fraction of negative examples. To mitigate its impact, the positive examples are resampled n times, where the value of n changes for different algorithms. The value of n varies between 3 and 12. It is observed that this is highly effective for all the ML models, except k-NN.

[0118] The models with the highest NPVs (as shown in Figure 18) are selected and combined into an ensemble model. For the intergraph dataset, although the Naive Bayes and k-NN ($k = 3$) models have perfect NPV values, these models are not selected. This is because Naive Bayes and k-NN have very low precision values of 0.005 and 0.008, respectively, on this dataset. This overshadows their perfect NPV scores. The final ensemble model is shown in the table in Figure 19. The numbers in the parentheses indicate the number of neurons in the two hidden MLP layers.

[0119] In the table in Figure 20, the confusion matrix of the final ensemble model is shown. The confusion matrix reports the true positives (TP), false positives (FP), false negatives (FN), and true negatives (TN).

[0120] In Figure 20, it is observed that there are 93 positive predictions. The disclosed framework can reduce the search space of manual analysis to 93 instances from the original search space of 1625 instances. This is a 94.3% reduction in manual effort. Manual examination of these 93 instances leads to the discovery of the 26 true positives as novel possible exploits. The drawback of using the ML/CSP approximation is that it fails to include the 15 false negatives in the search space, thus missing the detection of 15 novel possible exploits.

[0121] The final ensemble model is evaluated with the following metrics.

[0122] Precision: Precision is defined as:

[0123]
$$Precision = \frac{TP}{TP+FP} \quad (3)$$

[0124] A higher precision implies a lower FP. This implies that smaller manual effort is devoted to manually examining infeasible exploits, thus resulting in higher automation efficiency.

[0125] Recall: Recall of a model is defined as:

$$[0126] \quad Recall = \frac{TP}{TP+FN} \quad (4)$$

[0127] A high recall value enables the user of the framework to discard the negative predictions from the manual examination set with high confidence. This requires the model to have a minimal FN.

[0128] F1 Score: F1 Score is the harmonic mean of precision and recall and is defined as:

$$[0129] \quad F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (5)$$

[0130] F1 score aims to strike a balance between precision and recall. It is a useful metric when there is an uneven class distribution.

[0131] MCC: MCC is a measure of quality of a binary prediction algorithm. It performs well even for imbalanced classes. It return a value between -1 and +1. -1 corresponds to a complete disagreement between observation and prediction, 0 corresponds to random guessing, and +1 corresponds to a perfect prediction system. It is defined as:

$$[0132] \quad MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \quad (6)$$

[0133] Accuracy: Accuracy portrays the overall performance of the framework. It is defined as:

$$[0134] \quad Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \quad (7)$$

[0135] The various performance metrics of the final ensemble model are presented in the table in Figure 21. A high NPV and accuracy are obtained. However, the precision, recall, and F1 scores are not as impressive as the NPV and accuracy. This shows that there is a trade-off among the various metrics. If the final ensemble model is constructed with a high F1 score, then the NPV suffers. This is a trade-off that has to be made by the security analyst. Since NPV was prioritized here, the precision and F1 scores suffer.

[0136] **5G-AKA Example:**

[0137] The novel possible exploits of the 5GCN reported in the previous sections can lead to novel attacks at the higher layers of the network or increase the ease of execution of existing attacks in the protocol and application layers. In this section, the impact of 5GCN vulnerabilities on the protocol layer is demonstrated.

[0138] Authentication key agreement (AKA) is used in telecommunication networks to establish a secure and authenticated connection between the subscribers and service providers. It enables sharing of a secret key between the user and service provider that is used to secure all further communication.

[0139] The AKA protocols have evolved through generations of telecommunication networks. Today, the most widely used authentication mechanism in such networks is the 4G-AKA. The 3GPP Consortium has configured 5G-AKA to provide superior privacy and security guarantees than 4G-AKA. However, it has been shown that multiple 4G-AKA vulnerabilities still persist in 5G-AKA. 5G-AKA is also vulnerable to novel attacks that were not possible in previous generations of networks. The 5G-AKA protocol can be easily compromised if the 5GCN is vulnerable. This section analyzes the implications of the novel possible exploits on 5G-AKA security.

[0140] 5G-AKA Protocol:

[0141] The 5G-AKA protocol 300 authenticates a user equipment (UE) 302, a serving network (SN) 304, and a home network (HN) 306 to each other. It is a challenge-response based protocol where the UE 302 is authenticated as a legitimate user only if it succeeds in providing the expected response to a challenge provided by the HN 306. Unlike previous networks, the identity of the UE 302, called subscriber permanent identifier (SUPI) in 5G networks, is not sent directly. In 5G networks, the UE 302 sends a subscriber concealed identifier (SUCI) that prevents international mobile subscriber identity catcher attacks.

[0142] Often, the SN 304 and HN 306 are the same network. However, sometimes they are different. For example, when a UE 302 is roaming, its SN 304 is different from its HN 306. In this analysis, a separate SN 304 and HN 306 are considered because this scenario is more prone to attacks. The primary network functions involved in 5G-AKA are the Authentication Server Function (AUSF) 308, Authentication Credential Repository and Processing Function (ARPF) 310, and Security Anchor Function (SEAF) 312. A simplified outline of the 5G-AKA protocol 300 is shown in Figure 22. The details of the messages are abstracted for simplicity. AV denotes the authentication vector, XRES denotes the expected response from the UE, and HXRES denotes a hash of XRES.

[0143] Threat Model:

[0144] Most of the security analysis of the 5G-AKA protocol 300 so far has considered a threat model where the adversary has access to the UE 302 and the communication channels between different networks. The core network infrastructure is considered to be inaccessible

to the adversary. From Figure 22, it can be seen that the AUSF 308 and ARPF 310 communicate over a secure network because they belong to the same network (HN) 306.

[0145] However, it was shown previously herein that the internal components of the 5GCN can be compromised. To overturn the assumption of having an impenetrable 5GCN, the attack surface of the 5G-AKA protocol 300 is expanded for the analysis. In the threat model here, an adversary can compromise the network's private channels and the network functions as well. In previous sections, it was demonstrated how VNFs and other network components can be compromised by exploiting vulnerabilities of SDN, NFV, and IO peripherals.

[0146] 5G-AKA Security Analysis:

[0147] This section analyzes the implications of a compromised 5GCN on the security properties of the 5G-AKA protocol 300. A compromised 5GCN leads to unique exploits and also facilitates exploits that were unrealistic before.

[0148] Novel Attacks:

[0149] The vulnerabilities of NFV, SDN, and input/output (I/O) peripherals have a variety of potential consequences at the network level. These consequences include flooding (DoS) attacks, termination of sensitive VNFs, passive MiTM attacks (like eavesdropping), hijacking of VNFs, and active MiTM attacks (like modification of in-flight traffic). This section analyzes how these consequences can be exploited to compromise the 5G-AKA protocol 300.

[0150] Flooding attacks: The 5G-AKA protocol 300 is vulnerable to session confusion attacks triggered by a race condition in the AUSF-ARPF channel. This attack is demonstrated in Figure 23. When the ARPF 310 receives multiple authentication requests in parallel, it sends the AVs for all the requests to the AUSF 308 at the same time. This leads to a race condition in which the AUSF 308 is unable to distinguish which AV belongs to which UE 302. Thus, there is a high probability that the AUSF 308 sends the wrong credentials to the users. This is a probabilistic attack whose success rate can be increased with more parallel authentication requests from the adversary.

[0151] In the threat model of this attack, the adversary can hijack the VNFs on the SN 304 but not the VNFs on the HN 306. They can hijack the SEAF 312 on the SN 304 and use it to bombard the AUSF 308 with multiple network packets of SUCI(Attkr) simultaneously. The AUSF 308 generates authentication requests, *Auth. info. request (Attkr)*, for all of these packets and sends them to the ARPF 310. When the ARPF 310 receives all these packets simultaneously, along with *Auth. info. request (Victim)*, it leads to a race condition.

According to the 5G-AKA protocol specifications, the response of the ARPF 310 does not include the identity of the UE 302. Thus, the simultaneous reception of multiple (AV, XRES) pairs by the AUSF 308 causes a session confusion. It is probable that the AUSF 308 forwards the AV of the victim to the adversary as a result of this confusion. Now, the adversary can authenticate themselves as the victim.

[0152] The probability of success of this attack is $(1 - \frac{1}{n})$, where n depicts the total number of simultaneous authentication requests received by the ARPF 310. In Figure 23, the value of n is 2; thus, the probability of attack success is 0.5. Increasing the number of simultaneous authentication requests from the adversary's UEs 302 (by replay/flooding attacks from a compromised SEAF 312) will increase n , thus increasing the probability of attack success. The node "Flood VNF with requests/high priority requests" of the NFV attack graph in Figures 10(a)-(c) can be implemented via multiple possible exploits discovered by the disclosed framework to execute this attack.

[0153] Termination of sensitive VNFs: NFV and IO vulnerabilities can be exploited to forcibly terminate targeted VNFs. This can be achieved by executing one of the following nodes in Figures 10(a)-(c): "Shutdown sensitive VNFs," "VNF crash," "DoS on target VNF." The disclosed framework predicts multiple possible exploits for implementing these nodes in a vulnerable 5GCN. Untimely termination of SEAF 312, AUSF 308, or ARPF 310 disrupts the 5G-AKA protocol 300. Although the adverse effects of such attacks can be mitigated by a fault-tolerant implementation of these functions, all ongoing authentication information is lost. This forces the UEs 302 to restart the 5G-AKA protocol 300.

[0154] Passive MiTM: Passive MiTM can be executed on the AUSF 308-ARPF 310 channel. Since this channel is considered to be secure by the 5G-AKA designers, it is not required to be encrypted. Operators would also prefer having no encryption to boost performance. In the analysis of attack graphs disclosed herein, multiple attack vectors were generated for launching privilege escalation attacks that give access to 5GCN resources. An adversary with access to the 5GCN infrastructure can eavesdrop on the secure channels. This leads to the disclosure of private information like AV, XRES, SUPI, and the secret keys of AUSF 308 and SEAF 312 to the adversary. The adversary can exploit the knowledge of XRES and SUPI to authenticate themselves on behalf of a legitimate UE 302. The secret key of AUSF 308 can be exploited to authenticate a fake base-station, thereby launching active MiTM attacks on UEs 302.

[0155] Hijacking of VNFs: Hijacking of sensitive VNFs like the SEAF 312, AUSF 308, or ARPF 310 can cause the 5G-AKA protocol 300 to prevent authentication of legitimate UEs 302 or authenticate adversaries with the credentials of a legitimate UE 302. The methodology disclosed herein demonstrates multiple access control and privilege escalation attacks in the graphs that can be exploited to hijack VNFs.

[0156] Active MiTM: Active MiTM attacks involve modifying the packets during transit. This compromises the integrity of network packets. Since the connections in the same network are assumed to be secure in the original 5G-AKA threat model, the operators are not required to have integrity checks on intra-network messages. The adversary can get access to the internal network by exploiting certain infrastructure vulnerabilities and modify the packets in transit. The methodology disclosed herein demonstrates multiple attack vectors for launching active MiTM attacks. The attack graphs in Figures 7(a)-(c), 8(a)-(c), and 11(a)-(b) demonstrate that there are multiple openings for MiTM attacks in a vulnerable 5GCN. The adversary can exploit them to modify the AV, XRES, AUSF secret key or SUPI in the AUSF 308-ARPF 310 channel without being detected. Modifying the SUPI or XRES will enable the adversary to authenticate themselves on behalf of a legitimate UE 302. Modifying the AUSF 308 secret key enables the user to launch a fake base station.

[0157] 5G-AKA Property Violations:

[0158] The 3GPP Consortium has detailed the security requirements of 5G system components in TS 33.501 v0.7.0. The security requirements that are related to the 5G-AKA protocol 300 can be expressed concisely through two secrecy properties and seven authentication properties. Every vulnerability of the 5G-AKA protocol 300, including the ones mentioned previously herein, violates at least one of these security properties. Hence, analyzing these properties may provide insights into what kinds of attacks are possible.

[0159] The secrecy properties of 5G-AKA are:

[0160] S1. The long-term secret key of the UE 302 should be unknown to the adversary.

[0161] S2. The adversary should not have access to the secret keys of AUSF 308 and SEAF 312.

[0162] The authentication properties of 5G-AKA are:

[0163] A1. SN 304 and UE 302 must agree on the identity of UE 302.

[0164] A2. UE 302 and SN 304 must agree on the identity of SN 304.

[0165] A3. HN 306 and SN 304 must agree on the identity of UE 302.

[0166] A4. UE 302 and HN 306 must agree on the identity of HN 306.

[0167] A5. UE 302 and HN 306 must agree on the identity of SN 304.

[0168] A6. UE 302, HN 306, and SN 304 must agree on the anchor key of SEAF 312, KSEAF.

[0169] A7. UE 302, HN 306, and SN 304 must agree that an anchor key KSEAF instance is not used more than once.

[0170] The security of the 5G-AKA protocol 300 is compromised if any of the aforementioned properties is violated. It has been shown that the compromise of participating components of the 5G-AKA protocol 300 leads to the violation of these properties. The consequences of compromising the 5GCN on the 5G-AKA properties is demonstrated in the table in Figure 24.

[0171] It is shown that 5GCN vulnerabilities and threats lead to the violation of many of the security properties of the 5G-AKA protocol 300. This demonstrates that 5GCN vulnerabilities also make the 5G-AKA protocol 300 vulnerable.

[0172] WhatsApp Security in 5G Networks Example:

[0173] This section analyzes how various existing and novel possible exploits of a vulnerable 5GCN can lead to targeted attacks in the application layer of the network. WhatsApp application was chosen for the security analysis herein. WhatsApp is the most widely used IM application in the world, with over 1.5 billion users. It is also one of the most secure IM applications, where all communications are end-to-end (E2E) encrypted. This section demonstrates that even WhatsApp can be compromised through network and protocol vulnerability exploits. Various WhatsApp attack vectors that are facilitated by the disclosed methodology include the following.

[0174] Impersonation of the victim via 5G-AKA: As described earlier, the adversary can authenticate himself as the victim during 5G-AKA protocol execution by exploiting any of the following attacks: flooding, passive MiTM, hijacking of VNFs, and active MiTM. Then, the adversary can use the victim's identity to impersonate them on WhatsApp.

[0175] Assisting WhatsApp impersonation through voicemail cracking: During registration of a WhatsApp account, the user can choose to be authenticated by a text message or a call. If the user chooses to be authenticated by a call and fails to receive the authentication call, then the one-time password voice message is saved in voicemail. It has been shown that voicemails can be easily hacked using brute force attacks. This attack has a low probability of being successful in a real-world situation because it requires the victim to either be offline or ignore the authentication call. This obstacle for the adversary can be bypassed by launching a DoS attack on the victim's network infrastructure. The framework described herein generates multiple novel possible exploits to launch a DoS attack on various

components of the 5GCN. Figures 7(a)-(c), 8(a)-(c), 10(a)-(c), and 11(a)-(b) show that DoS attacks can be launched on VNFs, VMs, switches, and SDN controllers. A DoS attack on the network infrastructure will terminate the victim's connection to the 5GCN, thus ensuring that they are offline. Now, the voicemail attack has a much higher probability of being successful.

[0176] Compromising encryption keys: E2E security of WhatsApp can be readily compromised if the adversary gets access to the WhatsApp encryption keys on the device. The WhatsApp keys are stored in a sandbox memory on the smartphone that is only accessible by the WhatsApp application. If an adversary has root privileges on the phone, they can access the WhatsApp encryption keys. Rootkits can be installed on the UE by combining MiTM attacks in our attack graphs with baseband attacks. Attack vectors that exploit rootkit injection attacks are described in Figures 8(a)-(c).

[0177] Lack of certificate pinning: WhatsApp does not implement certificate pinning on the UE. This makes the WhatsApp clients vulnerable to MiTM attacks through certificate proxying. The possible exploits for launching an MiTM attack at the network level were described earlier herein. These attacks can be executed in the absence of certificate pinning.

[0178] **Conclusion**

[0179] The attack graphs depicted in Figures 7(a)-(c), 8(a)-(c), 10(a)-(c), and 11(a)-(b) are configured to be as exhaustive as possible, though this is not intended to be limiting. All possible attack classes applicable to SDN, NFV, and malicious peripherals in a 5GCN were attempted to be included in these graphs. For application of the disclosed framework to a specific 5GCN implementation, 5GCN-specific graphs would be derived from the generalized graphs disclosed herein. For a given 5GCN architecture, the relevant nodes from the generalized graphs are extracted to form the architecture-specific graphs. For example, if a 5GCN does not use LLDP to establish network topology, the LLDP-specific nodes would be eliminated from Figures 7(a)-(c) for this 5GCN. If a 5GCN has a feature that warrants addition of new nodes to the graphs, ML can be used to predict the connections of the new nodes to the existing nodes. Thus, new nodes can be added to the graphs and create a 5GCN-specific attack graph for further analysis.

[0180] Probabilistic attack graphs, more popularly known as Bayesian attack graphs, have been extensively used to assess the security risk of networks. The framework disclosed herein can be extended to Bayesian attack graphs with minimal modifications. In a traditional Bayesian attack graph, each node represents a state of the system. An edge from state A to state B exists if an exploit of a vulnerability at state A takes the system to state B. The weight of this edge is equal to the probability of execution of the aforementioned exploit. Hence, the

graphs presented herein can be transformed into equivalent Bayesian attack graphs if the edges have weights corresponding to their probability of execution. These probabilities can be obtained for specific systems from the common vulnerabilities and exposures (CVE) databases. However, the disclosed framework is more useful than Bayesian attack graphs because it can also discover novel possible exploits in a system.

[0181] 5G communication systems have a huge potential, made possible by the integration of new technologies like NFV and SDN into the 5GCN. This gives rise to new vulnerabilities in the 5G system. As such, generally disclosed herein are embodiments for a system and method to analyze how various vulnerabilities of NFV, SDN, and malicious IO peripherals can interact with each other to compromise the security of the 5GCN. 119 novel possible exploits were discovered by analyzing the underlying patterns in the 113 existing attack vectors in SDN, NFV, and IO peripherals. It was shown that a compromised 5GCN may have devastating consequences on the end user. A compromised 5GCN was shown to trigger five unique types of attacks in the 5G-AKA protocol. These attacks can be further combined with infrastructure vulnerabilities to compromise targeted users at the application layer. This was demonstrated by analyzing four potential security loopholes in the WhatsApp IM application.

[0182] It is understood that the above-described embodiments are only illustrative of the application of the principles of the present invention. The present invention may be embodied in other specific forms without departing from its spirit or essential characteristics. All changes that come within the meaning and range of equivalency of the claims are to be embraced within their scope. Thus, while the present invention has been fully described above with particularity and detail in connection with what is presently deemed to be the most practical and preferred embodiment of the invention, it will be apparent to those of ordinary skill in the art that numerous modifications may be made without departing from the principles and concepts of the invention as set forth in the claims.

CLAIMS

What is claimed is:

1. A method for detecting security vulnerabilities in a fifth generation core network (5GCN), the method comprising:
 - constructing an attack graph from a plurality of regular expressions, each regular expression corresponding to a sequence of system level operations for a known 5GCN attack, the attack graph comprising a plurality of nodes, each node representing a system-level operation of the 5GCN, and a plurality of paths, each path representing a 5GCN attack vector;
 - performing a linear search on the attack graph to determine unexploited 5GCN attack vectors, wherein a path in the attack graph that does not represent a known 5GCN attack vector represents an unexploited 5GCN attack vector; and
 - applying a trained machine learning module to the attack graph to predict new 5GCN attacks, the trained machine learning module configured to determine a feasibility of linking unconnected nodes in the attack graph to create a new branch representing a new 5GCN vulnerability exploit.
2. The method of claim 1, wherein known 5GCN attacks comprise exploiting a vulnerability in at least one of a software defined network (SDN), network function virtualization (NFV), and input/output (I/O) peripheral of the 5GCN.
3. The method of claim 1, wherein new 5GCN attacks comprise exploiting a vulnerability in at least one of a software defined network (SDN), network function virtualization (NFV), and input/output (I/O) peripheral of the 5GCN.
4. The method of claim 1, wherein new 5GCN attacks comprise exploiting a combination of vulnerabilities from at least two of a SDN, NFV, and I/O peripheral of the 5GCN.
5. The method of claim 1, wherein constructing the attack graph further comprises representing each of the plurality of regular expressions as execution graphs.
6. The method of claim 5, wherein constructing the attack graph further comprises combining the plurality of execution graphs into an aggregated attack graph.

7. The method of claim 1, wherein unexploited 5GCN attack vectors are discovered based on a connection of nodes in the attack graph.
8. The method of claim 1, wherein linking unconnected nodes in the attack DAG is feasible when a sequence of operations represented by linking the unconnected nodes can be implemented in the 5GCN.
9. The method of claim 1, wherein the machine learning module comprises at least one of Naïve Bayes, Decision Tree, k-Nearest Neighbors, Support Vector Machines, and Artificial Neural Network.
10. The method of claim 1, further comprising training the machine learning module to predict new 5GCN attacks.
11. The method of claim 10, further comprising constructing a training dataset for training the machine learning module, the training dataset comprising all existing paths in the attack graph as feasible and a plurality of unconnected paths known to be infeasible.
12. The method of claim 11, wherein infeasible branches comprise infeasible sequences of system-level operations.
13. The method of claim 10, wherein the machine learning module is trained based on parameters for achieving higher negative predictive value and accuracy.
14. The method of claim 10, wherein the machine learning module is trained based on parameters for achieving higher F1 score, recall, and precision.
15. A system for detecting security vulnerabilities in a fifth generation core network (5GCN), the system comprising one or more processors configured to:
 - construct an attack graph from a plurality of regular expressions, each regular expression corresponding to a sequence of system level operations for a known 5GCN attack, the attack graph comprising a plurality of nodes, each node representing a

system-level operation of the 5GCN, and a plurality of paths, each path representing a 5GCN attack vector;

perform a linear search on the attack graph to determine unexploited 5GCN attack vectors, wherein a path in the attack graph that does not represent a known 5GCN attack vector represents an unexploited 5GCN attack vector; and

apply a trained machine learning module to the attack graph to predict new 5GCN attacks, the trained machine learning module configured to determine a feasibility of linking unconnected nodes in the attack graph to create a new branch representing a new 5GCN vulnerability exploit.

16. The system of claim 15, wherein known 5GCN attacks comprise exploiting a vulnerability in at least one of a software defined network (SDN), network function virtualization (NFV), and input/output (I/O) peripheral of the 5GCN.
17. The system of claim 15, wherein new 5GCN attacks comprise exploiting a vulnerability in at least one of a software defined network (SDN), network function virtualization (NFV), and input/output (I/O) peripheral of the 5GCN.
18. The system of claim 15, wherein new 5GCN attacks comprise exploiting a combination of vulnerabilities from at least two of a SDN, NFV, and I/O peripheral of the 5GCN.
19. The system of claim 15, wherein the processors are further configured to construct the attack graph by representing each of the plurality of regular expressions as execution graphs.
20. The system of claim 19, wherein the processors are further configured to construct the attack graph by combining the plurality of execution graphs into an aggregated attack graph.
21. The system of claim 15, wherein unexploited 5GCN attack vectors are discovered based on a connection of nodes in the attack graph.

22. The system of claim 15, wherein linking unconnected nodes in the attack DAG is feasible when a sequence of operations represented by linking the unconnected nodes can be implemented in the 5GCN.
23. The system of claim 15, wherein the machine learning module comprises at least one of Naïve Bayes, Decision Tree, k-Nearest Neighbors, Support Vector Machines, and Artificial Neural Network.
24. The system of claim 15, wherein the processors are further configured to train the machine learning module to predict new 5GCN attacks.
25. The system of claim 24, wherein the processors are further configured to construct a training dataset for training the machine learning module, the training dataset comprising all existing paths in the attack graph as feasible and a plurality of unconnected paths known to be infeasible.
26. The system of claim 25, wherein infeasible branches comprise infeasible sequences of system-level operations.
27. The system of claim 24, wherein the machine learning module is trained based on parameters for achieving higher negative predictive value and accuracy.
28. The system of claim 24, wherein the machine learning module is trained based on parameters for achieving higher F1 score, recall, and precision.
29. A non-transitory computer-readable medium having stored thereon a computer program for execution by a processor configured to perform a method for detecting security vulnerabilities in a fifth generation core network (5GCN), the method comprising:
- constructing an attack graph from a plurality of regular expressions, each regular expression corresponding to a sequence of system level operations for a known 5GCN attack, the attack graph comprising a plurality of nodes, each node representing a system-level operation of the 5GCN, and a plurality of paths, each path representing a 5GCN attack vector;

performing a linear search on the attack graph to determine unexploited 5GCN attack vectors, wherein a path in the attack graph that does not represent a known 5GCN attack vector represents an unexploited 5GCN attack vector; and

applying a trained machine learning module to the attack graph to predict new 5GCN attacks, the trained machine learning module configured to determine a feasibility of linking unconnected nodes in the attack graph to create a new branch representing a new 5GCN vulnerability exploit.

30. The non-transitory computer-readable medium of claim 29, wherein known 5GCN attacks comprise exploiting a vulnerability in at least one of a software defined network (SDN), network function virtualization (NFV), and input/output (I/O) peripheral of the 5GCN.
31. The non-transitory computer-readable medium of claim 29, wherein new 5GCN attacks comprise exploiting a vulnerability in at least one of a software defined network (SDN), network function virtualization (NFV), and input/output (I/O) peripheral of the 5GCN.
32. The non-transitory computer-readable medium of claim 29, wherein new 5GCN attacks comprise exploiting a combination of vulnerabilities from at least two of a SDN, NFV, and I/O peripheral of the 5GCN.
33. The non-transitory computer-readable medium of claim 29, wherein constructing the attack graph further comprises representing each of the plurality of regular expressions as execution graphs.
34. The non-transitory computer-readable medium of claim 33, wherein constructing the attack graph further comprises combining the plurality of execution graphs into an aggregated attack graph.
35. The non-transitory computer-readable medium of claim 29, wherein unexploited 5GCN attack vectors are discovered based on a connection of nodes in the attack graph.

36. The non-transitory computer-readable medium of claim 29, wherein linking unconnected nodes in the attack DAG is feasible when a sequence of operations represented by linking the unconnected nodes can be implemented in the 5GCN.
37. The non-transitory computer-readable medium of claim 29, wherein the machine learning module comprises at least one of Naïve Bayes, Decision Tree, k-Nearest Neighbors, Support Vector Machines, and Artificial Neural Network.
38. The non-transitory computer-readable medium of claim 29, wherein the method further comprises training the machine learning module to predict new 5GCN attacks.
39. The non-transitory computer-readable medium of claim 38, wherein the method further comprises constructing a training dataset for training the machine learning module, the training dataset comprising all existing paths in the attack graph as feasible and a plurality of unconnected paths known to be infeasible.
40. The non-transitory computer-readable medium of claim 39, wherein infeasible branches comprise infeasible sequences of system-level operations.
41. The non-transitory computer-readable medium of claim 38, wherein the machine learning module is trained based on parameters for achieving higher negative predictive value and accuracy.
42. The non-transitory computer-readable medium of claim 38, wherein the machine learning module is trained based on parameters for achieving higher F1 score, recall, and precision.

1/26

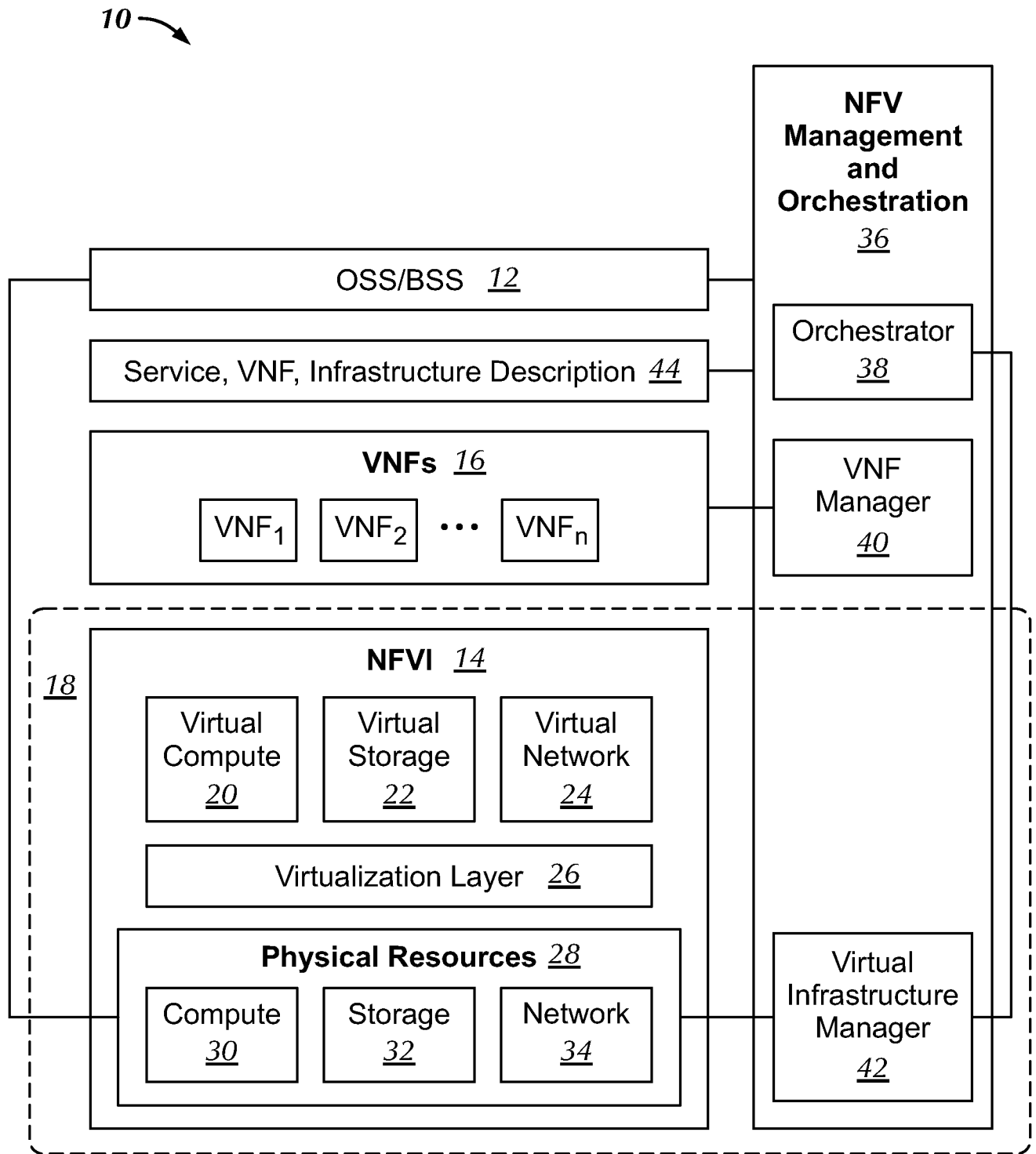


FIG. 1

2/26

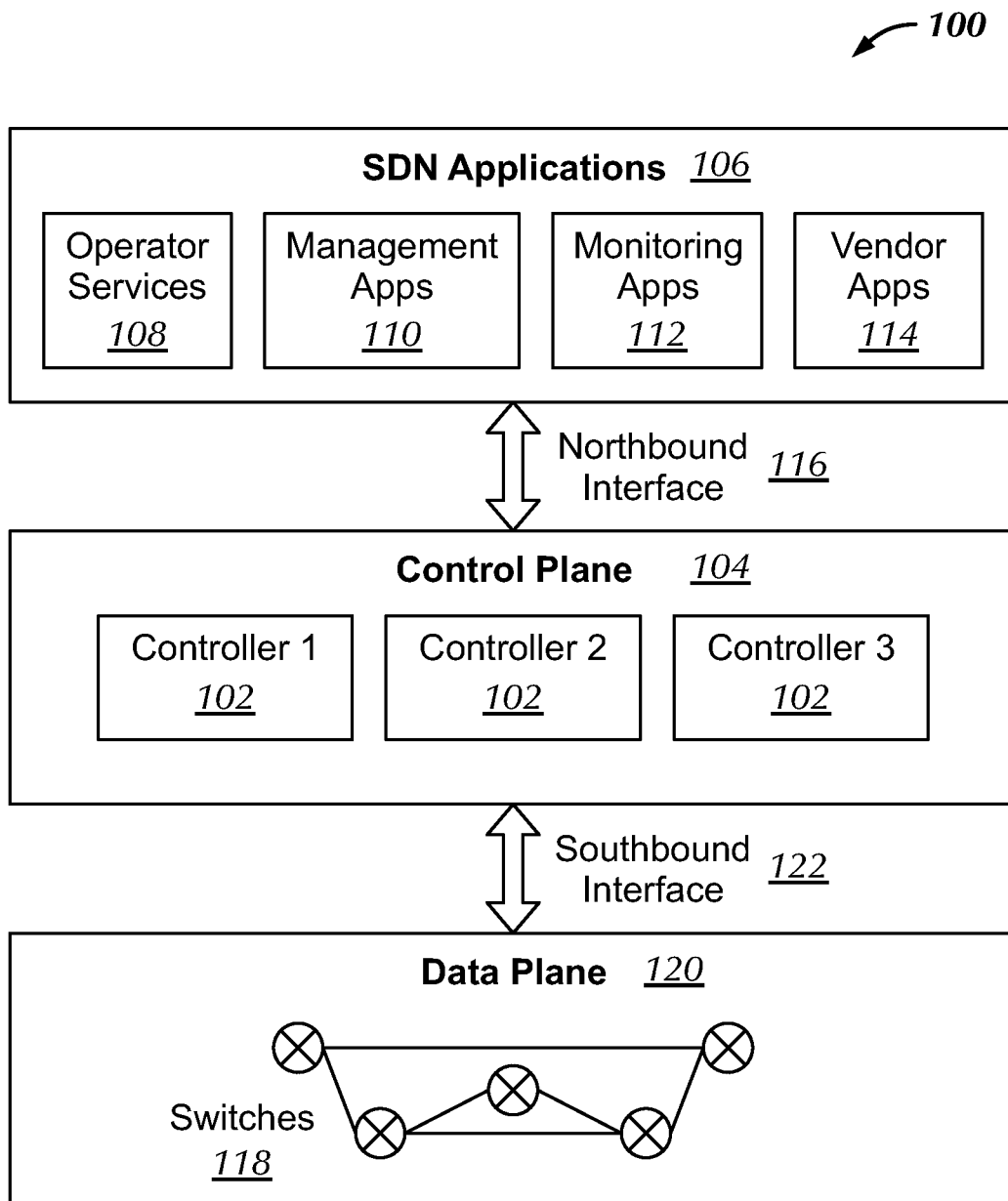


FIG. 2

3/26

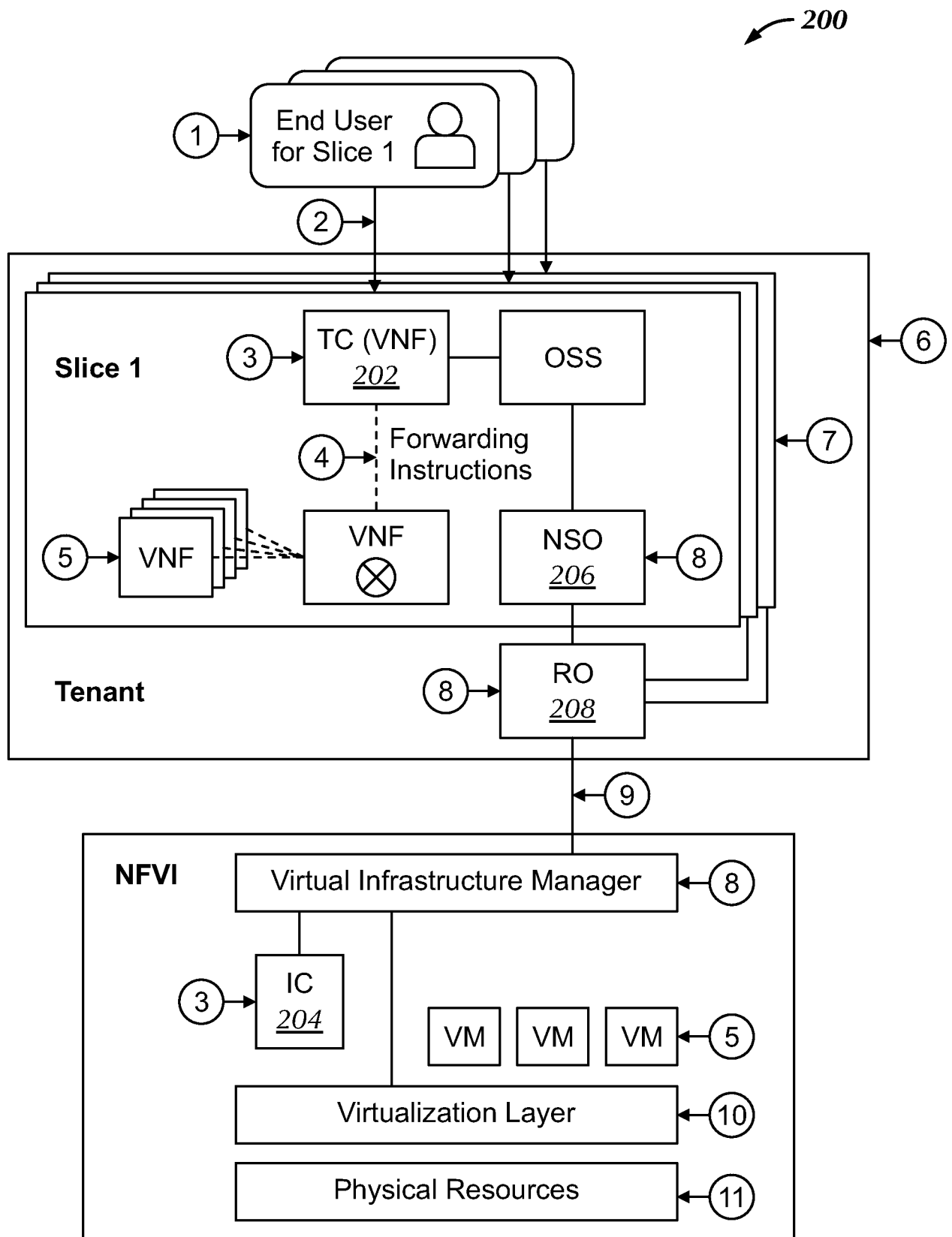


FIG. 3

		$A = \{a, b\}; B = \{c, d\}$
Operation	Definition	Example
Set Union (+)	Set union of two regular expressions	$A + B = \{a, b, c, d\}$
Concatenation (.)	Concatenation of strings of two regular expressions	$A.B = \{ac, ad, bc, bd\}$

FIG. 4

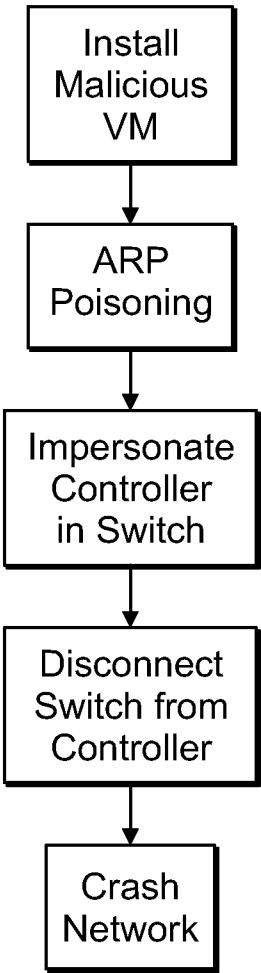


FIG. 5

Application layer	Entry point of attack	Regular expression
Abuse of privileges and authority	Malicious third party apps	ch_i (install malicious app). ch_n (gain control over tenant controller VM). $\{ch_j$ (disconnect sensitive apps) + ch_k (shutdown sensitive apps)). $\{ch_l$ (crash network) + ch_m (degrade network performance)}
Service disruption	Malware	ch_i (install malicious app). ch_j (gain control over tenant controller VM). $\{[ch_1$ (drop control messages to VNFs) + ch_2 (subvert order in which app handlers access control packets) + ch_3 (interfere in service chain)]. ch_k (disrupt control packet forwarding) + ch_4 (eavesdrop on control messages). ch_l (derive topology of network). ch_m (execute topology based attacks)]. $\{ch_o$ (crash network) + ch_p (degrade network performance)}
Application shutdown	Vulnerable northbound API	ch_i (exploit vulnerability in northbound API). $\{ch_j$ (issue system command). ch_k (terminate victim app) + ch_l (eavesdrop on messages between controller and app))
Control layer		
Dynamic flow rule tunneling	Malware & vulnerable switches	ch_i (install malicious app). ch_j (instruct conflicting/overlapping flow rules). ch_k (bypass sensitive VNFs like firewall)/Intrusion Detection Systems (IDS)). $\{ch_l$ (degrade performance) + ch_m (crash network + ch_n (DoS attack))
Controller poisoning (Poisoned network view)	Malware & vulnerable network services and protocols	ch_i (gain access to controller VM). $\{ch_j$ (send crafted LLDP packets). ch_k (poison network topology in controller by adding fake connections). ch_l (drop packets in data plane). ch_m (degrade performance) + ch_n (poison controller host profile reservoir). ch_o (install malicious VM). ch_p (redirect data packets to malicious VM). ch_q (MitM attack in data plane))

FIG. 6(a)

6/26

NOS misuse	Vulnerable controller	$\{ch_i(\text{malicious apps running at Application layer}) + ch_j(\text{rogue switch VM})\}$. (multiple attacks that are denoted in cells below)
		$ch_k(\text{execute system commands})$. $ch_l(\text{terminate controller})$.
		$ch_m(\text{degrade network performance})$
		$ch_k(\text{access sensitive network information})$. $ch_l(\text{execute deviant actions})$
		$ch_k(\text{modify flow rules})$. $\{ch_l(\text{eavesdrop on data plane packets}) + ch_m(\text{redirect data packets})\}$. $\{ch_n(\text{degrade network performance}) + ch_o(\text{MiTM attack in data plane}) + ch_p(\text{bypass security functions like firewalls/IDS})\}$
		$ch_k(\text{install rootkits})$
		$ch_k(\text{hijack network policy database})$
		$ch_k(\text{input invalid input data})$. $ch_l(\text{send controller in an invalid state})$.
		$ch_m(\text{degrade network performance})$
Packet-in-flooding	Faulty controller or compromised switch VMs	$\{ch_i(\text{malicious app}) + ch_j(\text{malicious switch VM})\}$. $ch_k(\text{send massive amounts of malformed packets})$. $ch_l(\text{switch-table misses of switch VM})$. $ch_m(\text{massive amount of packet-in messages sent to controller VM})$. $ch_o(\text{DoS attack on controller})$. $ch_p(\text{degrade network performance})$
Switch table flooding	Faulty controller or compromised switch VMs	$\{ch_i(\text{malicious app}) + ch_j(\text{malicious switch VM})\}$. $ch_k(\text{send massive amount of 'features-reply' messages to controller})$. $ch_l(\text{fill controller switch table with fake switches})$. $ch_m(\text{DoS attack on controller})$. $ch_n(\text{degrade network performance})$
Legitimate switch id hijacking		$ch_i(\text{malicious switch VM installed})$. $ch_j(\text{connect malicious VM to controller using DPIIP of target VM})$. $ch_k(\text{legitimate VM gets disconnected})$. $\{ch_l(\text{network crash}) + ch_m(\text{degrade network performance})\}$

FIG. 6(b)

Spanning tree poisoning	ch_i {send crafted LLDP packets to controller VM}. ch_j {poison spanning tree protocol with targeted fake links}. ch_k {disconnect targetted links}. $\{ch_l$ {network crash} + ch_m {degrade network performance}}	
Control Channel (CC)		
Passive MiTM	Unencrypted messages	ch_i {absence of crypto in CC}. ch_j {sniff packets on CC}. $\{ch_k$ {eavesdrop on control messages} + ch_l {eavesdrop on topology information} + ch_m {eavesdrop on management info}}
Active MiTM	Compromised southbound interface or vulnerable data links	ch_i {absence of crypto in CC}. ch_j {ARP poisoning}. ch_k {insert intruder host between controller and data plane}
Infrastructure layer		
DoS leveraging ARP poisoning	ch_i {ARP poisoning}. ch_j {impersonate controller VM}. ch_k {connect fake controller to target switch}. ch_l {disconnect target switch VM from network}. ch_m {degrade network performance}	
Flow-rule flushing/ modification	$\{ch_i$ {malicious app} + ch_j {install malware on controller VM + ch_o {gain access to controller VM}}. ch_k {send incorrect control messages to switches}. $\{ch_l$ {modify switch flow rules} + ch_m {flush switch flow rules}}. ch_n {degrade network performance}	
Flow-rule flooding	Side-channel attack (SCA)	ch_i {record round-trip time of packets;SCA}. ch_j {detect VM that has an almost full switch-table}. ch_k {detect types of packets causing table misses}. ch_l {send such packets repeatedly}. ch_m {flood switch table of VM}. ch_n {degrade network performance}

FIG. 6(c)

8/26

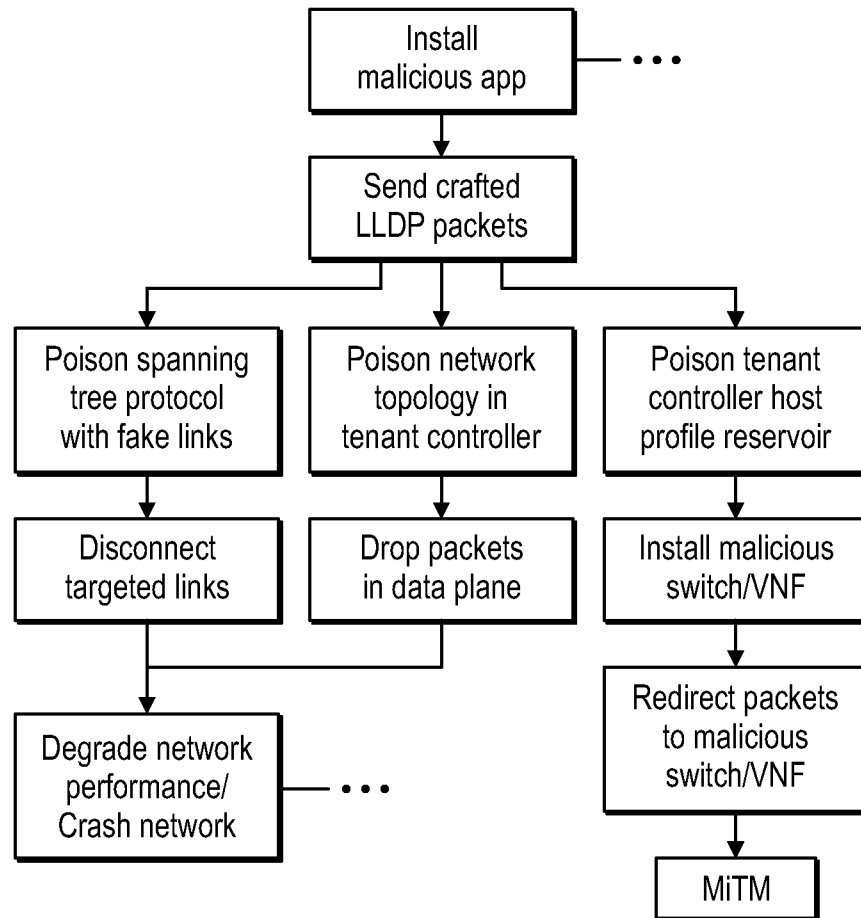


FIG. 7(a)

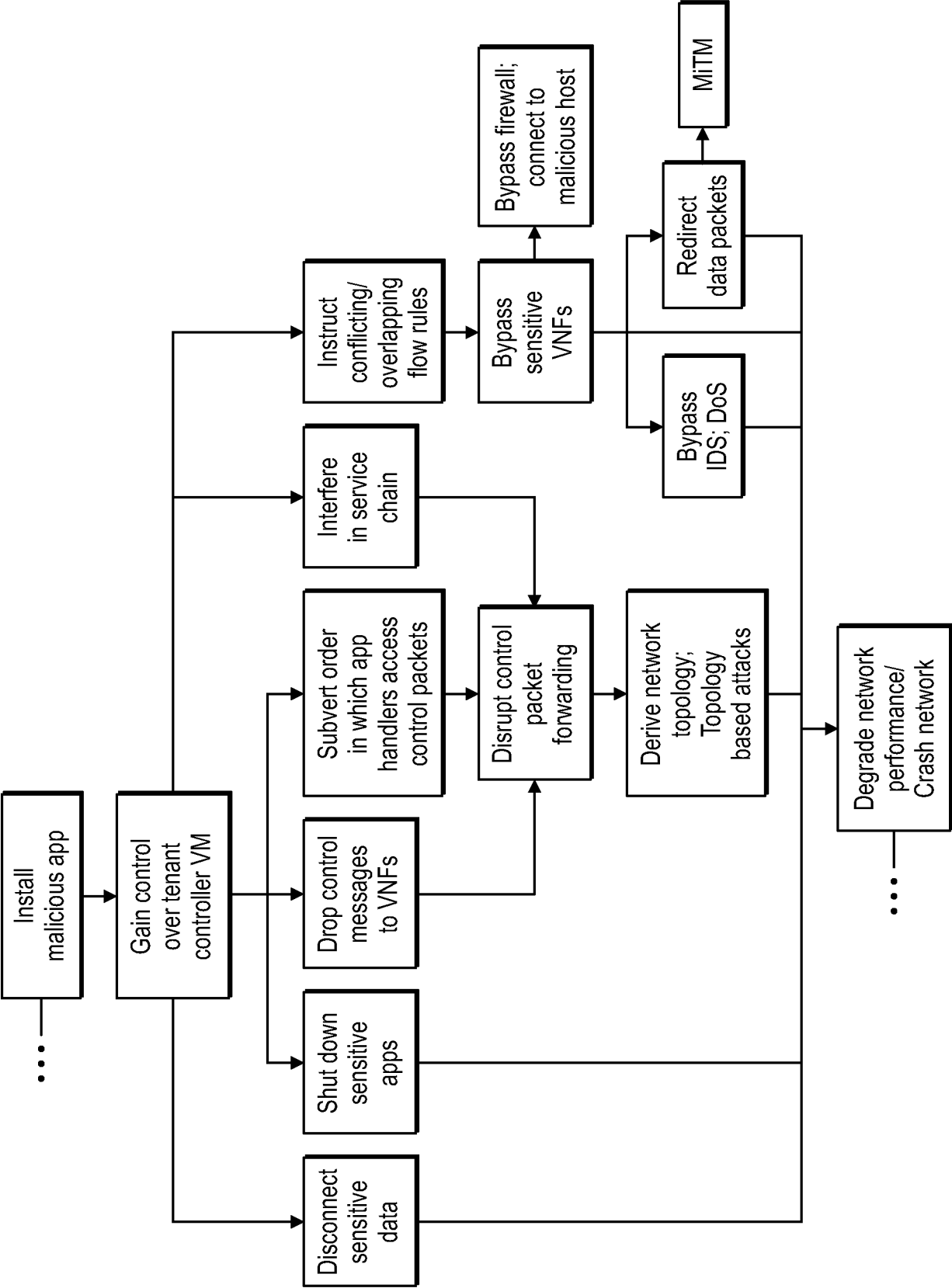
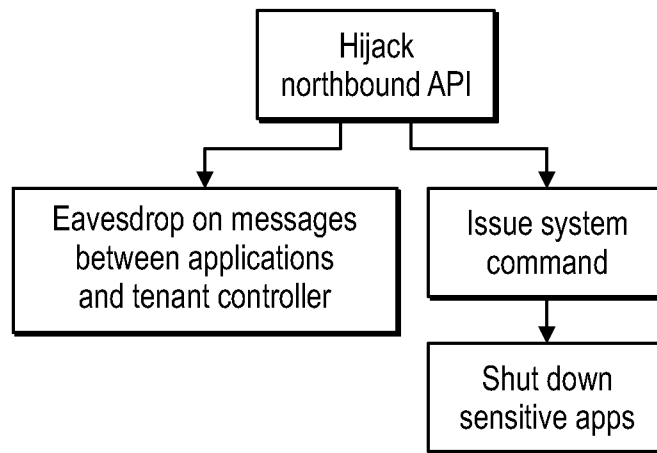
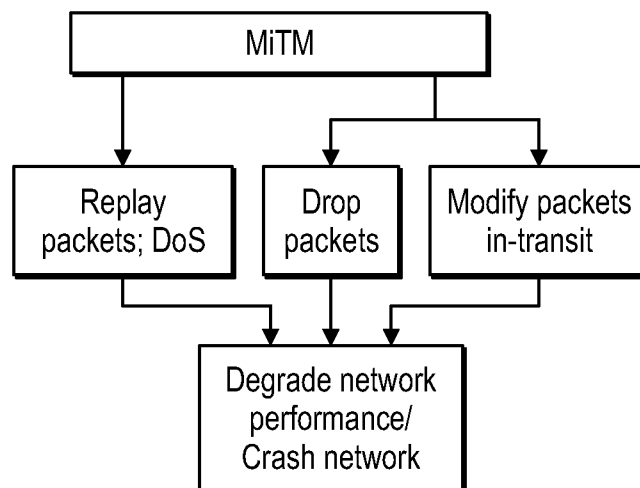


FIG. 7(b)

10/26**FIG. 7(c)****FIG. 8(a)**

11/26

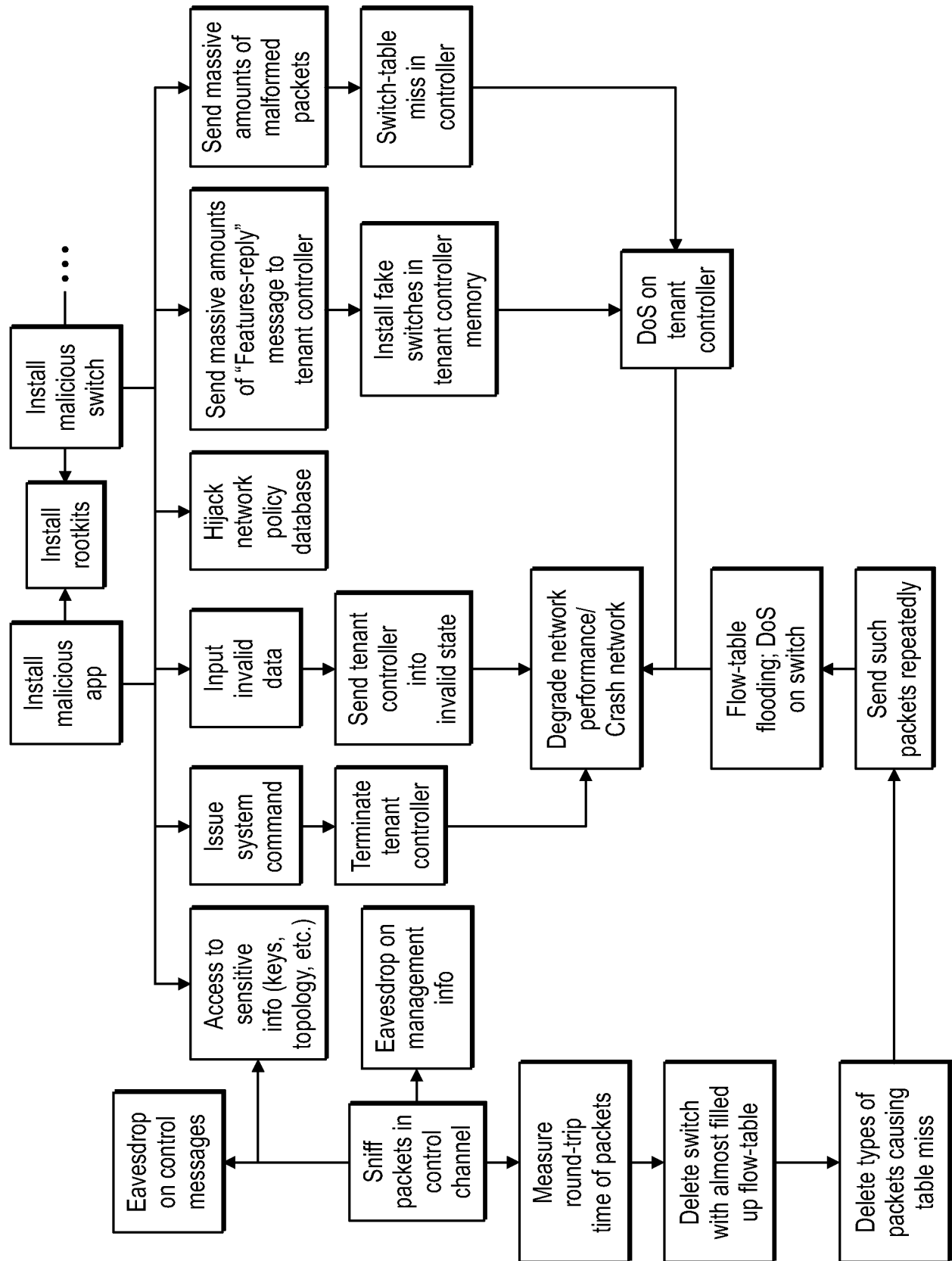


FIG. 8(b)

12/26

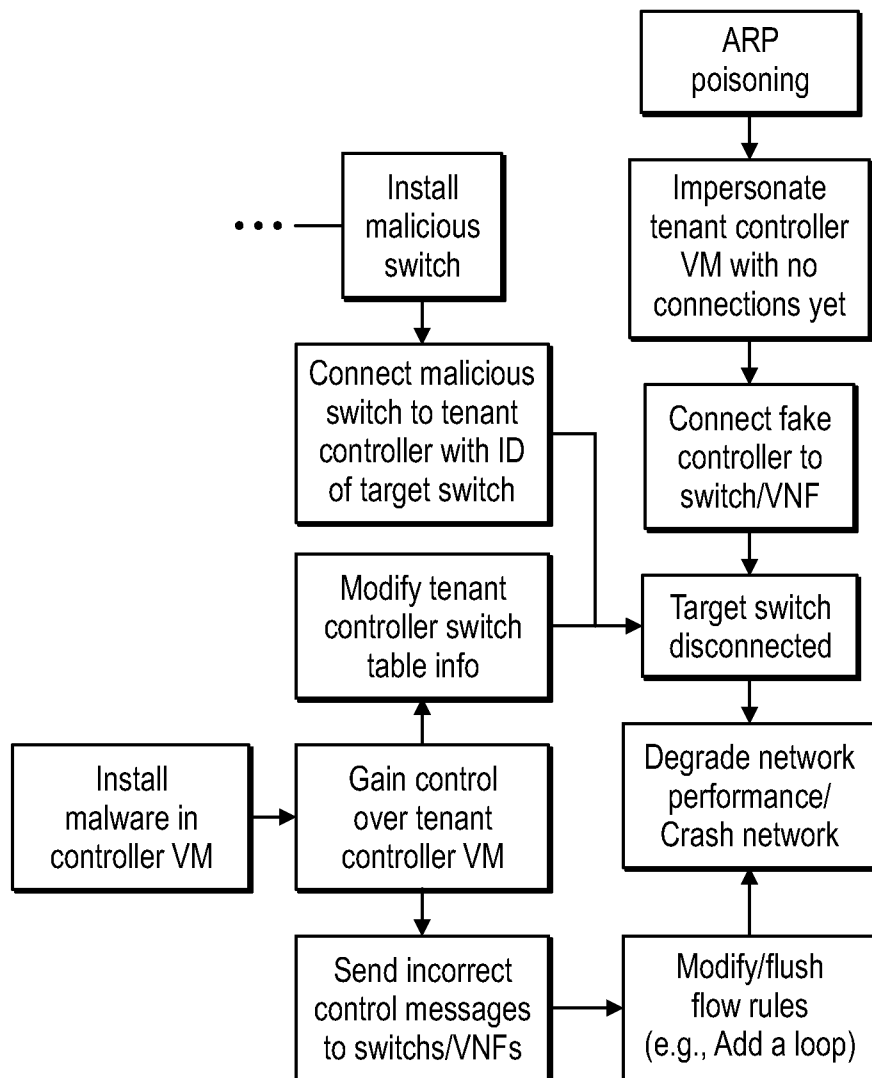


FIG. 8(c)

Topology-based attacks	Regular expression
Adding unauthorized connection in VNF	ch_i (modify VNF instantiation). ch_j (add unauthorized connection in VNF). ch_k (exploit weak crypto implementations). $\{ch_l(\text{eavesdrop on packets}) + ch_o(\text{add a loop in network}). ch_p(\text{orchestrator creates new instances of VMs (to handle excess load}). ch_q(\text{DoS on NFV infrastructure})\}$
Modifying firewall/IDS instantiation	ch_i (modify VNF instantiation). ch_j (modify rules in firewall virtual storage). ch_k (connect to malicious website) + ch_l (modify IDS rules). $\{ch_m(\text{flood network with incoming malicious traffic}). ch_p(\text{orchestrator creates new instances of VMs (to handle excess load}). ch_q(\text{DoS on NFV infrastructure}) + ch_m(\text{flood network with incoming DNS queries}). ch_p(\text{orchestrator creates new virtual DNS}). ch_q(\text{amplified DNS query request - DoS attack on victim})\}$
Passive MiTM	$\{ch_i(\text{modify VNF instantiation}). ch_m(\text{add a link to malicious VM}) + ch_j(\text{physical access of interfaces}). ch_k(\text{eavesdrop on the messages being sent}). ch_l(\text{infer topology of network})\}$
Active MiTM	ch_o (physical access to interface). ch_i (exploit weak crypto implementations). ch_j (modify packets in-transit). ch_k (crash the system) + ch_l (replay packets). ch_m (orchestrator creates new instances of VMs (to handle excess load)). ch_n (DoS on NFV infrastructure)
Exploiting Lights out Management (LOM)	
DoS attack on management network	ch_i (identify network port(s) having access to LOM). ch_j (flood the port with requests). ch_k (DoX attack on LOM port(s))
Exploiting LOM network	$\{ch_i(\text{SQL injection attack on virtual storage}) + ch_j(\text{SCA by physical access}) + ch_k(\text{cache poisoning attack}) + ch_l(\text{download unwhitelisted software}). ch_m(\text{gain control over hypervisor}) + ch_n(\text{dynamic memory overflow of hypervisor}). ch_o(\text{overwrite frame pointer of hypervisor}). ch_p(\text{code injection in hypervisor}). ch_q(\text{read secret LOM credentials}). ch_r(\text{crash the system}) + ch_s(\text{modify critical files on virtual storage}) + ch_t(\text{modify critical code on virtual compute})\}$
Exploiting hypervisor dependency on VNF (1)	ch_i (network fails). ch_j (hypervisor starts to boot). ch_k (hypervisor requests network configuration from VM running on top of it). $\{ch_l(\text{VM crashed; depends on hypervisor2 which has also crashed}) + ch_m(\text{crashed VM depends on hypervisor}). ch_n(\text{VM fails to boot}). ch_o(\text{DoS on hypervisor})\}$

FIG. 9(a)

Exploiting hypervisor dependency on VNF (2)	ch_i (network failure). ch_j (virtual forwarding function1 (VFF1) starts to boot). ch_k (requests access to VFF2). ch_l (VFF2 is crashed). ch_m (VFF2 requests access to VFF1). ch_n (deadlock arises, DoS on VFF1). ch_o (DoS on VFF2)
Exploiting insecure boot	ch_i (absence of secured boot authentication). ch_j (steal secret keys) + ch_k (rootkit injection + ch_p (reset configuration). ch_l (hypervisor compromise) + ch_m (VM compromise) + ch_n (orchestrator compromise) + ch_o (VM manager compromise)
Insecure Crash	
Compromising sensitive data	ch_i (VM/VNF crashes). $\{ch_j$ (local memory not cleared by hypervisor) + ch_o (remote memory not cleared by hypervisor)}. ch_k (new VM gets assigned same memory addresses as crashed VM). ch_l (new VM gets access to sensitive data) + ch_m (new VM gets access to keys). ch_n (new VM implements privilege escalation)
Exploiting absence of safety measures	ch_i (application within VM crashes but VM is still functional). ch_j (hypervisor resets/changes existing authorizations). ch_k (VM is restricted from performing required functions). ch_l (network crashes)
Privilege escalation	ch_i (VM _i crashes). ch_j (memory and authorizations are not cleared by hypervisor). ch_k (VM _j gets assigned the same memory location as crashed instance of VM _i). ch_l (VM _j gets same privileges as VM _i)
Authentication, Authorization, Accounting (AAA) attacks (s)	ch_i (weak authentication on NFVI manager). ch_j (access to hypervisor). ch_k (access to physical storage, compute and network). $\{ch_l$ (get secret keys) + ch_m (MiTM attacks) + ch_n (replay attacks) + ch_o (eavesdrop on communication) + ch_p (modify packets in-transit) + ch_q (assign low memory to VMs). ch_r (DoS on VMs) + ch_s (give unauthorized privileges to malicious actors) + ch_t (download unwhitelisted malware) + ch_u (add unauthorized connections)}
AAA attacks (2)	ch_i (weak authorization of hypervisor) + ch_j (weak authorization of orchestrator) + ch_k (weak authorization of VM) + ch_l (weak authorization of VNF managers)
Exploiting backdoors meant for testing	ch_i (virtualized switch in promiscuous mode). $\{ch_j$ (eavesdrop on VNF traffic with test process) + ch_k (adversary sends malicious traffic through test backdoor) + ch_l (shared memory access to test process)}. $\{ch_m$ (eavesdrop on sensitive data/secret keys) + ch_n (modify sensitive data of VNF in test/monitoring mode)}

FIG. 9(b)

Flooding attacks	$ch_i(\text{physical access to shared network recourse}). \{ch_i(\text{flood shared network with requests}) + ch_k(\text{flood shared network with high-priority messages})\}. ch_i(\text{DoS attack on target VM})$
Eavesdropping on shared resources	$\{ch_g(\text{virtual sharing of same network slice} + ch_i(\text{physical sharing of same network component})). ch_i(\text{eavesdrop on shared renounces}). ch_i(\text{absence of crypto on control plane}). ch_i(\text{reverse engineer the packets sent by target VM}). ch_k(\text{spoof target VM}). \{ch_m(\text{send modified packets with target VM id}) + ch_n(\text{request access to other VMs with target VM's id}). ch_o(\text{target VM is disconnected from these VMs in the virtual network}). ch_p(\text{replay packets received by target VM})\}. \{ch_q(\text{crash the system}) + ch_r(\text{launch DoS attack on another VM through target VM})\}$
SCA/Cache poisoning	$ch_i(\text{SCA analysis}) + ch_i(\text{cache poisoning}). ch_k(\text{extract crypto keys})$
Resources of virtual infrastructure	
Local storage attacks	$ch_i(\text{install malware at hypervisor level}). ch_i(\text{force hypervisor to fill up local storage with log}). ch_k(\text{local storage insufficient for VMs}). \{ch_l(\text{degrade network performance}) + ch_m(\text{DoS attack on VMs})\}$
Remote attacks	$ch_i(\text{install malware at hypervisor level}). \{ch_i(\text{force hypervisor to fill up remote storage with logs}). ch_k(\text{remote storage insufficient for VMs}) + ch_n(\text{remote control channel degradation}). \{ch_l(\text{degrade network performance}) + ch_m(\text{DoS attack on VMs})\}$
Memory pressure attacks	$ch_i(\text{install malware at hypervisor level}). ch_i(\text{consume kernel memory}). \{ch_l(\text{degrade network performance}) + ch_m(\text{crash the system})\}$
CPU attacks	$ch_i(\text{install malware at hypervisor level}). ch_i(\text{cause scheduler unfairness}). \{ch_l(\text{degrade network performance}) + ch_m(\text{crash the system})\}$
OS resource exhaustion	$ch_i(\text{install malware at hypervisor level}). ch_i(\text{consume file handles}). + ch_n(\text{consume event channels}). + ch_o(\text{insufficient resources for OS}). \{ch_l(\text{degrade network performance}) + ch_m(\text{crash the system})\}$

FIG. 9(c)

16/26

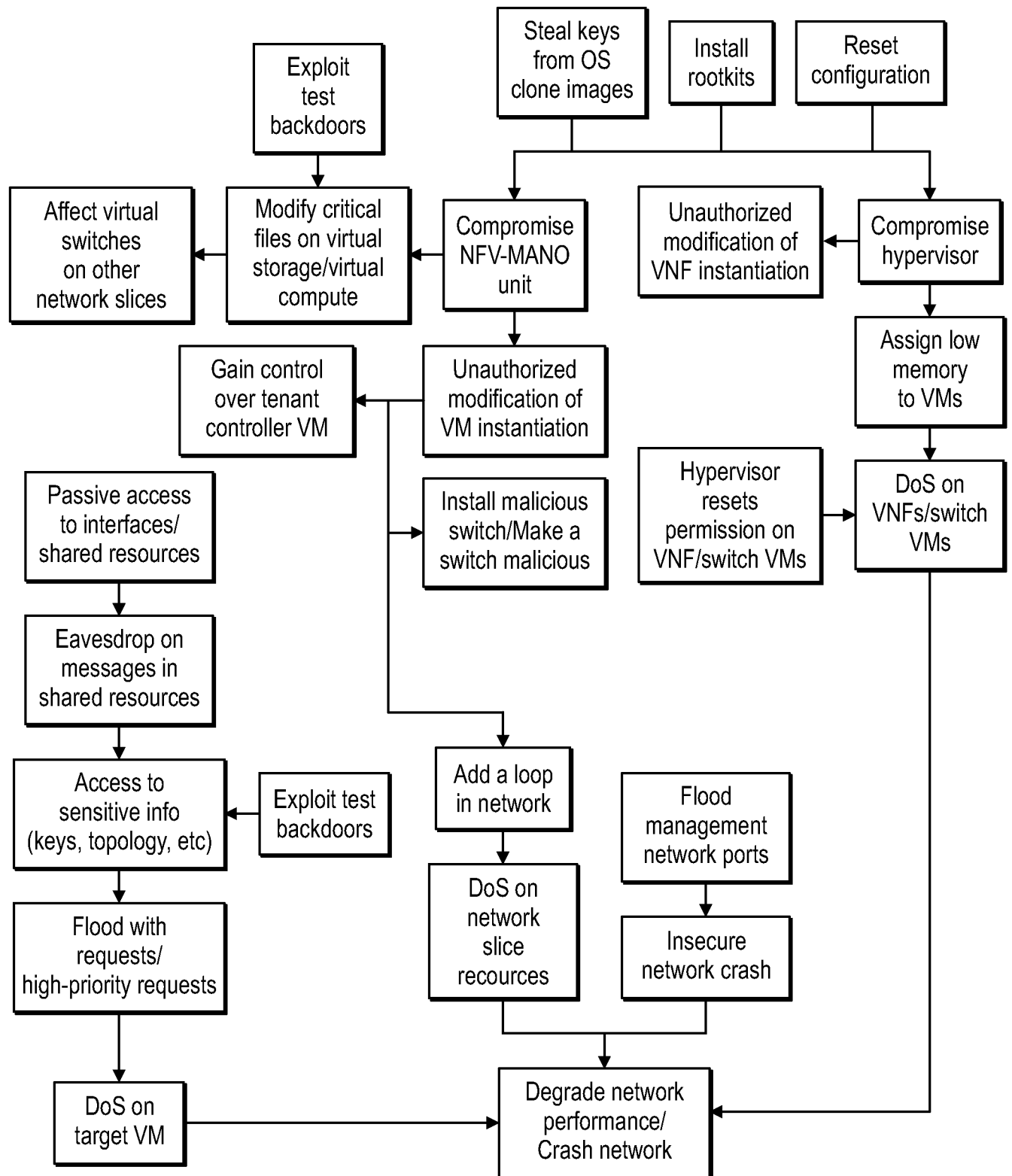
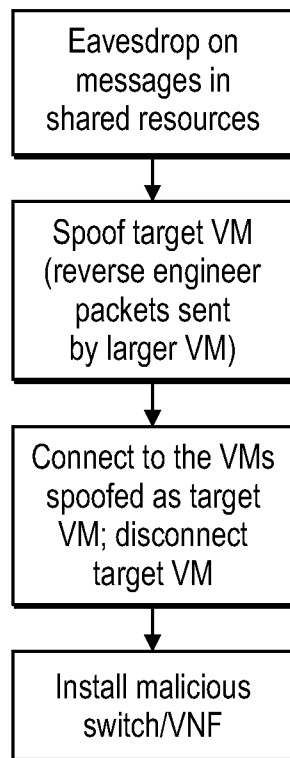
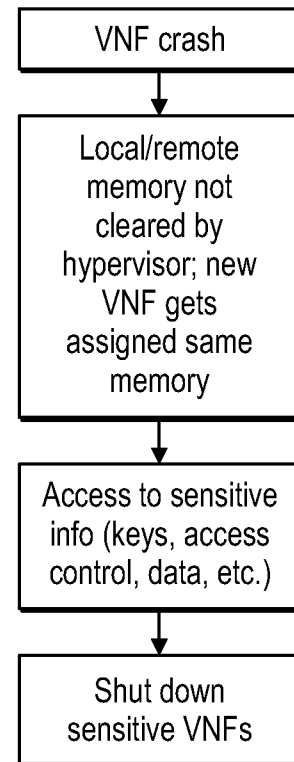
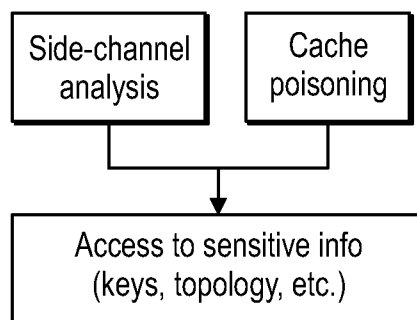


FIG. 10(a)

17/26

**FIG. 10(b)****FIG. 10(c)****FIG. 11(a)**

18/26

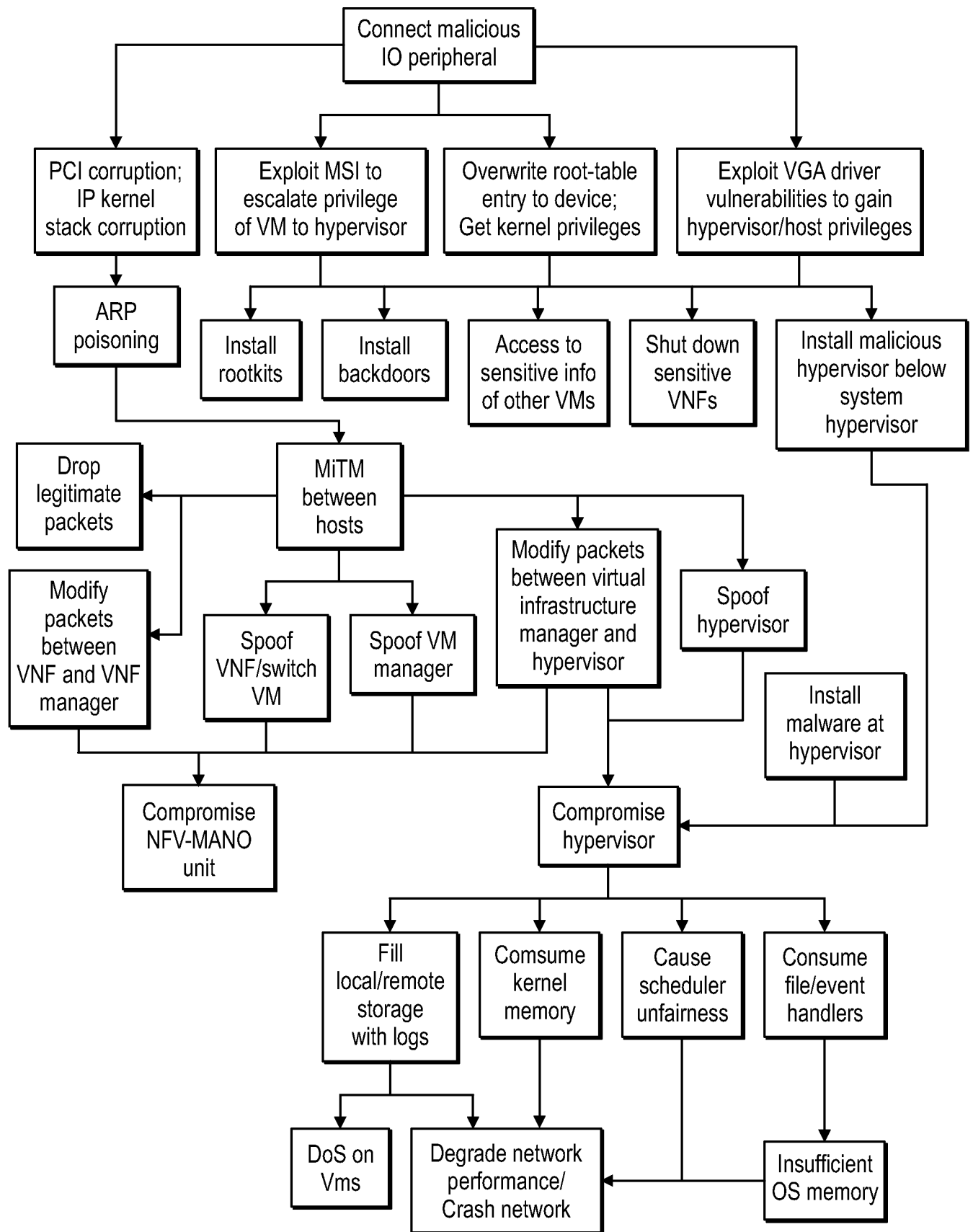


FIG. 11(b)

19/26

Graph	Number of Attack Vectors
SDN-CP	14
SDN-DP	25
NFV	25
Malicious Peripheral	49
Total	113

FIG. 12

Category	Novel Possible Exploit
SDN-CP	Drop control messages to VNFs → Disconnect targeted links in the network
SDN-DP	Gain control of tenant controller VM → Hijack network policy database
NFV	Install a malicious switch → Modify critical files on virtual storage or virtual compute
SDN-CP, SDN-DP	Hijack northbound API → Input invalid data to tenant controller, forcing it to go to an invalid state
NFV, SDN-CP	Exploit backdoors for testing → Poison tenant controller host profile reservoir
SDN-DP, NFV	Flood the switch table of target virtual switch → Exploit the insecure crash recovery of NFV to shut down new VNFs assigned the same memory as the crashed VNF
Malicious peripheral, SDN-DP	Compromise NFV-MANO unit → Issue system command to terminate controller
Malicious peripheral, NFV	Connect malicious peripheral and exploit MSI vulnerabilities → Gain hypervisor privilege

FIG. 13

20/26

Category	Number of Novel Possible Exploits
SDN-CP	36
SDN-DP	23
NFV	36
Malicious Peripheral	0
Inter-graph	24
Total	119

FIG. 14

Feature	Install Malicious Apps	Assign Low Memory to VM
Application Layer	1	0
Controller	0	0
Application-controller Interface	0	0
VNF	0	1
Network Infrastructure	0	1
Management Layer	0	1
Hypervisor	0	1
Flooding	0	1
Access Control	0	0
Data Plane	0	0
SCA	0	0
Control Channel	0	0
Sensitive Information	0	0
SDN-CP	1	0
SDN-DP	1	0
NFV	0	1
Malicious Peripheral	0	0
Head	1	0
Tail	0	0

FIG. 15

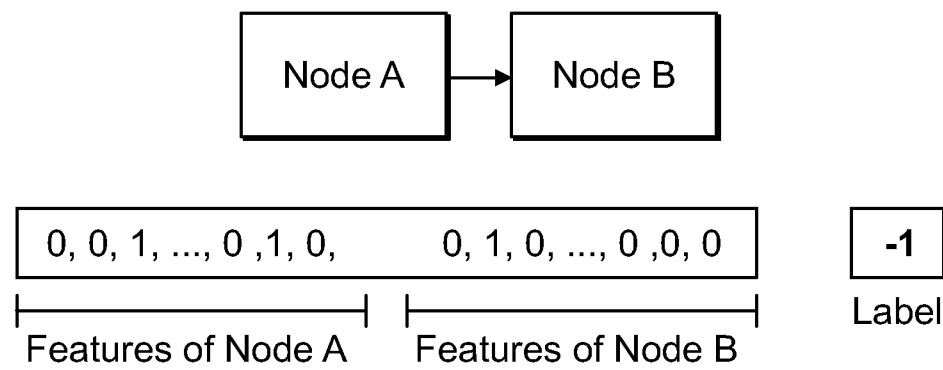


FIG. 16

Graph	Training Set	Training Set
SDN-CP	552	98
SDN-DP	898	159
NFV	510	90
Malicious Peripherals	690	122
Inter-graph Connections	6548	1156
Total	9198	1625

FIG. 17

22/26

Algorithm	SDN-CP	SDN-DP	NFV	Malicious Peripherals	Intergraph
Naive Bayes	94.44	94.91	89.8	97.44	100
Decision Tree	90.7	94.63	89.02	98.26	99.74
k-NN (k=3)	87.76	93.67	87.64	97.54	100
SVM	93.33	95.65	91.38	98.86	99.74
Neural Network	95.0	96.9	91.67	98.98	99.82
CSP	94.38	96.69	95.12	98.35	99.65

FIG. 18

Graph	Algorithm
SDN-CP	MLP (6,2)
SDN-DP	MLP (8,2)
NFV	CSP
Malicious Peripherals	MLP (5,2)
Inter-graph	MLP (5,2)

FIG. 19

23/26

	Actual = True	Actual = False	Sums
Predicted = True	26	67	93
Predicted = False	15	1517	1532
Sums	41	1584	1625

FIG. 20

Metric	Value
NPV	0.99
Precision	0.28
Recall	0.63
F1 Score	0.36
MCC	0.4
Accuracy	0.95

FIG. 21

24/26

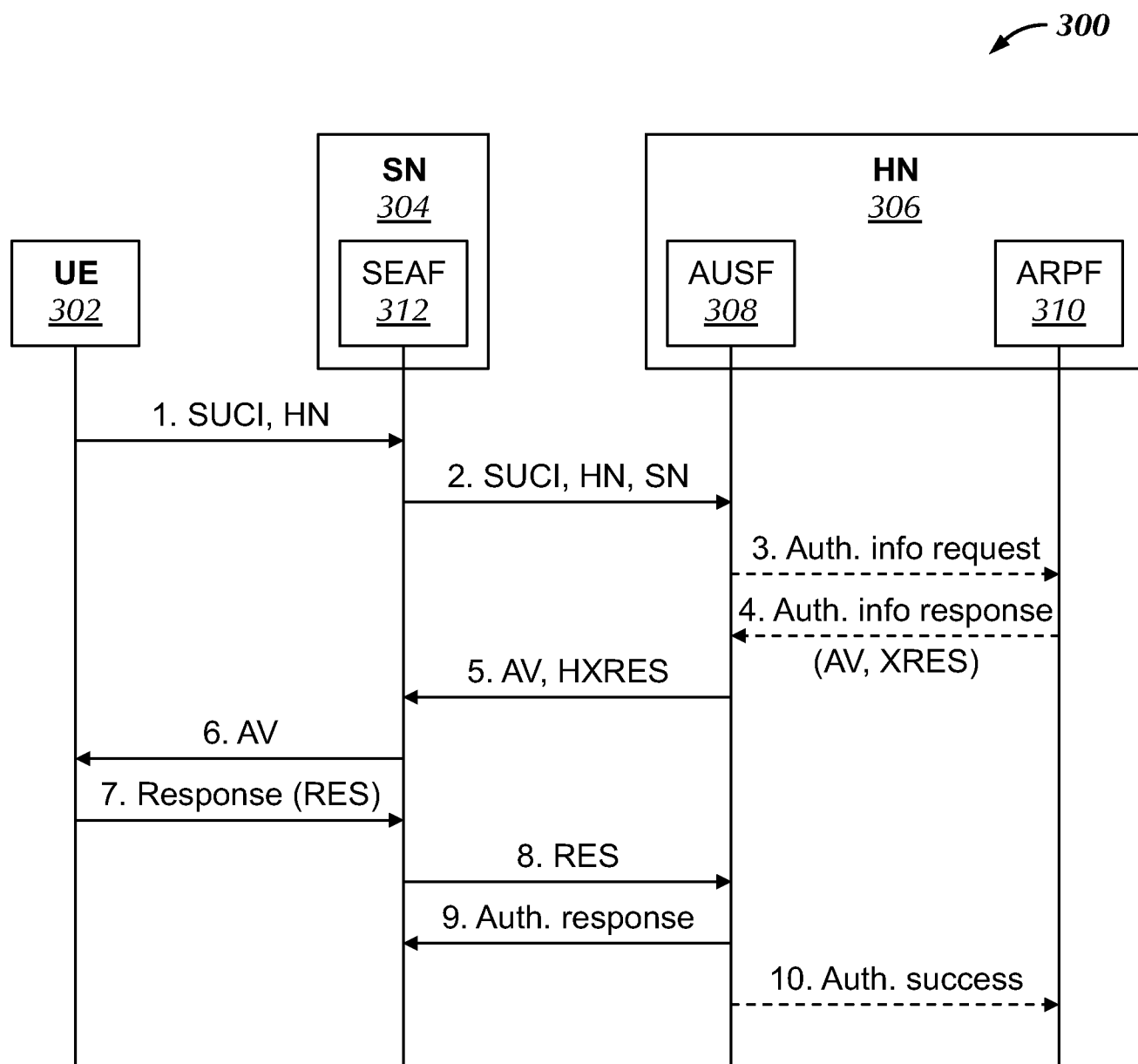


FIG. 22

25/26

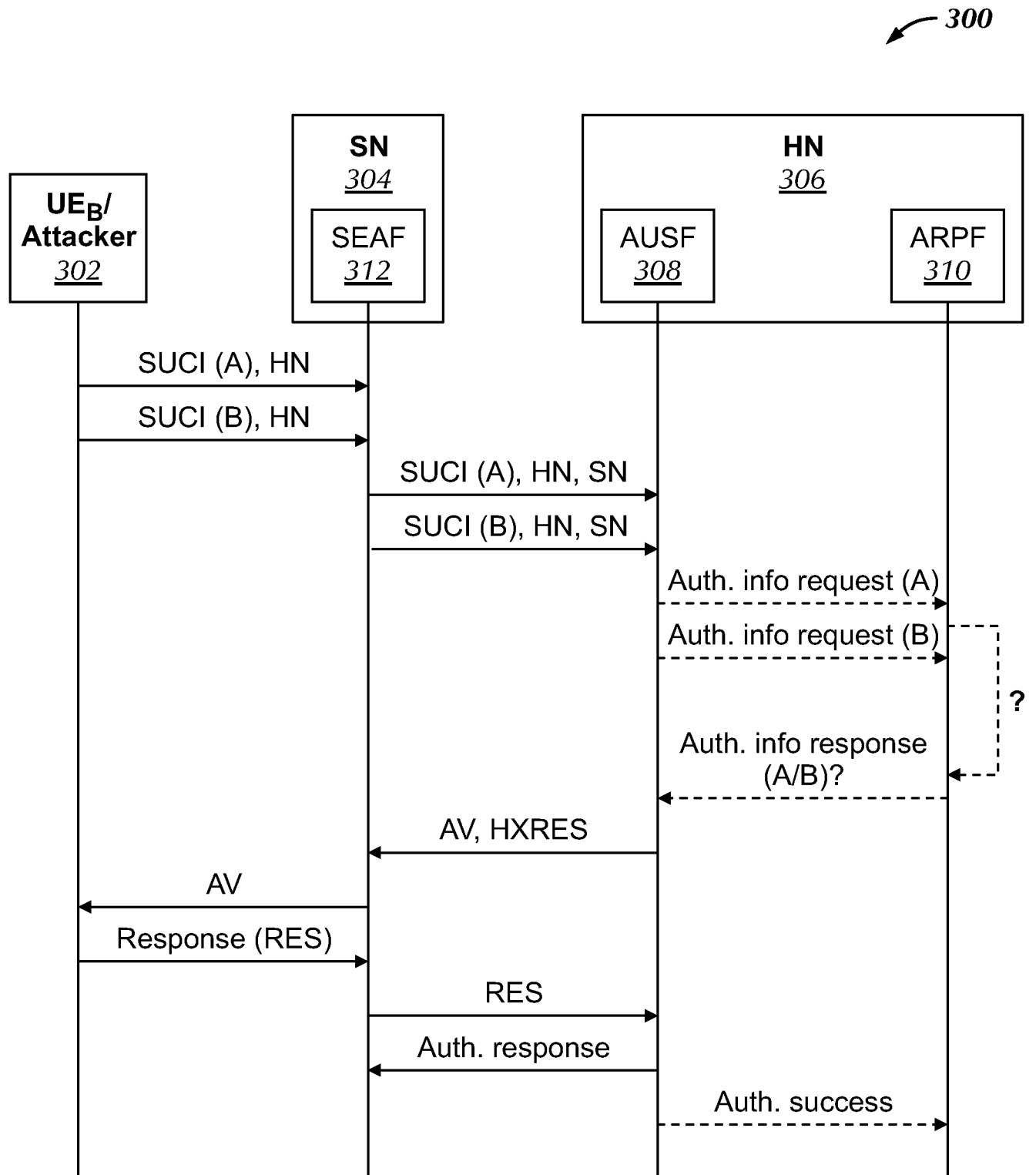


FIG. 23

26/26

Compromised Element	S1	S2	A1	A2	A3	A4	A5	A6	A7
AUSF-ARPF channel; Passive MiTM	✓	✗	✗	✗	✓	✓	✗	✗	✗
AUSF-ARPF channel; Active MiTM	✓	✗	✗	✗	✓	✗	✗	✗	✗
SEAF	✓	✗	✗	✗	✗	✓	✗	✗	✗
AUSF	✓	✗	✗	✗	✗	✓	✗	✗	✗
ARPF	✓	✗	✗	✗	✓	✗	✗	✗	✗

FIG. 24

INTERNATIONAL SEARCH REPORT

International application No.

PCT/US2021/058421

A. CLASSIFICATION OF SUBJECT MATTER

IPC(8) - G06F 8/70; G06F 9/54; G06F 17/00; H04L 29/06; H04L 29/08 (2022.01)

CPC - G05B 15/02; G06F 8/70; G06F 9/542; H04L 12/2823; H04L 12/283 (2022.01)

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

see Search History document

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

see Search History document

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

see Search History document

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2019/0364420 A1 (TELEFONAKTIEBOLAGET LM ERICSSON (PUBL)) 28 November 2019 (28.11.2019) entire document	1-42
A	— BINYAMINI et al. "An Automated, End-to-End Framework for Modeling Attacks From Vulnerability Descriptions." arXiv preprint arXiv:2008.04377 (2020). 10 August 2020 (10.08.2020) Retrieved on 07 January 2022 (07.01.2022) from <https://arxiv.org/abs/2008.04377> entire document	1-42
A	— CHENG et al. "An integrated network scanning tool for attack graph construction." International Conference on Grid and Pervasive Computing. Springer, Berlin, Heidelberg, 2011. May 2011 (05.2011) Retrieved on 07 January 2022 (07.01.2022) from <https://link.springer.com/chapter/10.1007/978-3-642-20754-9_15> entire document	1-42
A	US 2018/0300124 A1 (FOGHORN SYSTEMS, INC.) 18 October 2018 (18.10.2018) entire document	1-42
A	US 2008/0172716 A1 (TALPADE et al) 17 July 2008 (17.07.2008) entire document	1-42
A	— WAN et al. "Security Analysis Of 5G Mobile Networks." 2019 Fall Technical Forum. 30 September 2019 (30.09.2019) Retrieved on 07 January 2022 (07.01.2022) from <https://www.nctatechnicalpapers.com/Paper/2019/2019-security-analysis-of-5g-mobile-networks> entire document	1-42

☐ Further documents are listed in the continuation of Box C.☐ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"D" document cited by the applicant in the international application

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

14 January 2022

Date of mailing of the international search report

FEB 28 2022

Name and mailing address of the ISA/US

Mail Stop PCT, Attn: ISA/US, Commissioner for Patents
P.O. Box 1450, Alexandria, VA 22313-1450
Facsimile No. 571-273-8300

Authorized officer

Harry Kim

Telephone No. PCT Helpdesk: 571-272-4300