Towards a Framework for Assessing the Feasibility of Side-channel Attacks in Virtualized Environments

Tsvetoslava Vateva-Gurova[1], Jesus Luna[1,2], Giancarlo Pellegrino[1] and Neeraj Suri[1]
[1]*Dept of CS, TU Darmstadt, Darmstadt, Germany*
[2]*Cloud Security Alliance, Edinburgh, U.K.*

Keywords: Feasibility Analysis, Feasibility Factors, Security Classifications, Side-channel Attacks.

Abstract: Physically co-located virtual machines should be securely isolated from one another, as well as from the underlying layers in a virtualized environment. In particular the virtualized environment is supposed to guarantee the impossibility of an adversary to attack a virtual machine e.g., by exploiting a side-channel stemming from the usage of shared physical or software resources. However, this is often not the case and the lack of sufficient logical isolation is considered a key concern in virtualized environments. In the academic world this view has been reinforced during the last years by the demonstration of sophisticated side-channel attacks (SCAs). In this paper we argue that the feasibility of executing a SCA strongly depends on the actual context of the execution environment. To reflect on these observations, we propose a feasibility assessment framework for SCAs using cache based systems as an example scenario. As a proof of concept we show that the feasibility of cache-based side-channel attacks can be assessed following the proposed approach.

1 INTRODUCTION

The term virtualization is widely used in the IT community throughout decades starting from the late sixties until now, overcoming periods of less popularity to gain significance again during the last two decades (Popek and Goldberg, 1974; Figueiredo et al., 2005). A virtualized environment (VE) is characterized by the low-level abstraction that virtualization provides by decoupling the operating system from the hardware state. A software layer called hypervisor or Virtual Machine Monitor (VMM) is the foundation that enables multiplexing multiple tenants encapsulated in virtual machines (VM) on a single physical resource (cf. Figure 1). Such a multiplexing scenario is referred to as a multitenancy model (Mell and Grance, 2009). Due to its characteristics and benefits such as decreased operational costs, reduced server sprawl, etc. (Pearce et al., 2013), virtualization is the enabling technology also for other complex models such as server consolidation and Cloud computing (Padala et al., 2007; Marty and Hill, 2007; Uddin and Rahman, 2010). In such a complex context, a tenant might be assigned to reside on the same physical resources as their adversary or an attacker, and the VE is assumed to provide secure computing environment complying with the requirements of the tenants.

Many of the requirements on virtualization from its early stages are still valid today. (Popek and Goldberg, 1974) defined a VM as ”an efficient, isolated duplicate of the real machine”. From this definition we can elicit isolation as a key property of a VE. We consider isolation as the inability of one VM to gain information regarding the co-located VMs, as well as to affect or intervene with their operation. (Ristenpart et al., 2009; Zhang et al., 2012; Hlavacs et al., 2011; Wu et al., 2012) demonstrate side-channel and covert-channel attacks that manage to break the presumed strong logical isolation provided by the virtualization in the Cloud, and with this define the insufficient isolation as a key concern in the VE. A side-channel is a communication channel that stems from the usage of shared resources and can be exploited e.g., through observations or manipulations. Since the traditional intrusion detection systems are not designed to protect from such exploits, side-channel attacks (SCAs) are considered among the main threats for compromising the isolation in a VE.

1.1 Problem Statement

Although current academic research has shown that side-channel attacks are possible in a VE under certain conditions and considering different assumptions

Vateva-Gurova T., Luna J., Pellegrino G. and Suri N..
Towards a Framework for Assessing the Feasibility of Side-channel Attacks in Virtualized Environments.
DOI: 10.5220/0005052101130124
In *Proceedings of the 11th International Conference on Security and Cryptography* (SECRYPT-2014), pages 113-124
ISBN: 978-989-758-045-1

(Xu et al., 2011; Wu et al., 2012; Hlavacs et al., 2011; Zhang et al., 2012; Ristenpart et al., 2009; Yarom and Falkner, 2013), and the industrial world tries to address this threat (Amazon Web Services, 2014), no clear answer to the question *under which conditions are SCAs feasible in a specific execution environment* (referred to as *context* for the rest of the paper) has been given. The lack of a generic SCAs classification that takes into consideration the conditions under which these attacks have been demonstrated, impedes answering this and other related questions in a systematic way. Thus, the assessment of feasibility of side-channel attacks taking into consideration the specific context is needed, to also aid research on assessment of the strength of the isolation provided by a specific virtualized environment, and the actual countermeasures to mitigate SCAs. To the best of our knowledge, there is no existing framework that estimates the feasibility of a context based SCA in a VE.

1.2 System Model

Our primary goal is to address this problem and to estimate the feasibility of cache-based SCAs that might compromise the isolation in the VE. For this purpose, we focus our system model on the VE and the hierarchy of caches as the shared medium used for the conduct of SCAs (cf. Figure 1). We briefly explain how a cache-based side-channel attack can work in this context. We assume that two distinct virtual machines - VM1 and VM2 - are running on the same physical machine. Next, we assume that a malicious process is running in VM2 and aims at compromising the confidentiality of VM1 through observations of the access patterns to the shared cache. The process in VM2 continuously writes the same data into the cache and measures the time needed to fill up the cache. If VM1 has meanwhile accessed the cache, VM2's access time will be increased, as the previously written data by VM2 has been replaced by VM1's data. Thus through repeated measurements in an idealized scenario, the malicious process in VM2 can infer VM1's access patterns to the cache and derive information through it.

1.3 Contributions

Our objective is to investigate the types of SCAs, to determine the conditions where side-channels are exploitable for a specific context. For this purpose, we derive a generic classification which can serve as a basis for analysis and comparison of SCAs with respect to various characteristics. In addition, it can be used as a basis for quantification of different aspects of

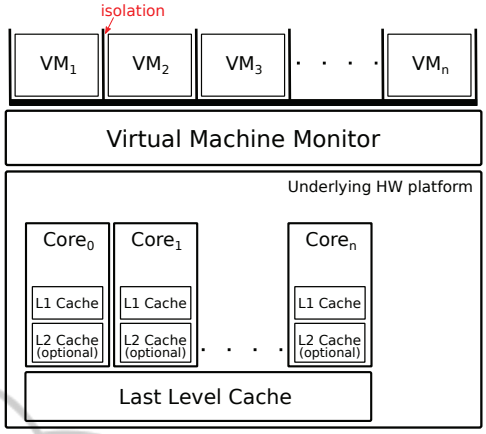

Figure 1: Isolation in a virtualized environment.

these attacks. Our supposition is that by means of an extensive classification that comprises various aspects of an attack, it is easier to identify what the possible mitigation paths in a specific context are and what protection mechanisms can be applied to decrease the probability of an exploitation of side-channels.

From this generic classification, we aim to provide guidance on estimating the feasibility of cache-based SCAs in a VE with respect to a specific context. We do not claim to provide absolute statements regarding the feasibility of a certain attack (as this depends on the adopted means), but rather aim at providing information about the conditions under which certain types of attacks are more or less probable, complementary to VE isolation assessment research. To support our approach we consider the family of demonstrated SCAs in a VE and provide a proof of concept based on it. Our overall contributions being: *(i)* to derive a generic classification of SCAs, *(ii)* to show how the provided classification can be used to estimate the feasibility of a cache-based SCA in the VE, and *(iii)* to provide a proof of concept for the conducted feasibility analysis.

The remainder of the paper is structured as follows. Section 2 investigates the state-of-the-art in the area of SCAs classification and their feasibility in the virtualized context. Section 3 describes the proposed classification and Section 4 details in the feasibility analysis, and provides a proof of concept.

2 RELATED WORK

The area of SCA based compromising of isolation in virtualized environments is an actively researched area (Kim... (Kim et al., 2012; Li et al., 2013; Stefan et al., 2013). Also, much effort has been devoted to formulate different sophisticated side-channel at-

tacks in order to demonstrate the relevance of this threat for virtualized environments and multitenant scenarios (Ristenpart et al., 2009; Zhang et al., 2012; Hlavacs et al., 2011; Wu et al., 2012; Xu et al., 2011). With this context, we overview the state of the are for classification and feasibility assessment of SCAs.

Classification of SCAs. There exists a variety of initiatives for classification of side-channel attacks targeting cryptographic modules. Depending on the way measured data is analyzed, the scientific literature usually distinguishes between simple side-channel attacks (SSCA) and advanced or differential side-channel attacks, as presented in (Clavier et al., 2010; Bauer et al., 2013; Zhou and DengGuo, 2005). (Clavier et al., 2010) proposes a refinement of this categorization distinguishing between horizontal and vertical side-channel analysis. This approach takes into account whether a single power curve is analyzed or the same time sample is analyzed in different execution curves. Bauer et al. also perform an extensive study on side-channel analysis and propose a SCA taxonomy (Bauer et al., 2013). It contains three classification categories. The first considers whether the attack is a simple or an advanced SCA. The leakage type is included as a second category. The third one contains information regarding whether an attack is profiled or not. However, the proposed classifications are oriented towards power and electromagnetic analysis attacks on cryptographic modules, and are not extensive enough to take into consideration the general environmental context. This makes them inapplicable for the classification of side-channel attacks in the virtualized environment.

Anderson et al. propose a classification for attacks on cryptographic processors in (Anderson et al., 2006). Although their work does not explicitly focus on side-channel attacks, most of the adversary scenarios they mention as examples for the proposed classification fall into this category. In addition to the categories proposed by Anderson et al., our framework takes into consideration the traditional classification of SCAs existing in the literature dividing them into active and passive (Zhou and DengGuo, 2005), as well as the distinction between trace-driven, access-driven and time-driven SCAs, as proposed by Zhang et al. in (Zhang et al., 2012). (Kim et al., 2012) , on the other hand, considers only trace-driven and time-driven attacks and further refine them in active and passive. All the proposed categorizations and taxonomies in this area contribute to the research field of side-channel attacks, however they do not consider the characteristics of the execution environment or under which conditions a specific attack can be performed or the limitations therein. To this end, our work extends the state-of-the-art in the field by proposing a classification that is general enough to include the existing approaches and to address their limitations.

Feasibility Assessment of SCAs. (Mowery et al., 2012) expressed their doubts about the feasibility of AES cache timing attacks on the x86 architectures. Their research has been inspired by an unsuccessful attempt to conduct a side-channel attack using the cache as a channel. They argue that the existing preventive mechanisms and technological advances make it impossible to conduct the specified attack on x86 architecture. This work addresses a specific approach and considers one type of attack aiming at compromising the confidentiality of a victim when executing AES encryption. We are unaware of the existence of a generic feasibility assessment methodology focusing on the threats resulting from the exploitation of covert channels in VE. (Xu et al., 2011) argue that depending on the bit rate of the covert channel exploited, the attack might be harmless. They make this valuable observation relying on information provided in (Department of Defense, 1985). Additionally, they specify different factors that can influence the bandwidth of a side-channel e.g., hardware specification, workloads in other VMs on the same physical host, hypervisor configuration. We gain inspiration from their work and aim to continue their investigation by showing that different factors of the system impact the feasibility of the attack by modeling the execution environment.

(Zhou and DengGuo, 2005) aim at providing a feasibility evaluation for SCAs. However, the evaluation aspects they propose are not concrete and do not take into consideration the characteristics of the environment, but are rather generic. To the best of our knowledge, no research on feasibility assessment of SCAs in a VE that is extensive enough to consider contextual aspects exists.

3 CLASSIFICATION OF SCAS

We classify existing SCAs to facilitate their analysis and the easier assessment of security-related properties of the environment in which they are conducted. We classify SCAs defining 3 major categories as: (i) **approach**, (ii) **effect**, and (iii) **limitations**. We argue that the most important characteristics of an attack are determined by the way it is conducted, the effect it might have on the system under attack if it is successful, and the potential the attack has depending

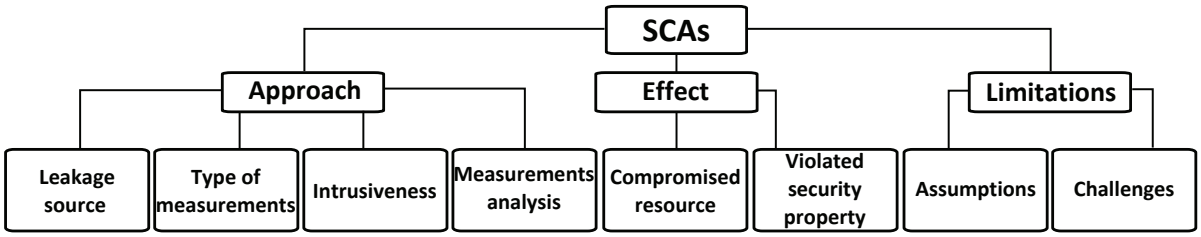

Figure 2: Overview of the proposed classification.

on the contextual limitations in terms of assumptions and challenges. Figure 2 gives an overview of the presented classification with a detailed explanation presented in the subsequent sections.

3.1 Approach

The approach describes the way the attacker targets compromising the isolation i.e., adversary's strategy. As shown in Figure 3, we differentiate the side-channel attacks in terms of the leakage source (i.e., the medium) being used to conduct the attack, the intrusiveness of the conducted attack, the type of measurements needed and the applied method for analysing the measured data.

3.1.1 Leakage Source

In terms of the medium that has been used for conducting the attack, we further refine the classification categorizing the SCAs into physical and architectural. The side-channel used in the case of architectural SCAs is an architectural component of the system e.g., virtual memory paging (Percival, 2005), L1 cache, used as a side-channel in (Zhang et al., 2012), L2 cache, exploited in (Xu et al., 2011), L3 cache, exploited in (Yarom and Falkner, 2013), etc. On the other hand, the physical SCAs use device components to conduct the attack e.g., power supply unit. Examples of attacks that exploit the power supply unit are given in (Hlavacs et al., 2011; Messerges et al., 1999).

3.1.2 Type of Measurements

Extending the classification of SCAs as presented in (Zhang et al., 2012), we differentiate the SCAs according to the type of measurements needed for the execution of the attack. We distinguish between SCAs considering device aspects, SCAs measuring timing information and SCAs using the characteristics of access patterns.

Device Aspects. Under observation in this case are physical device's aspects, such as power consumption, monitored for the conduct of the attacks (Hlavacs et al., 2011; Messerges et al., 1999), electromagnetic emanations, observed for the attacks (Agrawal et al., 2002; Carlier et al., 2004) or acoustic emanations, monitored for conducting (Song et al., 2001; Genkin et al., 2013). They are inspected while the device performs a specific operation e.g., cryptographic encryption. Measurements regarding the observed aspect are gathered and analyzed.

Timing Information. This class of attacks is known as time-driven attacks. A prerequisite for conducting them is that the execution times of the algorithm that is under attack have to be known in advance. Usually also the measurements have to be conducted many times to enable the statistical inference of information regarding the asset under attack. A representative of this class of attacks is described in (Kocher, 1996).

Access Pattern. This type of attacks is referred to as access-driven attacks. It exploits the information leaking from the usage of shared architectural assets, such as caches. Although time measurements might also be involved in this type of attacks, they are only used as a mean to infer information regarding the access pattern to the observed architectural component. Contrary to the SCAs where timing information is required, the timing measurements in this case are not necessarily precise. The attacks described in (Ristenpart et al., 2009; Percival, 2005; Zhang et al., 2012; Xu et al., 2011; Yarom and Falkner, 2013; Wu et al., 2012) are representative for this category.

3.1.3 Intrusiveness

We distinguish between intrusive and non-intrusive attacks, gaining insight from the widely-cited classes of active and passive side-channel attacks, as well as considering the distinction between invasive, semi-invasive and non-invasive side-channel attacks (Anderson et al., 2006).

Intrusive SCAs. The intrusive attacks require direct access to the internal components of the observed

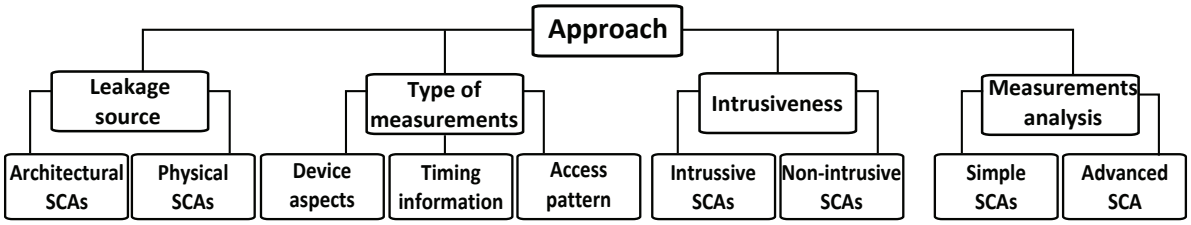

Figure 3: Detailed overview of the category "Approach".

device. They intervene with device's operation.

Non-intrusive SCAs. The non-intrusive SCAs, to the contrary, are passive attacks that only observe the operation of a device without intervening with it. In this case only externally available non-intentionally leaked information is exploited. Representative for this class of attacks are (Kocher, 1996; Song et al., 2001; Genkin et al., 2013; Agrawal et al., 2002; Ristenpart et al., 2009; Yarom and Falkner, 2013; Genkin et al., 2013), among others.

3.1.4 Measurements Analysis

We also distinguish the SCAs according to the way the gathered data is analyzed. In terms of the method used to evaluate the collected measurements, we differentiate between simple and advanced SCAs, as proposed in (Zhou and DengGuo, 2005).

Simple SCAs. A characteristic of this type of attacks is that usually they require a single trace to achieve their goal e.g., to obtain a secret key. A prerequisite for the successful conduct of this type of attacks is that the obtained information which is related to the attacked instructions needs to be larger than the noise. (Yarom and Falkner, 2013) manage to recover a significant part of a secret key by capturing a single decryption or signing operation.

Advanced SCAs. This type of SCAs considers the correlation between the processed data and the side-channel information. As this correlation is typically very small, a lot of measurements are needed, and statistical methods have to be applied for their evaluation. (Xu et al., 2011) is representative for this class of attacks.

3.2 Effect

We differentiate the attacks according to the effect they have on the system considering the asset under attack and the security property that has been violated. The optional subclass Effectiveness aims to provide information regarding how effective the attack is. These subcategories are explained in more detail below.

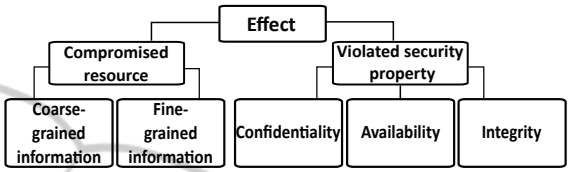

Figure 4: Detailed overview of the category "Effect".

3.2.1 Compromised Resource

This subcategory refers to the resources under attack, or which assets are the main goal of the attack. Here we can differentiate whether the attacker is targetting compromising fine-grained information, coarse-grained information or non-classified data. Ristenpart et al. for instance managed to gain coarse-grained information detecting activity spikes in Cloud scenario in (Ristenpart et al., 2009). On the other hand, Zhang et al. were the first to demonstrate a side-channel attack in a VE which manages to extract fine-grained data, namely an ElGamal key in (Zhang et al., 2012). Although the borders between these terms are rather obscure, we consider as fine-grained information e.g., an encryption key. Coarse-grained is the information that can be used as a basis to perform an attack. This information is usually not so specific. It could be probable location of a virtual machine in the Cloud, or the activity spikes of some specific machine.

3.2.2 Security Property

This subcategory takes into account which security property has been violated - confidentiality, availability or integrity.

Confidentiality. In case the attacker exploiting a side-channel manages to extract information regarding the victim, we consider that the confidentiality of the victim has been compromised in spite of the fact that the success of the SCA will not necessarily lead to leaking confidential information. Referring to the

state-of-the-art analysis we conducted, we consider that the primary goal of the SCAs is to compromise the confidentiality of the victim. Depending on the methods involved in pursuing this, however, some attacks might also affect other security properties, as described below.

Availability. We argue that conducting a SCA can indirectly affect or even compromise the availability of the system, even though this is not the primary goal of the attack. An example for that is given in (Zhang et al., 2012) where resources in terms of CPU are taken away from the victim (the victim's VM is frequently preempted) when conducting the attack.

Integrity. It is rather unlikely that integrity of the system will be affected through a side-channel attack, but for the purpose of completeness we add it as a category to the classification.

3.3 Limitations

The limitations on the way of conducting a SCA can be related to challenges (e.g., due to preventive mechanisms), or assumptions which have to be made for the system under attack so that the attack is successful. The classification of the limitations can help us estimate the potential for success of a certain attack depending on the prerequisites which have to be taken into account in advance, before the actual exploit of the side-channel. A more detailed explanation is given below.

3.3.1 Assumptions

The state-of-the-art in the field of side-channel attacks shows that typically the approaches for exploiting a side-channel proceed on a variety of assumptions. Classifying these assumptions in a systematic way helps the better comprehension of the attack and the elaboration on whether these assumptions are realistic or not for the real-world or for some predefined specific scenario.

Access Level to Shared Resources. This is a common prerequisite for all side-channel attacks. The side-channel that links the attacker to the victim should be present in order to be able to conduct the attack. Different attacks, however, require different access to the system under attack. We distinguish between physical access, proximity to the physical device and access to architectural components.

- Physical access - some of the side-channel attacks require having a physical access to the device hosting the victim so that measurements of different physical aspects of the system under attack can take place. An example of such an attack is given in (Messerges et al., 1999) describing how the power dissipated by the smartcard is monitored at the ground pin of the smartcard. For this, the attacker needs to attach a resistor to the device.
- Proximity to the physical device - for other attacks it is sufficient to have a physical proximity to the device without accessing it directly. One example is measuring the electromagnetic emanation as described in (Agrawal et al., 2002). In this attack in order to measure the induced emanations, placing probes as close as possible to the device is a prerequisite. Another representative for a SCA that requires proximity to the device under attack is the one described in (Genkin et al., 2013). In this case a microphone has to be placed near the physical device while performing cryptographic operations to record the acoustic emanations.
- Architectural access - depending on the approach, it might suffices that the attacker has a remote access to some architectural component e.g., use the same CPU on the Cloud as the victim. In the multicore architectures this can be considered as a challenge under certain circumstances. Among the attacks that exploit the architectural access to shared components are the ones presented in (Ristenpart et al., 2009; Xu et al., 2011; Yarom and Falkner, 2013; Zhang et al., 2012).

Required Knowledge. Having some previous knowledge about the system under attack is also often a precondition for conducting a side-channel attack.

- Training data - to the best of our knowledge, most of the demonstrated side-channel attacks require having training data (Zhang et al., 2012). For conducting a SCA based on the recorded acoustic emanations of a computer, the authors of (Genkin et al., 2013) also need previosly gathered information in order to map an acoustic pattern to the bits of the private key. Acquiring training data might be challenging.
- Acquaintance with the system under attack - for the successful conduct of a side-channel attack the attacker needs to be aware of the characteristics of the system under attack, and to take them into account when implementing the attack (Mowery

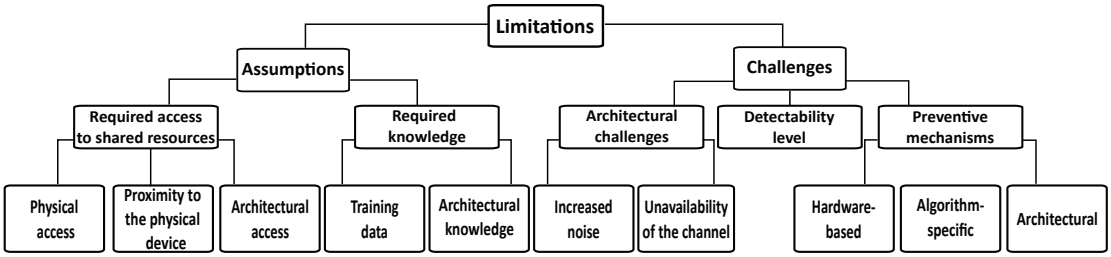

Figure 5: Detailed overview of the category ”Limitations”.

et al., 2012). For instance, in the case of a cache-based SCA, the attacker can tweak to use the first level cache or the last level cache depending on whether the victim is using the same CPU core or not. Without having this knowledge, the attacker might make false assumptions regarding the system under attack which can result in an unsuccessful attack.

3.3.2 Challenges

Researchers face a variety of challenges on the way of conducting side-channel attacks. They can be architectural, due to preventive mechanisms, or related to the intervention level with the victim the attack requires. More detailed classification is provided below.

Architectural Challenges. We distinguish between architectural challenges that affect the noise in the channel and might harden the attack or result in easier exploit, and architectural challenges that affect the availability of the channel.

- Noise - there are a variety of factors that can affect the noise in the channel. It might be increased e.g., due to scheduling policies, interference with other processes for the shared resources, core migration in a SMP system, etc., as identified in (Ristenpart et al., 2009; Percival, 2005; Zhang et al., 2012; Xu et al., 2011; Yarom and Falkner, 2013; Wu et al., 2012; Mowery et al., 2012). (Kocher, 1996) also faces the problem of noise and define it as ”timing variations due to unknown exponent bits”. For the attack described in (Genkin et al., 2013), possible sources of noise can be e.g., acoustic emanations from other machines near the microphone or emanations from the device under attack, but due to operations that are not of interest.
- Channel’s unavailability - different factors can also influence the availability of the channel such as scheduling the attacker’s and victim’s processes to different CPU cores for the whole duration of the attack. (Mowery et al., 2012) mention core pinning which results in unavailability of the side-channel as one of the reasons for unsuccess of their attack. In (Genkin et al., 2013) the channel might become unavailable if for example the recording microphone is removed or gets broken.

Detectability Level. Characterizes the intervention level a specific side-channel attack requires and is related to the probability that the attack will be detected. Some side-channel attacks require preempting the victim in order to be able to conduct the required measurements (Zhang et al., 2012). Keeping low penetration rate is a prerequisite for the success of the attack.

Preventive Mechanisms. Due to different preventive mechanisms some side-channel attacks are hardened or even made impossible. These are classified as follows.

- Hardware-based - special hardware e.g., tamper resistant crypto modules might be employed to secure the system against a given class of side-channel attacks. Hardware-based preventive mechanism characterizes the context of the attack described in (Mowery et al., 2012). The preventive measures described in (Tiri et al., 2005; Ratanpal et al., 2004) are also representative for this class. Acoustic shielding can be applied as a hardware-based countermeasure for the attack described in (Genkin et al., 2013).
- Algorithm-specific - This applies mainly to the side-channel attacks targeting to compromise e.g., cryptographic keys. The attack can be hardened if algorithm-specific measures are applied. For instance, to protect the system from a cache-based SCA and to keep the keys used in an AES algorithm confidential, the AES instructions can be explicitly moved out of the cache. This will affect the performance of the algorithm, but will increase the security.
- Architectural - measures might be applied against SCAs related to specific architectural components e.g., the cache. (Kim et al., 2012) is a representative in this class of protection mechanisms. The

authors describe how to avoid cache-based SCAs by managing a set of locked cache lines per core that are never evicted from the cache. In that way a VM can hide memory access patterns.

4 FEASIBILITY OF SCAS

For the rest of this paper we focus on the feasibility analysis of cache-based SCAs in a VE given a specific context. We argue that contextual aspects can undoubtedly influence the attack by hardening it e.g., multicore environment can result in frequent migration of the victim among different cores, or by facilitating it e.g., enabled simultaneous multithreading can lead to easier deployment of side-channel attacks. The feasibility analysis is done with respect to the system model (cf. Figure 1) presented in Section 1, and based on the ”Challenges”-category from the classification we proposed. We describe a set of factors that have impact on the cache-based SCAs in a VE taking into account our investigation of the demonstrated side-channel attacks in the community. Based on how the presented factors affect the SCAs, we propose an initial approach to estimate the feasibility of a cache-based side-channel attack in a specific VE. A proof of concept based on demonstrated SCAs is provided to support our feasibility analysis. Future works (cf. Section 5) will focus on further validating and refining the analysis presented in this section.

4.1 Feasibility Factors

The classification we proposed in Section 3 and more precisely the class ”Challenges” servers as groundwork for the feasibility analysis we conduct. It can help us identify which are the factors turning an attack into a more, respectively less feasible one with regard to the context. The feasibility factors we are investigating are characteristics of the context in which the SCA is to be deployed and are described in more detail below.

4.1.1 Architectural Challenges

With respect to the architectural challenges, we distinguish between challenges having impact on the noise in the channel and challenges influencing the availability of the channel.

Noise. In this paper we refer to noise as the measurements related to data which is in the cache shared by the attacker’s and the victim’s VMs, but has no relevance to the data the attacker is interested in. Possible sources of noise for a cache-based side-channel attack in a virtualized environment are given below.

- Noise due to synchronization - basically the victim and the attacker are alternately using the side-channel. If they are not properly synchronized, the noise in the channel might increase dramatically, as the attacker will acquire measurements that are either redundant or not related to the victim. A proper synchronization is very challenging and highly depends on the implementation of the attack and the capabilities of the attacker.
- Noise due to scheduling - although it has relation to the noise due to synchronization, it depends more on the hypervisor’s configuration and the used scheduling policies rather than on the capabilities of the attacker.
- Noise due to interference with other VMs - in a VE it might happen than also third parties are using the same cache as the attacker and the victim. In this case, the attacker will have to sort out the measurements that are related to the third parties rather than the victim. The number of VMs sharing the side-channel can also affect the noise.
- Noise due to workload on the victim’s side - the attacker might be interested in a part of victim’s operations, but there is no guarantee that the acquired measurements are not related to other operations the victim is conducting that are not of interest to the attacker. In such a case the noise in the channel will be increased.
- Noise due to core migration - in simultaneous multiprocessing systems the virtual CPUs of the victim’s or attacker’s VM might be floated among different physical CPU cores. The attacker might remain unaware of this core migration which will also affect negatively the noise in the channel. It holds that the bigger the number of CPU cores is, the higher the probability of additional noise in the channel is.
- Noise due to hardware features - e.g., due to prefetching or CPU power saving. Prefetchers are designed to increase performance by speculating about future memory accesses. As modern prefetchers are complex and poorly documented, their use increases the noise in the covert channel, and filtering out the noise due to prefetching is rather challenging.
- Disabled simultaneous multithreading (SMT) - If SMT is enabled the processor resources such as caches are shared between threads. This results in a easily used side-channel between threads (Percival, 2005). We consider an environment with

enabled SMT as more prone to SCAs than an environment with disabled SMT i.e., SCAs are more feasible if SMT is enabled.

Channel's Availability. We consider the channel to be available in the cases when the attacker's and the victim's VMs are using the same cache. In the case of L1 cache-based SCA, which is private per core cache, channel's availability means that both VMs are running on the same processing core at least at some point in time. In cases when the VMs share the cache only for a limited amount of time, we regard the channel as partially available. Different factors might have impact on the channel's availability such as scheduling policies (e.g., core pinning vs. load balancing; work-conserving vs. non work-conserving), number of CPU cores (multicore vs. single core), as well as the frequency of interrupts allowed (e.g., Interprocessing interrupts).

4.1.2 Detectability Level

Here, as described in the proposed classification, we consider the potential that the SCA is detected and differentiate between detectability from hypervisor's perspective and detectability from victim's perspective. The frequency of preempting the victim (preemption rate) can be used as a possible measure for detectability level.

4.1.3 Preventive Mechanisms

With respect to the preventive mechanisms we distinguish between hardware-based, algorithm-specific and architectural.

Hardware-based - Here we consider special hardware that has been deployed to protect the system against the relevant type of attack e.g., tamper resistant crypto modules might be employed to secure the system against a SCA aiming at breaking AES encryption.

Algorithm-specific - We consider this case if there are some measures applied to protect exactly the algorithm that is a target of the attack - e.g. move AES instructions out of the cache.

Architectural - this case is considered if it is known that the cache implementation provides some mechanisms for protection against side-channel attacks.

4.2 Feasibility Assessment

Our goal is to present the initial steps of how to estimate the feasibility of a cache-based SCA in a VE. For this purpose, we model the SCAs as a feasibility tree gaining insight from the attack trees presented in (Schneier, 1999), and taking as a basis the "Challenges" class of the proposed classification and the feasibility factors presented in Section 4.1. Figure 6 depicts the created feasibility tree. Hereby, we consider that an infeasible SCA can be represented by "1" at the root of the attack tree which is namely the "Challenges" category. We argue that with the increase of the described challenges the feasibility of the cache-based side-channel attack in a VE will decrease. From security perspective increasing the challenges can be seen as a protection goal and the proposed model - as guidelines what can be done to make a cache-based SCA less feasible. As proposed in the classification, the "Challenges" category is further refined into the categories "Architectural challenges" (subdivided into "Noise" and "Unavailability of the channel"), "Detectability" (divided into "Hypervisor's detectability" and "Victim's detectability") and "Preventive mechanisms" (subdivided into "Hardware-based preventive mechanisms", "Algorithm-specific preventive mechanisms" and "Architectural preventive mechanisms"). We model these three subcategories as follows:

- Architectural challenges - "1" means too much noise or unavailability of the channel.
- Detectability - "1" represents that the attack can be detected either by the victim or by the hypervisor.
- Protection mechanisms - "1" means that there are protection mechanisms applied to secure the system against cache-based side-channel attacks.

The same idea is applied to the rest of the tree. In case the property or the characteristic depicted in the leaf is present, we model it by "1" in our tree, otherwise - by "0". For example, if many processes are running in the victim's VM, and the attacker is interested only in the cache access pattern of one of them, the respective subcategory which models the noise due to victim's workload in the tree will be represented by "1". Having described how to construct the feasibility tree, we have to model the relationships between the categories in order to be able to estimate the feasibility of a given attack. This is important, as the information is available only at leaves-level, and based on this information we want to derive a conclusion regarding the feasibility of the attack (represented at the root of the tree). The relationship between the subcategories

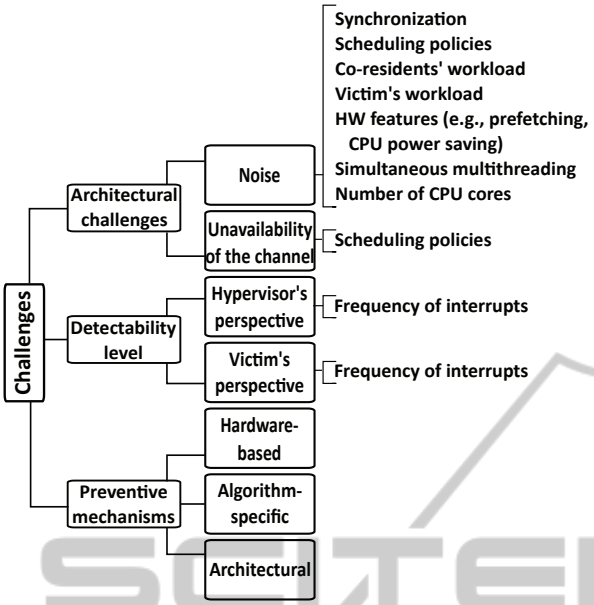

Figure 6: Feasibility tree of a SCA in a VE.

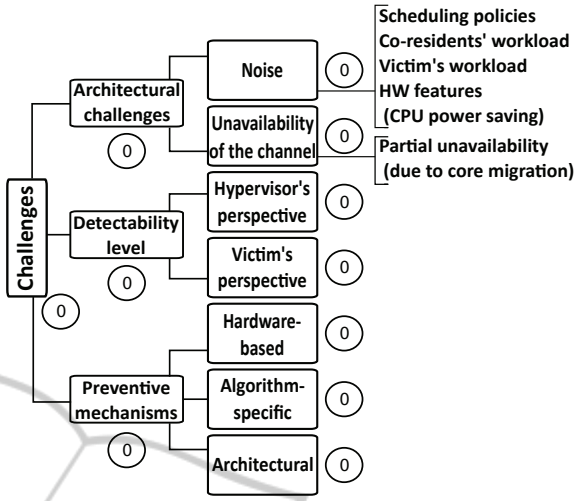

Figure 7: Classification of the challenges of Attack 1.

of the root can be modeled as OR-relationships, as we speculate that (i) if there is a mechanism applied to protect the system against this specific type of attack, or (ii) if the architectural challenges are present to large extend i.e., too much noise or constantly unavailable channel, or (iii) if the attack is detectable by the hypervisor (under the assumption that the hypervisor is not malicious) or the victim, the SCA is too challenging or infeasible. In other words, one of these conditions must hold, for the attack to be infeasible. Following the same intuition, we model the relationships at the next tree-level as OR-relationships. We argue that it suffices that there is too much noise in the channel and the relevant measurements cannot be filtered out, or that the channel is constantly unavailable for the attack to be infeasible. In addition, we believe that if a preventive mechanism is applied to protect the system against cache-based SCAs it will be infeasible to deploy such an attack.

4.3 Proof of Concept

As a proof of concept for the proposed approaches, we refer to the state-of-the-art in the field of side-channel attacks. We consider some of the most prominent side-channel attacks being demonstrated recently in virtualized environments to (i) show that they can be classified as proposed in Section 3 and to (ii) analyze their feasibility applying the guidelines explained in Section 4. We consider three distinct attacks - the work described in (Zhang et al., 2012) referred to as Attack 1, the attack described in (Mowery et al., 2012) referred to as Attack 2 and a Cloud-based scenario in which the attacker and the victim do not share the same physical resources as Attack 3 for the rest of this paper. For brevity we classify only the challenges the authors have identified when conducting the respective attacks.

Following the guidelines, given in Section 4, first we model the three attacks as feasibility trees. The results are given in Figures 7, 8 and 9. As can be seen from the figures, whereas in Attack 1 the authors do not mention the existence of any preventive mechanisms applied to protect the system against side-channel attacks, in Attack 2 the adversary faces the problem of a hardware-based protection mechanism. This is also the main difference in terms of challenges between the first two attacks' contexts. In Attack 1 the authors try to cope with the various *Architectural challenges* due to *increased noise* or *partial unavailability* of the side-channel. Since the unavailability of the channel is partial and the noise is not so much that it cannot be filtered out, the attack from Attack 1 is a successful one. From the challenges the authors face when conducting it, we identify the sources of noise and unavailability of the channel, as shown in Figure 7. On the one hand, the attacker has to cope with interference from other processes from the victim's VM and other VMs which we classify as *noise due to workload*. On the other hand, the adversary has to filter out the *noise resulting from the scheduling*. Beyond that, the observed *CPU is floating among cores* and the availability of the channel is not always given. As the attacker manages to solve this problem using interprocess interrupts, we model this as *partial unavailability*. Apart from that, the authors do not mention to face problems due to detectability or protection mechanisms specially applied to protect the system against cache-based side-channel attacks.

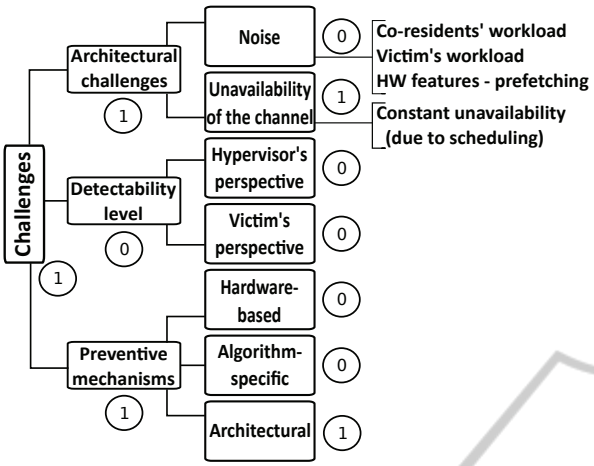

Figure 8: Classification of the challenges of Attack 2.

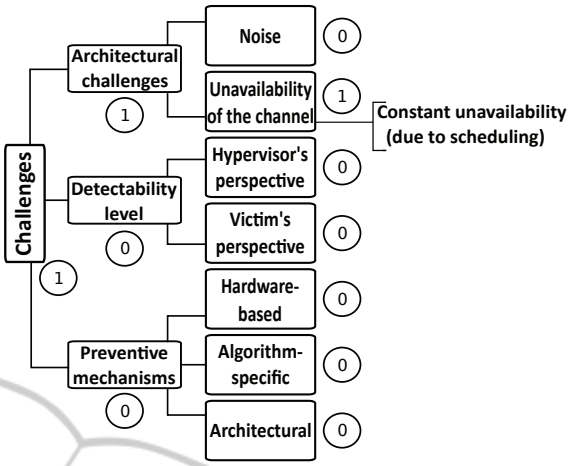

Figure 9: Classification of the challenges of Attack 3.

The main problem that the adversary faces in Attack 2 is the *hardware-based protection*. The authors of (Mowery et al., 2012) argue that an AES cache-based side-channel attacks are becoming infeasible namely due to (i) the AES-NI (Intel Corporation, 2010) and (ii) the multicore processors. Following the notion of our framework, we share their opinion considering AES-NI as a hardware-based protection measure against the specific type of attack which leads to infeasibility of the attack. As described in the previous section, the *multicore processor systems* can make an attack infeasible if their usage leads to *constant unavailability* of the channel. This applies to the case described by Mowery et al. in which the scheduler can pin the virtual machines to different cores. If the channel is partially unavailable, as in Attack 1, this challenge can be overwhelmed. This attack also faces challenges related to *noise due to prefetching* or *workload*, as shown in Figure 8, but the main problem is the hardware-based protection. In Attack 3 we have a Cloud and 2 VMs distributed to different physical machines. Even if we assume that there are no sources of noise, or any protection mechanisms, in this case the attack is infeasible due to the *constant unavailability of the channel* (cf. Figure 9).

5 DISCUSSION

The isolation the VE provides among the virtual machines is not considered secure anymore due to the demonstration of sophisticated side-channel and covert-channel attacks managing to compromise the confidentiality in the virtualized environment. Depending on the environment, however, these attacks are more respectively less feasible. This paper classifies existing side-channel attacks taking into consideration the characteristics of the context and provides an initial approach for a feasibility analysis for cache-based side-channel attacks. This research will be extended by identifying additional feasibility factors and their impact on the exploitability of a side-channel as well as by introducing the notion of weights as a way to prioritize the most important feasibility factors. Furthermore, a quantitative framework for estimating the risk related to side-channel attacks in a specific environment can be developed.

ACKNOWLEDGEMENTS

Research supported by TU Darmstadt's project LOEWE-CASED and the Deutsche Forschungsgemeinschaft Graduiertenkolleg 1362 - DFG GRK 1362.

REFERENCES

Agrawal, D., Archambeault, B., Rao, J., and Rohatgi, P. (2002). The EM Side-Channel(s). In *CHES*, volume 2523 of *LNCS*, pages 29–45. Springer-Verlag.

Amazon Web Services (2014). Amazon Virtual Private Cloud User Guide- Dedicated Instances. http://awsdocs.s3.amazonaws.com/VPC/latest/vpc-ug.pdf.

Anderson, R., Bond, M., Clulow, J., and Skorobogatov, S. (2006). Cryptographic Processors-a survey. *Proceedings of the IEEE*, 94(2):357–369.

Bauer, A., Jaulmes, E., Prouff, E., and Wild, J. (2013). Horizontal and vertical side-channel attacks against secure RSA implementations. In *CT-RSA*, pages 1–17. Springer-Verlag.

Carlier, V., Chabanne, H., Dottax, E., and Pelletier, H. (2004). Electromagnetic Side Channels of an FPGA Implementation of AES. *IACR Cryptology ePrint Archive*, page 145.

Clavier, C., Feix, B., Gagnerot, G., Roussellet, M., and Verneuil, V. (2010). Horizontal Correlation Analysis on Exponentiation. In *International Conference on Information, Communications and Signal Processing*, LNCS. Springer-Verlag.

Department of Defense (1985). Trusted Computer System Evaluation Criteria. Technical Report DoD 5200.28-STD, National Computer Security Center, Ft. Meade, MD 20755. Also known as the "Orange Book".

Figueiredo, R., Dinda, P. A., and Fortes, J. (2005). Guest Editors' Introduction: Resource Virtualization Renaissance. *Computer*, 38(5):28–31.

Genkin, D., Shamir, A., and Tromer, E. (2013). RSA Key Extraction via Low-Bandwidth Acoustic Cryptanalysis. Cryptology ePrint Archive, Report 2013/857 http://eprint.iacr.org/.

Hlavacs, H., Treutner, T., Gelas, J. P., Lefevre, L., and Orgerie, A. C. (2011). Energy Consumption Side-Channel Attack at Virtual Machines in a Cloud. In *International Conference on Cloud and Green Computing (CGC 2011)*.

Intel Corporation (2010). Secure the enterprise with Intel AES-NI. http://www.intel.com/content/www/us/en/enterprise-security/enterprise-security-aes-ni-white-paper.html. Last accessed on 22.04.2014.

Kim, T., Peinado, M., and Mainar-Ruiz, G. (2012). STEALTHMEM: system-level protection against cache-based side channel attacks in the cloud. In *USENIX Security symposium*, pages 11–11. USENIX Association.

Kocher, P. C. (1996). Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems. In *CRYPTO*, pages 104–113. Springer-Verlag.

Li, P., Gao, D., and Reiter, M. K. (2013). Mitigating access-driven timing channels in clouds using StopWatch. In *DSN*, pages 1–12. IEEE.

Marty, M. and Hill, M. (2007). Virtual hierarchies to support server consolidation. *SIGARCH Comput. Archit. News*, 35(2):46–56.

Mell, P. and Grance, T. (2009). The NIST Definition of Cloud Computing. Technical Report 800-145, National Institute of Standards and Technology (NIST).

Messerges, T., Dabbish, E., and Sloan, R. (1999). Investigations of power analysis attacks on smartcards. In *USENIX WOST*, pages 17–17. USENIX Association.

Mowery, K., Keelveedhi, S., and Shacham, H. (2012). Are AES x86 cache timing attacks still feasible? In *CCSW*, pages 19–24. ACM.

Padala, P., Zhu, X., Wang, Z., Singhal, S., and Shin, K. (2007). Performance Evaluation of Virtualization Technologies for Server Consolidation. Technical Report HPL-2007-59, HP Laboratories Palo Alto.

Pearce, M., Zeadally, S., and Hunt, R. (2013). Virtualization: Issues, security threats, and solutions. *ACM Comput. Surv.*, 45(2):17:1–17:39.

Percival, C. (2005). Cache missing for fun and profit. In *The technical BSC Conference (BSDCan)*.

Popek, G. and Goldberg, R. (1974). Formal requirements for virtualizable third generation architectures. *Commun. ACM*, 17(7):412–421.

Ratanpal, G. B., Williams, R., and Blalock., T. (2004). An on-chip signal suppression countermeasure to power analysis attacks. *Dependable and Secure Computing*, 1(3):179–189.

Ristenpart, T., Tromer, E., Shacham, H., and Savage, S. (2009). Hey, you, get off of my cloud: exploring information leakage in third-party compute clouds. In *CCS*, pages 199–212. ACM.

Schneier, B. (1999). Attack trees. *Dr. Dobb's Journal*, 24(12):21–29.

Song, D. X., Wagner, D., and Tian, X. (2001). Timing Analysis of Keystrokes and Timing Attacks on SSH. In *USENIX Security Symposium*, pages 25–25. USENIX Association.

Stefan, D., Buiras, P., Yang, E., Levy, A., Terei, D., Russo, A., and Maziéres, D. (2013). Eliminating Cache-Based Timing Attacks with Instruction-Based Scheduling. In Crampton, J., Jajodia, S., and Mayes, K., editors, *ESORICS*, volume 8134 of *LNCS*, pages 718–735. Springer-Verlag.

Tiri, K., Hwang, D., Hodjat, A., Lai, B., Yang, S., Schaumont, P., and Verbauwhede, I. (2005). A side-channel leakage free coprocessor IC in 0.18 μm CMOS for embedded AES-based cryptographic and biometric processing. In *Design Automation Conference*, pages 222–227.

Uddin, M. and Rahman, A. A. (2010). Server consolidation: An approach to make data centers energy efficient and green. *International Journal of Engineering and Scientific Research*, 1.

Wu, Z., Xu, Z., and Wang, H. (2012). Whispers in the hyper-space: high-speed covert channel attacks in the cloud. In *USENIX Security symposium*, pages 9–9. USENIX Association.

Xu, Y., Bailey, M., Jahanian, F., Joshi, K., Hiltunen, M., and Schlichting, R. (2011). An exploration of L2 cache covert channels in virtualized environments. In *CCSW*, pages 29–40. ACM.

Yarom, Y. and Falkner, K. (2013). Flush+Reload: a High Resolution, Low Noise, L3 Cache Side-Channel Attack. *IACR Cryptology ePrint Archive*.

Zhang, Y., Juels, A., Reiter, M., and Ristenpart, T. (2012). Cross-VM side channels and their use to extract private keys. In *CCS*, pages 305–316. ACM.

Zhou, Y. and DengGuo, F. (2005). Side-Channel Attacks: Ten Years After Its Publication and the Impacts on Cryptographic Module Security Testing. Cryptology ePrint Archive, Report 2005/388.