A Testbed for Trusted Telecommunications Systems in a Safety Critical Environment

Ian Oliver[1], Aapo Kalliola[1], Silke Holtmanns[1], Yoan Miche[1], Gabriela Limonta[2], Borger Vigmostad[2], and Kiti Muller[3]

[1] Cybersecurity Research Group
Nokia Bell-Labs
Finland
ian.oliver@nokia-bell-labs.com
[2] Mobile Networks: Radio Cloud
Nokia Networks
Finland
[3] Medical and Neuroscience Group
Nokia Bell-Labs
Finland

Abstract. Telecommunications systems are critical aspects of infrastructure with more safety-critical systems utilising their capabilities. Domains such as medicine and automotive applications are required to be resilient and failure tolerant. We have constructed a testbed environment that can be configured into various telecommunication operator configurations based around Network Function Virtualisation, Edge Cloud and Internet-of-Things along with trusted computing.Utilising a medical application as the motivating case to demonstrate reliability, resiliency and as a compelling demonstration we can investigate the interaction of these security technologies in telecommunications environment while providing a safety-critical use case.

1 Introduction

Telecommunications systems are now firmly established as pieces of critical infrastructure and are indeed designed and constructed to be as resilient to failure as possible. These systems however are extending from their infrastructure role into more diverse and distributed systems encompassing the now increasing reliance on IoT devices beyond that of traditional user equipment such as mobile phones. The forthcoming 5th generation (5G) telecommunication systems explicitly addresses aspects of latency, communication frequency, authorisation and other aspects of Internet-of-Things (IoT), connected cars and of virtualised infrastructures [1].

Given this increased functionality and ubiquitous nature, telecommunications systems are being used as a platform for providing services to safety-critical systems: GSM-R (now LTE-R), Tetra have been used in rail and public safety systems (police, border protection, military etc). Some governments even con-

sider the usage of commercial networks for their public safety communications[4]. The ability to provide private services through virtual operators and virtualised infrastructure then lends capabilities to other domains, such as medical and automotive domains. Indeed in the latter case the concept of the connected car with a plethora of sensors, video streaming, data collection, on-board processing and ad hoc networking is a canonical example of this.

This trend of running safety-critical systems, which almost by default now means 'cyber-physical', will continue and is extending into domains such as medicine. In that respect, we have a need to understand how said systems behave under anomalous situations and how the suite of security, privacy and trust technologies can best interact to provide better protections related to overall system integrity and its resilience under duress.

This paper presents our testbed environment [19] for examining the security properties of such systems. We utilise a motivational medical use case for demonstration built upon the telecommunications' "Network Function Virtualisation (NFV)- Edge Cloud - IoT architecture". We are currently exploring aspects of Interconnection Network security - the 'inter-telco operator internet' (SS7/Diameter) protocols affecting 5G/LTE and 3G systems, novel distributed denial of service (DDoS) analysis and protection/mitigation mechanisms especially related to virtualised workload, and the use the trusted computing environments for integrity measurements, identity and attestation.

The rest of this paper will introduce the testbed components and the current set of experiments/proof of concepts.

2 Testbed Components

We have (and are continually) constructing a testbed environment for evaluating and prototyping various security technologies and their interactions. The testbed is primarily focused on the secure provisioning of telecommunications infrastructure in a safety-critical environment - in this case we demonstrate with a medical case based on a remote medic interacting with a local medic. A visualisation of this is presented in figure 1, which shows one possible configuration to be used in testing. The testbed - while demonstrating a medical case - can be retargeted towards automotive and rail cases as required - all application domains which require not only security but some degree of real-time performance.

The application for demonstration is relatively straightforward and is based around the idea of remote surgery or some suitable medical procedure [24] to be performed at a remote site. An example of this has been the proposed usage of a robotic device [23] to enable a neurologist working to provide accurate or confirmed diagnosis at a remote location in cases of stroke, for example, a single neurologist can provide service to a number of emergency departments with aid of local medical personnel.

[4] https://www.kauppalehti.fi/lehdistotiedotteet/nokia–finnish-state-security-networks-group-and-telia-finland-trial-prioritization-of-public-safety-traffic-over-lte-networks/8JWFyZEX

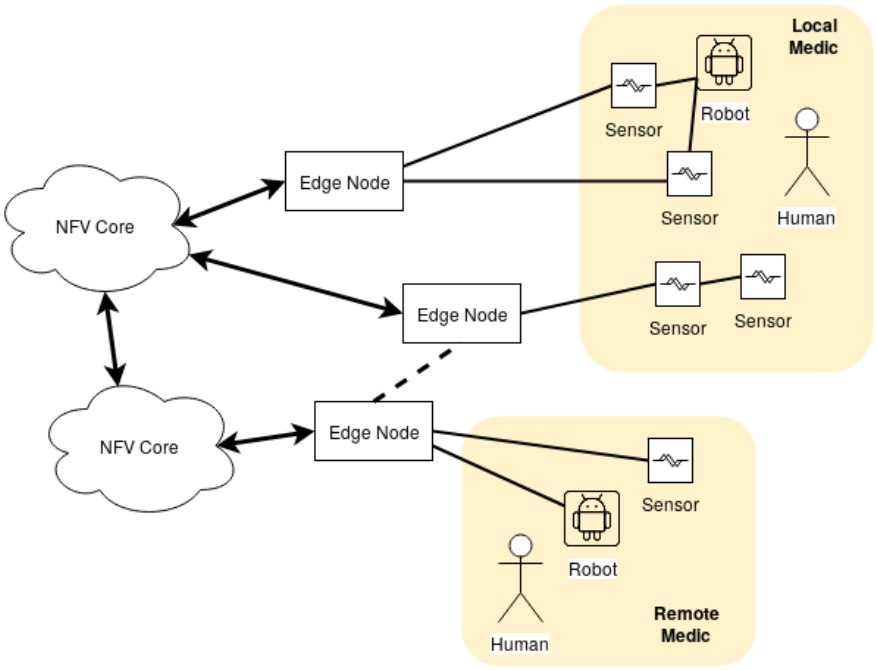

Fig. 1. Example Instantiation of Testbed

The hardware being used consists of Nokia AirFrame NFV servers with customised Linux based operating system, various base station 5G/LTE and Wifi controller components with 'Edge' functionality. IoT components come in various kinds from small scale sensors - cameras, physiological measurements etc - to larger scale independently functioning components such as the 'robot surgeon' played by a Universal Robots UR3 arm[5] and other devices accordingly. Additionally, we take into consideration the role of a human in both the remote and local roles in all scenarios.

The properties being investigated by the testbed come from the provisioning of such applications in a telco/5G environment where service availability and reliability is to be preserved in terms of system integrity and defence when under attack, eg: denial of service, specific telco core attacks, system tampering etc.

- Ensuring that the application data and instructions are not tampered with; including the workload elements (hardware, software)
- Ensuring that the application continues to function under network attack
- Ensuring that the application fails or degrades in a known or safe manner if the above characteristics are not met

[5] https://www.universal-robots.com/products/ur3-robot/

2.1 High-Level Components

The high-level architecture components are classified into three parts with the distinction being made on overall computing resources and 'interaction' with the end-user. For example, a server providing telecommunications functionality would be considered 'core', but a sensor recording temperature as 'IoT'. This classification is not meant to be read as being absolute in nature.

- **Core**: meaning the telecommunications network and centralised workload, servers - collectively known as Network Function Virtualisation (NFV)
- **Edge**: elements 'outside' of the core cloud but providing, relatively independently, application and core services. This might range from base station/Wifi controllers to localised application platforms
- **IoT**: elements such as sensors, direct human or machine interaction etc [21, 29].

2.2 Communications Infrastructure

While the core elements can rely upon high-speed, fixed communications lines, the edge and IoT elements communicate over a diverse range of systems: wifi, fixed line and of course 5G and LTE. For the medical application the configuration is such that a single core, NFV cloud provides the base telecommunications systems: subscriber databases, base stations, messaging equipment/services, etc. We use software defined networking (SDN) to provide the necessary, dynamically configurable control over the routing and network topology set up for certain applications.

2.3 Application Components

The application components are either run as hardware components (eg: robot + controller), IoT devices, 'bare-metal' software running as processes on some operating system, microcontroller - Intel NUC, ATMEL and ARM based devices - etc, or on some virtualised environment such as provided by some hypervisor, eg: Docker.

In this respect, this allows us effectively to implement any application that requires strong interaction with telecommunication systems. For example, in the medical case certain parts of the application, such as real-time video streaming, require close to core implementation, while data aggregation can be made in the edge nodes before being sent to core. Further, the medical application has security and privacy requirements which require virtualised networking to be set up to ensure these characteristics.

3 Security Test Technologies

In this section, we outline the three areas of active research regarding telecommunications system security, these are:

- Attestation and trusted environments
- Distributed denial of service and network anomaly protection
- Interconnection Network protocol protection

3.1 Attestation

Trusted computing concepts based around the ideas of Trusted Platform Module (TPM), Remote Attestation (RA) and Trusted Execution Environments (TEE) [30, 28] are well known. Their application in cloudified environment is limited; however, mechanisms such as the TPM quote, a strong identity of the actual TPM chip (and thus computing element), run-time kernel level integrity measurements and remote attestation provide a reliable mechanism for ensuring that systems confirm to a known configuration [26]. Extending these mechanisms beyond x86 servers to ARM, microcontroller and virtualised components is being prototyped within this testbed [12, 2, 3].

We have a generic attestation *environment* based around the ability to utilise any number of measurement mechanisms for any given identifiable and attestable element [27, 6, 7]. We extend this out to a more generic notarisation - ostensibly blockchain based - mechanism which provides longer term identities [4] and known measurements globally for hardware components and particularly virtual images and software components, while at a local level would be supplemented by measurements for that local system's hardware in conjunction with measurements for virtual machines taken at particular points during that machine's life cycle, eg: start, suspension, migration etc.

The attestation is made through a combination of policies denoting known or good states with rules over various aspects of the machine's measurement. A TPM quote contains not just a measure but also information about the TPM clock, reboot counts, firmware versions etc. We also utilise the ability to reason over integrity measurements over time giving the opportunity to make correspondence with other system aspects, such as comparing a given machine's TPM reboot count with, for example, reboots triggered by system updates and patching. Figure 2 shows the result of one audit of a given element and the cross-referencing of particular attestation rules against certain policies for machine configuration.

The core of the testbed utilises OpenStack as the virtualisation management environment. Here, we utilise the remote attestation to guide workload placement and add trust to VM life cycle operations. Virtual machines (implementing VNFs) are measured as images using simple cryptographic hashing [33, 32]. Life cycle operations, such as instantiation (running), suspension and migration [8] in particular, present further check points. The testbed provides a proof of concept interaction between the virtual machine management and attestation server for the purposes of providing trust over the operation on the virtual machine - this is described later in section 4. Mechanisms for extending this *into* the virtual machine are under investigation [22], for example, virtual trusted platform module (vTPM) [5] being one potential solution.

Two other aspects are under investigation, the first being the distribution of attestation functionality away from a single centralised attestation server. Partially inspired by blockchain style distribution and also to investigate localising trust responsibilities. The second derives from the first which is to form a system-wide graph of who trusts whom, once each element is capable of deciding which other elements it trusts (and to what degree). This latter *trust graph* also

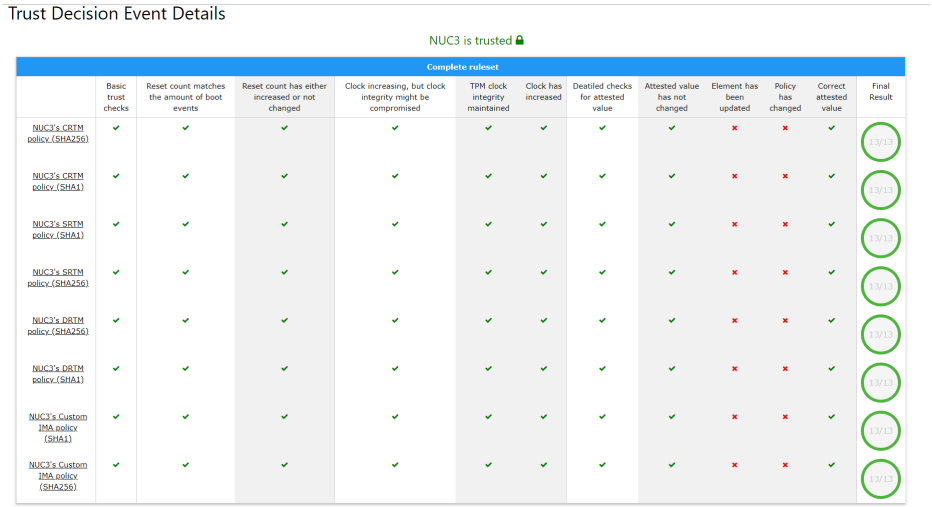

Fig. 2. Example Result of Machine Trust Audit

addresses ideas of the degree of trust between nodes and the transitivity of trust [18].

3.2 DDoS Protection

The defence mechanism is implemented on the testbed network as part of the software defined networking (SDN) as an extension of the mechanisms introduced in [20]. Typical functional blocks in the implementation include traffic sampling capabilities in the network either directly from switches or indirectly through an SDN controller, a feature extraction and learning VNF, and, finally, the rule creation and deployment logic in the form of an SDN application running on top of the SDN controller.

In order to effectively mitigate DDoS attacks, the defence logic must have an understanding of the structure of the network within the security control domain, and of the traffic flows and insertion points into the security control domain. This information can be gained from e.g. the SDN controller, which may directly contain capabilities for providing network graphs, or the graphs may be deduced from the overall set of known nodes and the rules deployed on the nodes for traffic routing.

The full testbed is constructed with multiple different networking technologies. In reality, SDN capabilities are unlikely to extend through all traffic forwarding points in the network. Thus, the DDoS protection mechanism must also be capable of reasoning where the edges of the "controllable" security control domain lie within the network. Any rules targeting traffic flowing in from edges would ideally be placed as near to the edge as possible in order to avoid forwarding malicious traffic within the network and incurring the related costs or

risking bandwidth exhaustion. An example of this configuration in the context of the testbed is shown in figure 3

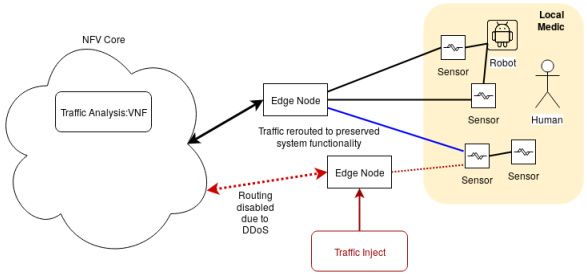

Fig. 3. Example Instantiation of Networking Architecture and Protection in the Testbed

3.3 Interconnection Network (SS7/Diameter) Protection

The network was considered closed - accessible only by mostly state owned trusted telecoms operators - it has now been opened to facilitate new services and operators. In our testbed, we, effectively, create a private telecom operator for processing medical data and connecting the medics in figure 1 together.

We isolate the control and user data related to this application by running it fully on protected and firewalled virtual nodes. In unprotected networks, utilising the commands provided by SS7 [10] and Diameter [11], a large number of potentially serious attacks can be made against individual subscribers or IoT devices, network equipment, including subscriber modification [14], location tracking [16] and password interception [15] over SMS.

In a similar manner to that shown in figure 3, we can inject such commands, directly and indirectly, into our network against VNFs existing in the core and also some edge agents and elements providing telecommunications network functionality. As these attacks can be made over a long period of time, detection is difficult. The testbed utilises machine learning over these protocols to attempt to detect the weak signals of such an attack [25].

The virtual network for critical applications has full network functionality. Its security is based on a trust system, dynamic DoS protection and a signalling firewall protection. This allows us to protect against the following attack types:

- Availability attacks (DoS) coming via interconnection, malicious updates, bots.
- Sensitive data extraction using roaming interfaces, remote code execution.

In addition, our testbed allows extensive monitoring and logging of unusual events and, by that, a better possibility to identity a potential attacker and its technique. This monitoring and logging of events is used to identify key features of abnormal and normal traffic on a connection specific or partner specific basis.

4 Scenarios, Results and Future Work

The testbed has been designed to support a number of security experiments with a 'compelling' safety related use case, in order to facilitate the necessity and the effect of providing, or not providing, specific kinds of security. The choice of a medical case is new for Nokia, but a mapping to automotive or rail can be easily made, which facilitates the presentation of such a system. In this short description of the testbed environment, we have concentrated on three particular areas, of course, this extends out to other areas, such as identity provisioning, update management, security orchestration and blockchain.

As a practical test, we have analysed the performance of the DDoS mitigation system by overloading a bottleneck link between robot control functionality and actual robot with injected malicious traffic. Without defences such an attack seriously impairs or completely prevents correct robot operation. With the DDoS mitigation system in place, the normal traffic is protected from overload-induced packet drops, and normal operation is maintained.

The amount of information that is generated through detecting security anomalies is significant, and reacting to these in a meaningful manner is complex. One aspect of this work has been to introduce the concept of security orchestrator [17] as a system-wide component, though implemented as part of the core management and operations functionality.

The use of a *security orchestrator* allows for more coördinateed responses across the system as a whole. For example, if a machine fails an integrity measurement the local response might be to migrate workload away, whereas a wider - system - response would be to additionally isolate that machine at various protocol levels and via SDN, prepare other machines for additional workload by migrating their workload in a coordinated manner, reducing the overall trust relationships in the system and causing additional protection mechanisms to be brought dynamically on-line. We actually aim to move to a more 'graceful' response to system failure than is currently possible.

Two significant results have come from extended analysis of attestation quotes: the first being that trust is fragile due to updates, file changes, configuration changes and, often, due to the misimplementation of TCG standards; the second being that reaction to a trust failure can not be to remove the element (machine, VM or device) from the network but to react in an appropriate manner.

The role of trust beyond boot-time measurement and, in the case of OpenStack, when a virtual machine is instantiated - the attestation server provides information about which physical hardware a VM that requires trust can be started on - are currently the only points where trust is considered. Ignoring the case where measurements can be taken inside a VM and reported by some vTPM mechanism - this has other issues that are not well solved in the current CPU architectures - we are left with the situation of when to take measurements. A basic pattern emerges where an operation on a virtual machine is the trigger for both a pre and post-operation measurement of the hardware environment coupled with a measurement of the virtual machine itself. These measures can then be used a) to ensure that the underlying hardware has remained in a known state (eg: no reboots, or if reboot has occurred, no changes), and b) that the

virtual machine snapshot/instance has not changed [9] under operations, such as suspend and migrate. We can extend this, in the case of suspend operations, to ensure that when the machine is being restarted then it is on the same physical hardware backed up through TPM sealing, for example.

We have developed using root cause analysis (RCA) [13] a set of causal factor trees (CFTs) over the attestation results that identify the possible causes of failures. Each attestation rule can be mapped to its corresponding causal factor tree for analysis, which is then linked to a mitigation procedure. Often, it is the case that a trust failure is an indication of a failed or not yet performed update, system management error, a system crash etc.

Figure 4 shows a common situation in which an element fails its attestation checks due to inconsistencies in the number of reboots reported by the TPM and the amount stored on the attestation server. This kind of failure is common in server class hardware, since they perform a series of self-tests on start up, which are only recorded by the TPM. By using RCA and CFT, we can identify situations in which trust or trustworthiness of an element can be safely recovered. Then, the existing attestation rules can be extended to include this behaviour as normal and avoid a failure in the future.

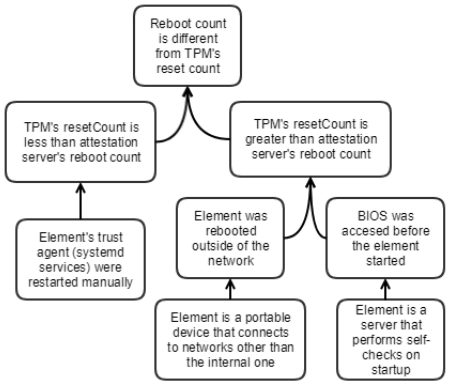

Fig. 4. Causal Factor Tree for One Kind of Common Trust Failure

An investigation is now under way regarding how each of the above interact. One scenario is that DDoS attacks now appear to act as covers for more sophisticated attacks, such as those seen in Interconnection Network. By relating the points of attack and the routing graph [31], we can deliberately reduce the amount of trust between components, or vice versa, if a component fails attestation then the network routing can be directed via additional checks, such as firewalls or sandboxing. The aim here is, primarily, to preserve availability and functionality of the system by increasing the resilience.

Returning to the medical application - this as noted has provided the use case and requirements for the kinds of resilience and mitigations that we are

constructing and evaluating within this testbed. For the user of the medical application they can be sure that the data being received is uncorrupted and that the service they are providing over the telecommunications infrastructure is reliable. For example, taking the cases above, we have been able to examine scenarios:

- provisioning of real-time data feeds and interaction under DDoS attack
- dynamic reorganisation of routing to ensure no loss of traffic and minimised network performance loss
- ensuring that the VNFs providing the application are of good integrity
- ensuring that the telco infrastructure remains reliable, and the medical applications isolated, under interconnection attacks
- ensuring the hardware, especially edge and IoT node are untampered

We continue this work with more integration of the above, addressing alternate domains - particularly rail and automotive, applying the lessons learnt in constructing the testbed and experiment into those domains and evaluating automated and proactive, proportionate responses to network/system anomalies/failures.

Acknowledgement. This work has been partially funded by EU ECSEL Project SECREDAS (Grant Number: 783119) and EU Horizon 2020 Project SCOTT (Grant Number: 737422).

References

1. Ahmad, I., Kumar, T., Liyanage, M., Okwuibe, J., Ylianttila, M., Gurtov, A.V.: 5g security: Analysis of threats and solutions. In: IEEE Conference on Standards for Communications and Networking, CSCN 2017, Helsinki, Finland, September 18-20, 2017. pp. 193–199. IEEE (2017), https://doi.org/10.1109/CSCN.2017.8088621
2. Ambrosin, M., Conti, M., Ibrahim, A., Neven, G., Sadeghi, A.R., Schunter, M.: SANA. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security - CCS'16. pp. 731–742. ACM Press, New York, New York, USA (2016), http://dl.acm.org/citation.cfm?doid=2976749.2978335
3. Asokan, N., Brasser, F., Ibrahim, A., Sadeghi, A.R., Schunter, M., Tsudik, G., Wachsmann, C.: SEDA: Scalable Embedded Device Attestation http://www.ics.uci.edu/ gts/paps/seda-CCS15.pdf
4. Augot, D., Chabanne, H., Chenevier, T., George, W., Lambert, L.: A user-centric system for verified identities on the bitcoin blockchain. In: Data Privacy Management, Cryptocurrencies and Blockchain Technology, pp. 390–407. Springer (2017)
5. Berger, S., Cceres, R., Goldman, K.A., Perez, R., Sailer, R., Doorn, L.: vtpm: Virtualizing the trusted platform module. In: In USENIX Security. pp. 305–320 (2006)
6. Berger, S., Goldman, K., Pendarakis, D., Safford, D., Valdez, E., Zohar, M.: Scalable Attestation: A Step Toward Secure and Trusted Clouds. In: 2015 IEEE International Conference on Cloud Engineering. pp. 185–194. IEEE (mar 2015), http://ieeexplore.ieee.org/document/7092916/

7. Chen, L., Landfermann, R., Löhr, H., Rohe, M., Sadeghi, A.R., Stüble, C.: A protocol for property-based attestation. In: Proceedings of the first ACM workshop on Scalable trusted computing - STC '06. p. 7. ACM Press, New York, New York, USA (2006), http://portal.acm.org/citation.cfm?doid=1179474.1179479
8. Danev, B., Masti, R.J., Karame, G.O., Capkun, S.: Enabling secure vm-vtpm migration in private clouds. In: in ACSAC 2011. pp. 187–196
9. Dewan, P., Durham, D., Khosravi, H., Long, M., Nagabhushan, G.: A hypervisor-based system for protecting software runtime memory and persistent storage. In: Proceedings of the 2008 Spring Simulation Multiconference. pp. 828–835. SpringSim '08, Society for Computer Simulation International, San Diego, CA, USA (2008), http://dl.acm.org/citation.cfm?id=1400549.1400685
10. Dryburgh, L., Hewett, J.: Signaling System No. 7 (SS7/C7): Protocol, Architecture, and Applications. Cisco Press (2003)
11. Fajardo, V., Arkko, J., Loughney, J., Zorn, G.: Diameter Base Protocol. RFC 6733 (Oct 2012), https://rfc-editor.org/rfc/rfc6733.txt
12. Ghosh, A., Sapello, A., Poylisher, A., Chiang, C.J., Kubota, A., Matsunaka, T.: On the Feasibility of Deploying Software Attestation in Cloud Environments. In: 2014 IEEE 7th International Conference on Cloud Computing. pp. 128–135. IEEE (jun 2014), http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6973733
13. Ghosh, M., Varghese, A., Gupta, A., Kherani, A.A., Muthaiah, S.N.: Detecting misbehaviors in vanet with integrated root-cause analysis. Ad Hoc Networks 8(7), 778 – 790 (2010), http://www.sciencedirect.com/science/article/pii/S157087051000034X
14. Holtmanns, S., Miche, Y., Oliver, I.: Subscriber profile extraction and modification via diameter interconnection. In: Yan, Z., Molva, R., Mazurczyk, W., Kantola, R. (eds.) Network and System Security - 11th International Conference, NSS 2017, Helsinki, Finland, August 21-23, 2017, Proceedings. Lecture Notes in Computer Science, vol. 10394, pp. 585–594. Springer (2017), https://doi.org/10.1007/978-3-319-64701-2-45
15. Holtmanns, S., Oliver, I.: SMS and one-time-password interception in LTE networks. In: IEEE International Conference on Communications, ICC 2017, Paris, France, May 21-25, 2017. pp. 1–6. IEEE (2017), https://doi.org/10.1109/ICC.2017.7997246
16. Holtmanns, S., Rao, S.P., Oliver, I.: User location tracking attacks for LTE networks using the interworking functionality. In: 2016 IFIP Networking Conference, Networking 2016 and Workshops, Vienna, Austria, May 17-19, 2016. pp. 315–322. IEEE (2016), https://doi.org/10.1109/IFIPNetworking.2016.7497239
17. Jäger, B.: Security orchestrator: Introducing a security orchestrator in the context of the ETSI NFV reference architecture. In: 2015 IEEE TrustCom/BigDataSE/ISPA, Helsinki, Finland, August 20-22, 2015, Volume 1. pp. 1255–1260. IEEE (2015), https://doi.org/10.1109/Trustcom.2015.514
18. Jøsang, A., Pope, S.: Semantic constraints for trust transitivity. In: Proceedings of the 2Nd Asia-Pacific Conference on Conceptual Modelling - Volume 43. pp. 59–68. APCCM '05, Australian Computer Society, Inc., Darlinghurst, Australia, Australia (2005), http://dl.acm.org/citation.cfm?id=1082276.1082284
19. Kalliola, A., Lal, S., Ahola, K., Oliver, I., Miche, Y., Holtmanns, S.: Testbed for security orchestration in a network function virtualization environment. In: 2017 IEEE Conference on Network Function Virtualization and Software Defined Networks, NFV-SDN 2017, Berlin, Germany, November 6-8, 2017. pp. 1–4. IEEE (2017), https://doi.org/10.1109/NFV-SDN.2017.8169857

20. Kalliola, A., Lee, K., Lee, H., Aura, T.: Flooding ddos mitigation and traffic management with software defined networking. In: 4th IEEE International Conference on Cloud Networking, CloudNet 2015, Niagara Falls, ON, Canada, October 5-7, 2015. pp. 248–254. IEEE (2015), https://doi.org/10.1109/CloudNet.2015.7335317
21. Kennell, R., Jamieson, L.H.: Establishing the genuinity of remote computer systems (2003), https://dl.acm.org/citation.cfm?id=1251374
22. Liu, Q., Weng, C., Li, M., Luo, Y.: An in-vm measuring framework for increasing virtual machine security in clouds. IEEE Security & Privacy 8(6), 56–62 (2010), https://doi.org/10.1109/MSP.2010.143
23. Lukander, K., Jagadeesan, S., Chi, H., Müller, K.: Omg!: A new robust, wearable and affordable open source mobile gaze tracker. In: Proceedings of the 15th International Conference on Human-computer Interaction with Mobile Devices and Services. pp. 408–411. MobileHCI '13, ACM, New York, NY, USA (2013), http://doi.acm.org/10.1145/2493190.2493214
24. Marja, S., Jyrki, K., Tero, J., Peter, E., Toni, J., Petri, P., Kiti, M., Jarno, S.: Live delivery of neurosurgical operating theater experience in virtual reality. Journal of the Society for Information Display 26(2), 98–104
25. Miché, Y., Oliver, I., Holtmanns, S., Kalliola, A., Akusok, A., Lendasse, A., Björk, K.: Data anonymization as a vector quantization problem: Control over privacy for health data. In: Buccafurri, F., Holzinger, A., Kieseberg, P., Tjoa, A.M., Weippl, E.R. (eds.) Availability, Reliability, and Security in Information Systems - IFIP WG 8.4, 8.9, TC 5 International Cross-Domain Conference, CD-ARES 2016, and Workshop on Privacy Aware Machine Learning for Health Data Science, PAML 2016, Salzburg, Austria, August 31 - September 2, 2016, Proceedings. Lecture Notes in Computer Science, vol. 9817, pp. 193–203. Springer (2016), https://doi.org/10.1007/978-3-319-45507-5-13
26. Oliver, I., Holtmanns, S., Miche, Y., Kalliola, A., Lal, S., , Ravidas, S.: Experiences in trusted cloud computing. In: 11th IEEE International Conference on Network and System Security, NSS 2017, Helsinki, Finland, August 2123. Springer (2017)
27. Oliver, I., Lal, S., Ravidas, S., Taleb, T.: Assuring virtual network function image integrity and host sealing in telco cloud. In: IEEE ICC 2017, Paris, France (May 2017)
28. Osborn, J.D., Challener, D.C.: Trusted Platform Module Evolution. Johns Hopkins APL Technical Digest 32(2), 536–543 (2013)
29. Seshadri, A., Luk, M., Perrig, A.: SAKE: Software Attestation for Key Establishment in Sensor Networks. In: Distributed Computing in Sensor Systems, pp. 372–385. Springer Berlin Heidelberg, Berlin, Heidelberg (2008), http://link.springer.com/10.1007/978-3-540-69170-9_25
30. TCG: Trusted Platform Module Library, Part 1: Architecture. Trusted Platform Module Library Specification, Family 2.0 Level 00, Revision 01.38, The Trusted Computing Group (September 2016)
31. Thottan, M., Martino, C.D., Kim, Y.J., Atkinson, G., Choi, N., Mohanasamy, N., Jagadeesan, L., Mendiratta, V., Simsarian, J., Kozicki, B.: The Network OS: Carrier-grade SDN control of multi-domain, multi-layer networks. Bell Labs Technical Journal 21, 1–29 (2017)
32. Yeluri, R., Castro-Leon, E.: Trusted Virtual Machines: Ensuring the Integrity of Virtual Machines in the Cloud, pp. 161–178. Apress, Berkeley, CA (2014)
33. Yu, A., Qin, Y., Wang, D.: Obtaining the integrity of your virtual machine in the cloud. In: Lambrinoudakis, C., Rizomiliotis, P., Wlodarczyk, T.W. (eds.) IEEE 3rd International Conference on Cloud Computing Technology and Science, CloudCom 2011, Athens, Greece, November 29 - December 1, 2011. pp. 213–222. IEEE Computer Society (2011), https://doi.org/10.1109/CloudCom.2011.37