Inability of existing security models to cope with data mobility in dynamic organizations

Trajce Dimkov, Qiang Tang, Pieter Hartel
{trajce.dimkov,qiang.tang,pieter.hartel}@utwente.nl

August 3, 2008

Abstract

In dynamic organizations, the mobility of data outside the organizational perimeter causes an increased level of threats such as the loss of confidential data and the loss of reputation. Some modeling tools, like Microsoft's TAM, play an important role in identifying threats in traditional IT systems. In these IT systems the physical infrastructure and roles are assumed to be static. We show that current modeling tools are not powerful enough to help the designer identify the emerging threats due to mobility of data and change of roles, because they do not include the mobility of IT systems nor the organizational dynamics in the security model. Researchers have proposed new security models that particularly focus on data mobility and the dynamics of modern organizations, such as frequent role changes of a person. We show that none of the new security models simultaneously considers the data mobility and organizational dynamics to a satisfactory extent. As a result, none of the new security models effectively identifies the potential security threats caused by data mobility in a dynamic organization.

1 Introduction

In the last decade three main trends have emerged. The first is information omnipresence raised by the increasing usage of mobile devices. As mobile technology improves, the market becomes rich with devices capable of complex computations, long battery life and vast storage. The increased usage of mobile devices allows data to be present everywhere. The second trend is the increasing usage of outsourcing. Highly trained specialist become scarce and thus expensive. Organizations cope with the scarcity problem through search for specialists in other countries, where access to highly trained cheap workforce is easier. Organizations gain access to the highly trained workforce by becoming decentralized and by outsourcing whole business processes and departments. The last trend is the increasing cooperation between organizations. To increase market share, organizations carry out joint projects with other organizations and extensively

hire part-time consultants. In this paper we consider outsourcing and networked organizations as dynamic organizations.

Information omnipresence increases the risk of attacks that include physical tampering with mobile devices. Outsourcing and networked organizations are dynamic, making the distinction of roles in an organization difficult to define and maintain, which leads to increased risk from social engineering attacks. In social engineering a person uses influence and persuasion to deceive an employee with the intention of detaining confidential information [1].

Researchers from the industry are aware of the increase of mobility and the impact mobility has on security [2, 3, 4]. A number of mechanisms, such as best practices of protecting against laptop theft and protecting information in laptops are proposed to help the organization mitigate the new threats [28, 29, 30, 31]. In parallel new hardware, such as tamper resistant dongles and hard drives with encryption capabilities are also introduced to cope with physical attacks, protecting the sensitive data in laptops and handheld devices [5, 32, 6]. All of the solutions partially restrict the data mobility and are based on best practice criteria.

A step towards understanding the security implications of the mobility of data in a dynamic organization is to create a model that includes the digital, physical and social aspect of the world. When an adversary tries to compromise a system, the adversary uses all available resources, which besides digital penetration include physical possession of a device and usage of social means to acquire sensitive information.

The digital, social and physical aspect are defined by Wieringa [7] and we quote his definitions below:

> The physical world is the world of time, space, energy and mass measured by kilograms, meters, second, Amperes, etc. The social world is the world of conventions, money, commercial transactions, business processes, job roles, responsibility, accountability, etc. structured in terms of conceptual models shared by people. At the interface between the social and physical worlds we have the digital world which consists of symbols that have a meaning for people.

Problem Information omnipresence, outsourcing and cooperation between organizations increases data mobility and role changes more than ever, making it increasingly difficult to secure the data.

Contribution We show that threats that arise from mobility of data in dynamic organization can not be presented with the existing threat modeling techniques. We define the requirements for an integrated security model and look in the literature at alternative models of the world that can represent the mobility of data in a dynamic organization. We (1) analyze state of the art security models using attack scenarios presented in a case study, (2) show that none of the new security models consider both of data mobility and organizational dynamics to a satisfactory extent, and (3) present requirements for an integrated model that addresses this deficiency.

The remainder of the paper is organized as follows. Section 2 provides a case study of current threats that include mobility of objects, interaction between person with a machine and interaction between two people. Section 3 introduces the requirements for an integrated security model of the world that is able to present the attacks presented in the case study. Section 4 presents the analysis of current models and shows to which extent the security models satisfy the requirements of the integrated security model. Section 5 concludes the paper.

2 Case study

To provide a focus for the analysis, we present a laptop case study. The case study looks at attacks that arise from the mobility of the laptop and social engineering. The laptop is a powerful mobile device which is extensively used by all organizations and which is capable of containing a vast amount of data.

The rest of this section summarizes two types of attacks. The first type of attack is based on permanent physical possession of the laptop and focuses on the confidentiality of the data stored inside. The second type of attack introduces social engineering as a way to provide access to the laptop and focuses on the integrity of the data in the laptop.

2.1 Confidentiality of the data in a laptop

Having physical possession of the laptop leads to a spectrum of attacks not possible through remote access. In a remote attack, the confidentiality of sensitive data mainly depends on the strength of the encryption key and the encryption algorithm. With the possibility of physical attacks on a laptop, another aspect comes into focus: the location where the encryption keys are stored.

If the adversary is in possession of the laptop, the adversary is also in possession of the encryption keys, making the storing of encryption keys in tamper resistant hardware crucial. The threat model of a storage device [8, 9] provides a variety of options for the adversary to consider, such as removal or tampering with parts of the device. If the keys are stored in the hard drive, the adversary starts from the easiest way, by physically unmounting the hard drive and attaching the hard drive to another laptop, effectively circumventing all protection provided by the host device. Another approach is to tamper with the hard drive and physically extract the keys. The need for a good protection of the encryption keys has become widely acknowledged after the coldboot attack [10], which is therefore worthy of further study.

To present the coldboot attack, we first introduce a simplified example of presenting encrypted data to a user as shown in Figure 1. The snapshot is taken from the TAM and modified (ex: numbers are added to present the sequence of the calls), to give a better overview of the example. The user presents to the operating system a key coupled with a request that defines the data the user wants to read (1). The operating system forwards the request to the hard drive (2) and recovers the encrypted data (3). Then, the encrypted data together

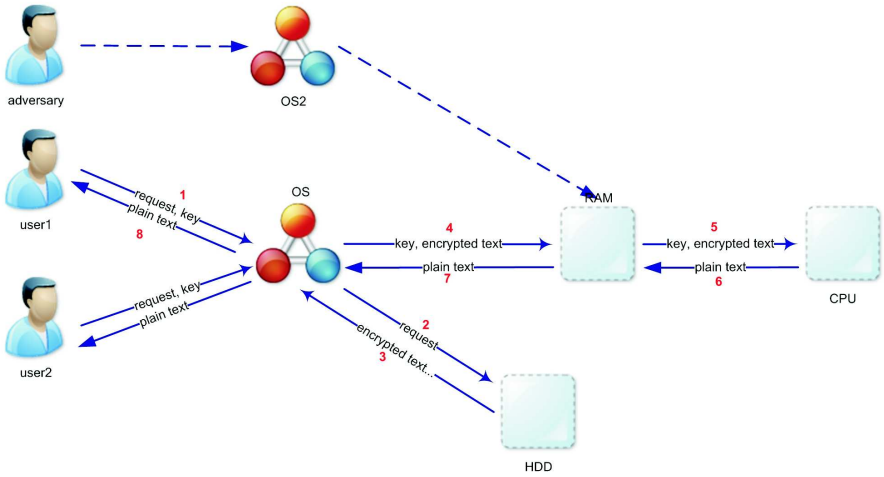


Figure 1: Coldboot attack

with the key is loaded into the RAM (4). From the RAM the data is fed into the processor (5), which as a result returns the plain text (6). The plain text is then sent to the user through the operating system (7,8). In the coldboot attack, the adversary does not target the hard drive with the sensitive information, but the RAM where the encryption keys are stored. The adversary steals the laptop while turned on, and freezes the RAM memory using canisters of compressed air to slow down the data degradation [11]. Then the adversary physically transfers the RAM to another computer, and dumps the memory on a hard drive. Later, the adversary has all the time needed to use search algorithms on the dumped memory to get the encryption keys.

To model the coldboot attack and physical tampering with devices, we need to be able to model the tamper resistance of components in a laptop. We also need to present the removal/addition of components in the laptop. The properties of these attacks led us to searching for security models that are able to present these features. The models we found are presented in Section 4.

2.2 Rootkit attacks on a laptop using social engineering

Stealing a laptop provides an instantaneous benefit to the adversary. However, installing malware that sends data periodically from the internal network of the organization to the adversary is more dangerous. To infect the network, the adversary needs to combine social engineering with malicious software such as rootkits, making the mobile device an excellent carrier of the malicious software. A rootkit [33] is software that hides itself and other files from diagnostic and security software and is used in a bundle with viruses, Trojans and other malicious software. Because the rootkit is hard to detect, the infected laptop can periodically harvest data from the internal network of the organization and

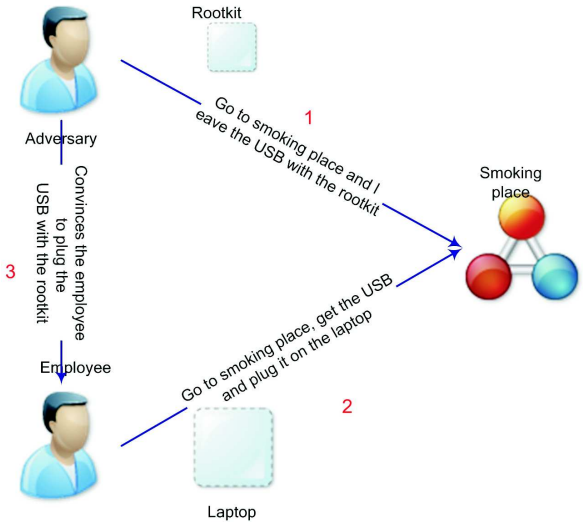


Figure 2: Road apple attack

send the data to the adversary without being noticed. A rootkit can be installed on the ROM of any peripheral device [34], in the ACPI tables in the BIOS [35] or in the RAM of the laptop [36].

There are several ways an adversary can use to install a rootkit on a laptop. Here we consider two variations of the road apple attack as presented in Figure 2. The term road apple refers to an apple that is found on a road, tempting the finder to take it. In the IT world, the apple is usually an infected generic dongle with the logo of the organization left by the adversary in a social place (1) of the organization, such as a cafeteria. When an employee finds the dongle (2) he may be tempted to plug the dongle into his laptop [37]. In this version of the road apple attack the adversary has no direct interaction with the employee.

Another approach by the adversary to realize the road apple attack is through direct interaction with the employee (3). For example, the adversary impersonates higher level management and builds a trust relationship with the employee. The adversary provides fake identity and simulates an emergency, asking to send a file he has on a dongle through the laptop of the employee. If the employee plugs the dongle on the laptop, the dongle will install the rootkit without the employee's knowledge [12].

The road apple attack, as many other social engineering attacks [1] relies on activities occurring in the digital, physical and social world. Thus, we need a model able of presenting movement and roles, as well as physical and digital objects.

3 Integrated security model of the world

Motivated by the attacks described in detail in the previous section we did an exhaustive literature search for models that are capable of presenting the attacks. Most of the models we found focus on modeling the data from the digital aspect(ex. data flow) and only a limited number of models consider the location of the data. To the best of our knowledge there is no integrated security model which includes all three aspects(digital, physical, social), and thus there is no model that can truthfully represent the security implications on data mobility in dynamic organizations.

A model that will enable a security expert to observe the level of security of mobile data in dynamic organizations will give the security expert better insight in the threats and attack vectors, leading to an understanding of what kind of threat mitigation the security expert needs to implement. An integrated security model of the world will be a testbed for the effectiveness of the proposed mitigations. Here we provide requirements of an integrated security model of the world from the digital, social and physical aspect, together with the basic building blocks the model needs to include.

The requirements we want an integrated security model to achieve are:

1. *The model should be capable of representing the data of interest.*
2. *The model should be capable of representing the physical objects in which the data resides.*
3. *The model should be capable of representing the roles a user can have.*
4. *The model should define the interactions between the data, physical objects and the roles.*

The first three requirements present the digital, physical and social aspect of the world, while the last binds them together. Following the requirements and the definitions of the physical, digital and social aspect, elements of interest in the integrated security model are: *data*, *physical objects*, *roles* and *interaction relations*.

From the digital aspect represented by data objects, we believe that the integrated model needs to present the data at rest as well as data in movement. The spatial/temporal characteristic provides information about the movement of the objects which is needed to model the attacks presented in Section 2. To present tampering with a device, the model should be capable of presenting the physical properties of an object including the boundary of the object. From the social aspect we are interested in the transition of one role to another, as well as the interaction between roles. Through role interaction and role transition we can present the impersonation of an adversary and adversary's direct interaction with an employee as presented in Section 2.2.

To predict the behavior of the system at any given time we need a state based model. Schneider [13] argues that a static model can not enforce security policies because the capability of a user can change over time. Goguen [14] presents

Aspect	Element	Property
Digital	Data	Static
		Dynamic
Physical	Object	Spatial/temp
		Resistance
Social	Role	Transition
		Interaction

Table 1: Properties of interest for an integrated model

capability state model to present dynamic changes in the system, and based on the changes of the capability of a user, defines dynamic security policies. Here, Goguen uses predicates defined over the sequences of operations used to reach the current state, instead of using a predicate on a single state.

In an integrated security model of the world, states or a sequence of states, should be classified based on the properties we want to model. One example is distinguishing the difference between states that are possible in the real world and states that are not. Another example is classification between states that cause violation of security policy and states that do not violate the security policy.

4 Security models

This section presents the current state of the art security models with which we try to model the attacks presented from the case study. We focus on models from the computer science domain modeling a security property of the system, such as privacy or confidentiality. The first model (Subsection 4.1) is static and used in the software industry for generation of threats for a specific software application. Then we move into dynamic, state based security models [15, 16] (Subsections 4.3 and 4.4) that include mobility of the components in the system. These dynamic models are all inspired by the ambient calculus [17], for which we provide the basic structure. Later we explore how ambient calculus is extended to focus on different properties of the world in two other security models. We analyze the characteristics of these models with respect to the requirements presented in section 3. A detailed analysis is presented in the appendixes.

4.1 Model used for threat generation in software development

One of the first steps when looking at a security issue is to present a threat model [18] analysis the capability of the attacker in an appropriate level of detail. The

model defines a set of undesired behaviors of the system called threats. The model does not specify how these threats could happen (which makes the model attack independent) but just recognizes the existence of such threats. Usually, the input of a threat model is the model of the system, and the output is a set of threats. This set is later used as an input for risk assessment and report generation. In the literature, threat modeling focuses on applications. The scientific community has worked on a formalization of threat modeling [19, 20] and produced algorithms for threat generation [21, 22] and sorting [23]. This led to a number of tools which partially automate the threat modeling and generation process space[38, 39]. Here we consider the Microsoft Threat Analysis and Modeling Tool(TAM) which is the state of the art tool used for internal threat generation and analysis in software development organizations.

In Figure 1 we use the TAM to model the coldboot attack. Besides being able to model data structures and data flow the tool also presents physical objects as well as roles. TAM considers the physical component and the role as static and the data as dynamic, allowing to the TAM threat generation algorithm to focus on the flow of data. Although this reasoning is understandable and valid in software modeling, in the presented attacks TAM proves to be restrictive. TAM does not take into consideration the possibility that a component can be removed, such as the RAM in the coldboot attack nor that a component is mobile, such as the dongle in the road apple attack.

TAM presents neither role interaction nor role transition. Because of the lack of states, even with manipulation of the relationships and entities in the model, TAM can not present interaction between roles and role transition. The role in TAM is used to describe the privileges over a component in an access control table, but does not define transition between roles such as escalation of privileges between a normal and an administrator role nor any interaction between roles as can be seen in discretionary security models, such as delegation or separation of duty. As a result, TAM can not present the road apple attack where the adversary has direct interaction with the employee.

TAM can not present physical properties of a component. A component is defined through the service type the component provides and the data and roles the component interacts with. Since TAM does not consider the component as a physical object, the component's resistance to physical attacks cannot be expressed in the model. Thus, TAM is not capable of presenting the tamper resistance of a component, nor any other physical characteristic. Due to the inability of presenting physical properties, TAM is also incapable of presenting the degradation of data in the RAM in the coldboot attack.

We can change the meaning of the components to present the attacks from the case study, but not without changing or blurring the relationship between the components. We can "attach" a new operating system to the RAM. As the number of mobile components increases the number of such "attachments" also increases, degrading the model usability as well as blurring or changing the meaning of the relationship between components. TAM model "attachments" are used in modeling the coldboot attack as presented in Figure 1.

4.2 Ambient calculus

Ambient calculus [17] provides an excellent apparatus for modeling a world with mobile components. The calculus is capable of presenting spatial and temporal properties of a component(with running processes inside) in the model and is Turing complete. Ambient calculus serves as an inspiration for the state of the art security models that consider mobility of components. The ambient calculus has been expanded into typed ambient calculus [24], boxed ambient calculus [25] etc. All of these calculi focus on a specific security property such as boundary interference [26]. The calculus syntax is presented in Table 2.

P,Q,R::=	processes	M::=	message
(vn)P,	restriction	n,	name
0,	void	in M,	can enter into M
P|Q,	composition	out M,	can exit out of M
!P,	replication	open M,	can open M
M[P],	ambient	ϵ ,	null
M.P,	capability action	M.M',	composite
(n).P,	input action		
<M>	output action		

Table 2: Ambient calculus syntax

Ambient calculus does not define the properties of an entity nor the relationship between entities, making the calculus generic enough to present any model of interest. The calculus presents an excellent theoretical framework for reasoning about mobility. But, without additional formal naming convention and definition of the properties of interest in the component, can not be directly implemented in any model on which mechanisms such as policies or threat generation algorithms need to be applied.

Ambient calculus can not present tampering with a device. In ambient calculus data decides to leave the device or not based on the capability of the data, which is not the case when an adversary tampers with a device. Although tamper resistance can be presented through a stack of ambients, the manipulation of the stack can not be done at run time, because any rearrangement or removal of a layer requires a dynamic change of the capabilities of the data inside.

4.3 Model of Scott

Scott [15] builds a security model of the world by adding spatial relationship between the elements in the ambient calculus. The security model is based on a building block called an *entity*. An entity is a spatial location. Every entity belongs to only one of six defined *sorts*. To distinguish physical entities from digital entities, Scott defines a *context*, a physical/virtual machine capable of running code. Scott's model uses capabilities from ambient calculus (*in/out*) and renames the capabilities depending on which entity uses the capability.

If the entity is a person moving between rooms, the actions are *walk in/walk out*. If the entity is a person interacting with a laptop, the capability is *pick up/put down*. If the entity is an agent moving between contexts, the capability is *emit/receive*.

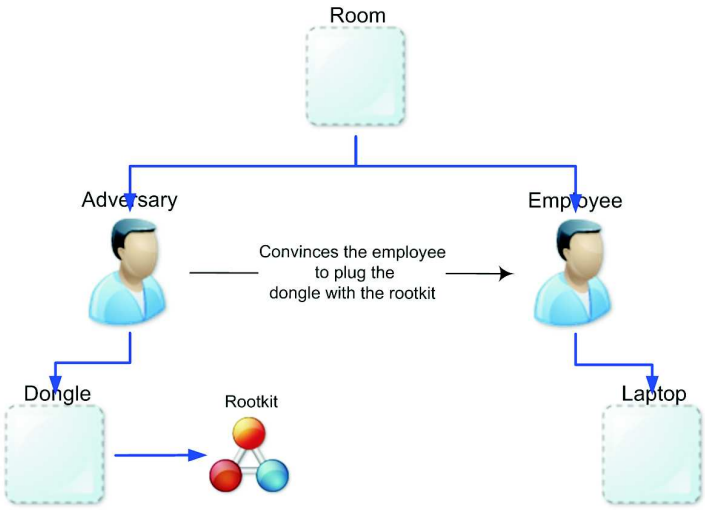


Figure 3: Road apple using entities

The model focuses on presenting the location of the entity, and the defined properties define the ability of one entity being contained in another which is presented in the sorts. To present tamper resistance of an entity, we can add multiple layers of protection to the data by inserting additional entities. But the definition of the emit/receive command teleports an entity from source address to destination address without taking in account the layers in between, making the model oblivious to the tamper resistance imposed by the device.

There is no social factor in the model of Scott. There is a sort `person`, but the meaning is spatial. The only capability this entity has is to pick up or leave a mobile entity. Through this we could present the coldboot attack, where the person physically changes the location of the RAM as well as the first version of the road apple attack. But the model fails to present the direct interaction between the adversary and the employee in the second version of the road apple, where the adversary directly interacts with the employee and convinces the employee to insert the dongle. Thus, the model can not fully present the road apple attack.

4.4 Model of Dragovic

Dragovic [16] presents a security model of the world by expanding Scott's model and focuses on exposure treats. The main building blocks are *data object*, which presents a collection of data with equal sensitivity as determined by a security

policy and *container*, which is an ambient (digital or physical) containing a data object or a lower level container. In the model of Dragovic, the container has as a boundary that protects the container or data object inside from the outside influences with variable degree of success. Every container propagates downwards its own influences in addition to the influences the container inherits from the parent container. Boundary transparency is defined based on the degree of protection the parent container offers to the child container. Dragovic uses *class* (similar to Scott's sort) to group elements. Another distinction is made by adding a *type* to the container, which presents the behavior of the container when exposed to an influence from the environment. Mobility of the data is presented by four operations: *enter*, *leave*, *migrate*, which atomically binds the previous two operators and *state_update*, which is used to update the status of the attributes of a container. The model presented by Dragovic [27, 16] besides considering the spatial/temporal characteristics of the object, considers the object's physical properties, such as the object's capability to resist influences from the surrounding environment, making the model excellent for presenting the tamper resistance of a device.

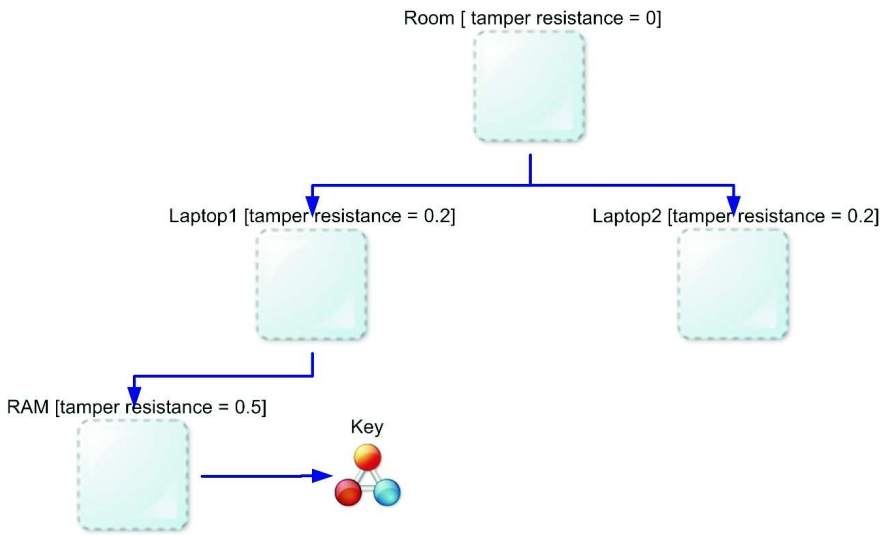


Figure 4: The coldboot attack using containers

The model of Dragovic includes Scott's model with the addition of the physical property of the objects, as well as the definition of sensitivity of data, allowing us to model tampering with a device and the coldboot attack to a level where all elements are realistically presented. When modeling the coldboot attack, we define the RAM as a container and the encryption key as a data object. The accessibility of the RAM is defined by the RAM's transparency in addition of the laptop's transparency. Before the coldboot attack, we consider the RAM

as a container with limited tamper resistance. After the RAM is removed from the laptop, the tamper resistance of the RAM increases due to the degradation of the data. Thus, we can successfully present the coldboot attack to a satisfactory level.

Dragovic does not define an object *person*, therefore there is no defined interaction between a person and a container. By presenting the employee and the adversary as containers, we are able to present the movement of the dongle with the rootkit from the adversary to the employee's laptop. Yet, we are not able to present the interaction between the adversary and the employee, where the employee is convinced to insert the dongle. Thus, we cannot model the road apple attack with direct interaction.

4.5 Comparison of the models

This section compares the analyzed modeling approaches. Table 3 presents the objects and properties of the objects we are interested in the analyzed models.

Element		Ambient calculus	TAM	Scott	Dragovic
Data	static	yes	yes	yes	yes
	dynamic	yes	yes	yes	yes
Physical	spatial/temp.	yes	no	yes	yes
	resistance	no	no	no	yes
Role	transitions	no	no	no	no
	interactions	yes	no	no	no

Table 3: Ability of the models to present digital/physical/social elements

From the presented results, we make the following observations. Ambient calculus is formal and capable of presenting most of the properties of interest. The other models impose restrictions on the model enabling them to focus on a specific area of interest, making the models less general than ambient calculus and thus preventing the models to represent some of the properties of interest. These models are easier to work with compared with ambient calculus. TAM is incapable of presenting any physical or social properties, because the model focuses on software representation and does not contain states. Scott and Dragovic can not present role transition and role interaction because they do not include any social element in the model.

Table 4 provides an overview of the model's ability to present tampering with a physical device, the coldboot attack, as well as the road apple attack with indirect(road apple 1) and direct(road apple 2) interaction between the adversary and the user.

Tampering with a device can be presented with the model of Dragovic because the model can contain information about the property of a device. TAM

Name of attack	Ambient calculus	TAM	Scott	Dragovic
Tampering	no	no	no	yes
Coldboot	yes	partially	yes	yes
Road apple 1	yes	no	yes	yes
Road apple 2	yes	no	no	no

Table 4: Ability of the models to present the case study attacks

does not have this capability, and thus is not able to present the tampering. The model of Scott can use multiple layers to represent resistance, but the teleporting ability of data makes any attempt to represent resistance obsolete. The operators in ambient calculus do not support teleporting, enabling the presentation of the tamper resistance through multiple layers. Yet, the capabilities of the data can not change dynamically based on the change of the layer structure, preventing the complete presentation of tampering with data.

All of the analyzed models can present the coldboot attack with various degree of success except the TAM.

We are able to present the spatial movement of the dongle from the adversary to the employees laptop, but are not able to present the social interaction between the adversary and the user, where the adversary convinces the user to plug the dongle. This is the reason why Scott and Dragovic can only partially model the road apple with direct interaction.

5 Conclusion

We analyze the capability of the state of the art security models to present the treats arising from mobility of data in dynamic organizations. We show that none of the state-of-the-art security models simultaneously consider the data mobility and organizational dynamics to a satisfactory extent. Software modeling tools, like Microsoft's TAM, consider the physical infrastructure and roles to be static and this makes it hard to present dynamic changes in the system. Security models for ubiquitous computing are state based, but focus on spatial/temporal characteristics and fail to recognize social interactions, which are vital for social engineering threats. As a result, we conclude that none of the state-of-the-art security models effectively identifies the potential security threats caused by data mobility in a dynamic organization.

The information omnipresence and dynamic organizations shift the stress from mainly digital attacks to a combination of digital, physical and social attacks. To cope with the threats, the paper presents the requirements for an integrated state base model. The goal of the proposed integrated model is to model the world from all three aspects, digital, physical and social and realistically present the possible attacks. The paper identifies the objects of interest

from all three aspects and presents an initial classification of the properties affecting the security on the identified objects.

Future work is to define a formal security model that satisfies the requirements provided here and to define the interactions between the identified objects, based on the properties of the objects. An interesting direction is to use the properties of the objects presented in the model of Dragovic and extend it with roles to cover the social aspect of the world.

Acknowledgements

We thank Roel Wieringa, Siv Hilde, and André van Cleeff for their help with the paper.

This research is supported by the Sentinels program of the Technology Foundation STW, applied science division of NWO and the technology programme of the Ministry of Economic Affairs under project number TIT.7628

References

Papers

[1] K.D. Mitnick and W.L. Simon. *The Art of Deception: Controlling the Human Element of Security*. Wiley, 2002.

[2] D. Lacey. Inventing the future–the vision of the Jericho forum. *Information Security Technical Report*, 10:186–188, 2005.

[3] G. Palmer. De-perimeterisation: Benefits and limitations. *Information Security Technical Report*, 10:189–203, 2005.

[4] J. Walker. The extended security perimeter. *Information Security Technical Report*, 10:220–227, 2005.

[5] C. Laird. Taking a hard-line approach to encryption. *Computer*, 40:13–15, 2007.

[6] E.W. Felten. Understanding trusted computing: will its benefits outweigh its drawbacks? *Security & Privacy Magazine, IEEE*, 1(3):60–62, 2003.

[7] R. Wieringa. Conceptual modeling in social and physical contexts. 2008.

[8] R. Hasan, S. Myagmar, A. J. Lee, and W. Yurcik. Toward a threat model for storage systems. In *1st ACM Workshop on Storage Security and Survivability (StorageSS)*, pages pages 94–102, Fairfax, Virginia, Nov 2005. ACM Press.

[9] L. Chen, D. Feng, and L. Ming. The security threats and corresponding measures to distributed storage systems. In *Lecture notes in computer science*, volume 4847, pages 551–559. Springer, 2007.

[10] J.A. Halderman, S.D. Schoen, N. Heninger, W. Clarkson, W. Paul, J.A. Calandrino, A.J. Feldman, J. Appelbaum, and E.W. Felten. Lest we remember: Cold boot attacks on encryption keys. *USENIX Security*, pages 45–60, 2008.

[11] P. Gutmann. Secure deletion of data from magnetic and solid-state memory. In *SSYM'96: Proceedings of the 6th conference on USENIX Security Symposium, Focusing on Applications of Cryptography*, pages 8–8, Berkeley, CA, USA, 1996. USENIX Association.

[12] M. AlZarouni. The reality of risks from consented use of usb devices. In C. Valli and A. Woodward, editors, *Proceedings of the 4th Australian Information Security Conference*, 2006.

[13] F. B. Schneider. Enforceable security policies. *ACM Trans. on Information and System Security*, 3(1):30–50, Feb 2000.

[14] J. A. Goguen and J. Meseguer. Security policies and security models. In *3rd Symp. on Security and Privacy (S&P)*, pages 11–20, Oakland, California, Apr 1982. IEEE Computer Society.

[15] D.J. Scott. *Abstracting Application-Level Security Policy for Ubiquitous Computing*. PhD thesis, University of Cambridge, 2004.

[16] B. Dragovic and J. Crowcroft. Containment: from context awareness to contextual effects awareness. *2nd Intl Workshop on Software Aspects of Context (IWSAC'05)*, 2005.

[17] L. Cardelli and A.D. Gordon. Mobile ambients. *Theoretical Computer Science*, 240(1):177–213, 2000.

[18] F. Swiderski and W. Snyder. *Threat Modeling*. Microsoft Press Redmond, WA, USA, 2004.

[19] D. Xu and K.E. Nygard. Threat-driven modeling and verification of secure software using aspect-oriented petri nets. *IEEE Transactions on Software Engineering*, 32(4):265–278, 2006.

[20] R. Chinchani, A. Iyer, H. Ngo, and S. Upadhyaya. A target-centric formal model for insider threat and more. *University at Buffalo*, 2004.

[21] E. Oladimeji. Security threat modeling and analysis: A goal-oriented approach. *Proceedings of the 10 th IASTED International Conference on Software Engineering and Applications*, 2006.

[22] J. Pauli and D. Xu. Threat-driven architectural design of secure information systems. *Proc. of the 7th International Conference on Enterprise Information Systems*, pages 136–143, 2005.

[23] Y. Chen, B. Boehm, and L. Sheppard. Value driven security threat modeling based on attack path analysis. In *Hawaii International Conference on System Sciences*, volume 00, page 280a, Los Alamitos, CA, USA, 2007. IEEE Computer Society.

[24] L. Cardelli, G. Ghelli, and A.D. Gordon. Types for the ambient calculus. *Inf. Comput.*, 177(2):160–194, 2002.

[25] M. Bugliesi, G. Castagna, and S. Crafa. Access control for mobile agents: The calculus of boxed ambients. *ACM Trans. Program. Lang. Syst.*, 26(1):57–124, 2004.

[26] C. Braghin, A. Cortesi, R. Focardi, and S. Bakel. Boundary inference for enforcing security policies in mobile ambients. In *TCS '02: Proceedings of the IFIP 17th World Computer Congresse*, pages 383–395, Deventer, The Netherlands, 2002. Kluwer, B.V.

[27] B. Dragovic and J. Crowcroft. Information exposure control through data manipulation for ubiquitous computing. In *NSPW '04: Proceedings of the 2004 workshop on New security paradigms*, pages 57–64, New York, NY, USA, 2004. ACM.

Web references

[28] J. Ryder. Laptop security, part one: Preveting laptop theft. *SecurityFocus*, July 2001.

[29] J. Ryder. Laptop security, part two: Preventing information loss. *SecurityFocus*, August 2001.

[30] B. Rudis. Protecting road warriors: Managing security for mobile users , part one. *SecurityFocus*, April 2004.

[31] B. Rudis. Protecting road warriors: Managing security for mobile users , part two. *SecurityFocus*, May 2004.

[32] Iron Key team. Benefits of secure usb flash drives, June 2007. www.ironkey.com.

[33] A. Shah. Analysis of rootkits: Attack approaches and detection mechanisms. 2006.

[34] J. Heasman. Implementing and detecting a pci rootkit. Black Hat Europe, 2006.

[35] J. Heasman. Implementing and detecting an acpi rootkit. Black Hat Federal, 2006.

[36] J. Butler and S. Sparks. Windows rootkits in 2005, part two. *SecurityFocus*, November 2005.

[37] S. Stasiukonis. Social engineering the usb way. Technical report, Secure Network Technologies Inc, 2006.

[38] Microsoft. Microsoft threat analysis and modeling v2.1.2, 2007.

[39] P. Saitta, B. Larcom, and M. Eddington. Trike v. 1 methodology document [draft]. http://www.trikemethod.net/, 2005.

Appendix A

In the following appendixes we model the device tampering, coldboot attack and road apple, which were presented in the case study. We analyze in modeling techniques presented in the paper in greater detail and discuss the shortcoming of the presented models.

Ambient calculus

For the semantics of the ambient calculus refer to the original paper [21]. The results below have been validated on the Basic Ambient Factory tool.

Tampering:

1. Without any information about the tamper resistance of the device, we can present the data leaving the device as.

```
device[data[out device]]
```

2. We can present the increased tamper resistance by using N layers of nested ambients.

```
device[pLayer1…[pLayerN[data[out pLayerN… out pLayer1.out device]]]]
```

The capability of leaving the device resides with the data, not with the device. Through this presentation, for every change on the stack of layers, the capability of the data needs to be changed dynamically.

A simple example is:

```
device[pLayer1[pLayer2 [data[out pLayer2.out pLayer1.out device]]]]
```

If `pLayer1` is removed from the device (the adversary circumvents one layer of defense or reduces the strength of the resistance of the device), the data ambient will never be able to get out of the device because `out pLayer1` capability will never be executed.
(NO)

Coldboot attack:

To present the attack, we model 2 laptops, the data of interest, the RAM which is being moved, and the destination hard drive. In steps (1) and (2), the ram moves from laptop1 to laptop2. In steps (3) and (4) the data moves from the ram to the hdd of laptop2. Every current action is presented in bold font.

laptop1[ram[**out laptop1**. in laptop2. data[out ram.in hdd]]] | laptop2[hdd[]] (1)
→laptop1[] | laptop2[hdd[]] | ram[**in laptop2**. data[out ram.in hdd]] (2)
→laptop1[] | laptop2[ram[data[**out ram**.in hdd]] | hdd[]] (3)
→laptop1[] | laptop2[ram[hdd[] | data[**in hdd**]]] (4)
→laptop1[] | laptop2[ram[hdd[data[]]]] (5)

(YES)

Road apple:

adversary[in cafeteria.usb[out adversary. in employee. in laptop. rootkit[]] | out cafeteria] | employee[in cafeteria | laptop[open usb] | out cafeteria] | cafeteria[]

Trace when the adversary succeeds in the attack:

adversary[**in cafeteria**.usb[out adversary. in employee. in laptop. rootkit[]] | out cafeteria] | employee[in cafeteria | laptop[open usb] | out cafeteria] |cafeteria[] (1)
→ employee[in cafeteria | laptop[open usb] | out cafeteria] | cafeteria[adversary[usb[**out adversary**. in employee. in laptop. rootkit[]] | out cafeteria]] (2)
→ employee[in cafeteria | laptop[open usb] | out cafeteria] | cafeteria[adversary**[out cafeteria**] | usb[in employee. in laptop. rootkit[]]] (3)
→ employee[**in cafeteria** | laptop[open usb] | out cafeteria] | cafeteria[usb[in employee. in laptop. rootkit[]]] | adversary[] (4)
→ cafeteria[employee[laptop[open usb] | out cafeteria] | usb[**in employee**. in laptop. rootkit[]]] | adversary[] (5)
→ cafeteria[employee[usb[**in laptop**. rootkit[]] | laptop[open usb] | out cafeteria]] | adversary[] (6)
→ cafeteria[employee[laptop[**open usb** | usb[rootkit[]]] | out cafeteria]] | adversary[] (7)
→ cafeteria[employee[laptop[rootkit[]] | **out cafeteria**]] | adversary[] (8)
→ cafeteria[] | employee[laptop[rootkit[]]] |adversary[] (9)

Besides being able to present the coldboot attack, the calculus is able to present and concurrent actions. In the above example there are two branching possibilities. The first branching can occur in (2), where the adversary enters the cafeteria but does not leave the dongle (out cafeteria capability in adversary executes before out adversary in usb). The second branching can occur in (5), where the employee enters the cafeteria but does not pick up the dongle (out cafeteria capability in employee executes before in employee in usb).
(YES)

Road apple with direct interaction between the adversary and the employee:

In this scenario, the adversary sends a message to the employee to plug in the dongle. After approval, the dongle goes from the adversary to the laptop of the user and infects the laptop.
One trace is:

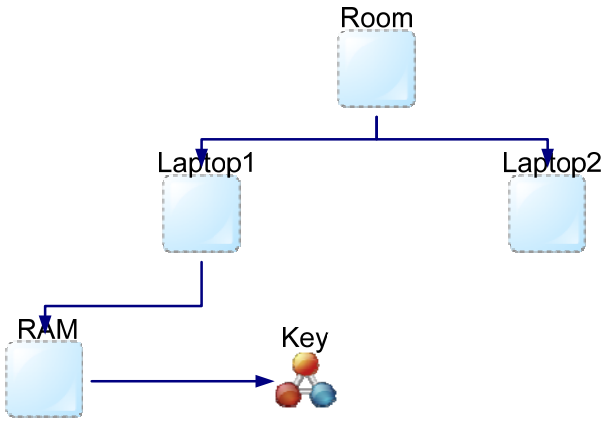
adversary[m1[**out adversary**.in employee] |open m2.usb[out adversary. in employee.in laptop.rootkit[]]] | employee[open m1.m2[out employee.in adversary] | laptop[open usb]] (1)
→ adversary[open m2.usb[out adversary. in employee.in laptop.rootkit[]]] | employee[open m1.m2[out employee.in adversary]| laptop[open usb]] | m1[**in employee**]
→ adversary[open m2.usb[out adversary. in employee.in laptop.rootkit[]]] | employee[m1[] | **open m1**.m2[out employee.in adversary] | laptop[open usb]] (2)
→ adversary[open m2.usb[out adversary. in employee.in laptop.rootkit[]]] | employee[m2[**out employee**.in adversary] | laptop[open usb]] (3)
→ adversary[open m2.usb[out adversary. in employee.in laptop.rootkit[]]] | employee[laptop[open usb]] | m2[**in adversary**] (4)
→ adversary[m2[] | **open m2**.usb[out adversary. in employee.in laptop.rootkit[]]] | employee[laptop[open usb]] (5)
→ adversary[usb[**out adversary**. in employee.in laptop.rootkit[]]] | employee[laptop[open usb]] (6)
→ adversary[] | employee[laptop[open usb]] | usb[**in employee**.in laptop.rootkit[]] (7)
→ adversary[] | employee[usb[**in laptop**.rootkit[]] | laptop[open usb]] (8)
→ adversary[] | employee[laptop[**open usb** | usb[rootkit[]]]] (9)
→ adversary[] | employee[laptop[rootkit[]]] (10)
(11)

(YES)

Appendix B
Model of Scott

Tampering:
Similarly to the ambient calculus, we can assume multiple layers of protection over the data. But, the defined command over data (*emit/receive)* uses teleporting and ignores the layers between the source path and destination path. Because of the teleporting, any obstacles placed between the data and the adversary can be ignored, and thus, no tamper resistance of a device can be presented.
(NO)

Coldboot attack:



Initial:

room[laptop1[RAM[key[0]]] | laptop2[0]]

Transformations:

$$\frac{\dfrac{laptop1 \xrightarrow{RAM} room; room \xrightarrow{RAM} laptop2}{laptop1 \xrightarrow{RAM} laptop2}}{\eta[laptop1] \xrightarrow{RAM} \eta[laptop2]}$$

Result:

room[laptop1[0] | laptop2[RAM[key[0]]]]

The transformation above shows how the RAM leaves laptop1 and goes to laptop2, through the room entity.

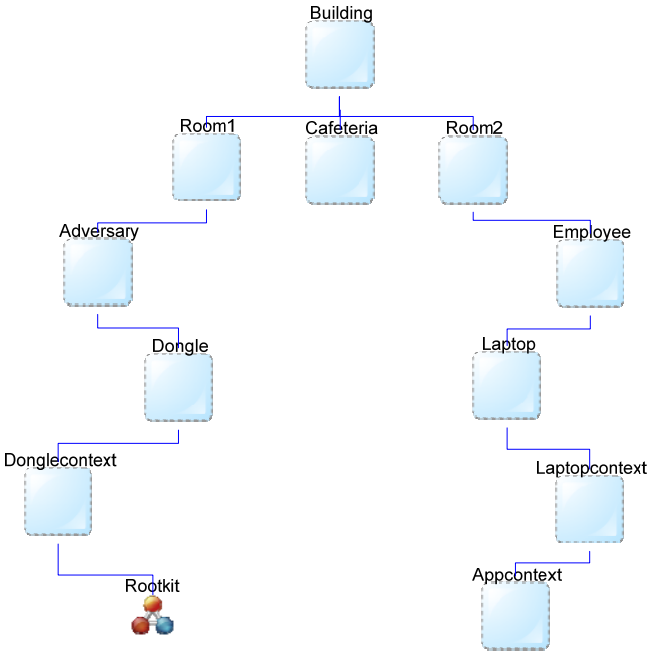
$$laptop1 \xrightarrow{RAM} room; room \xrightarrow{RAM} laptop2$$

This derivation can be read as: the RAM moves from the position of laptop1 to the position of laptop2. As previously stated, here we see the teleportation effect, since all entities in between (in this case room) are ignored in the last part of the formula.

$$\eta[laptop1] \xrightarrow{RAM} \eta[laptop2]$$

(YES)

Road apple:



Initial:

building[room1[adversary[dongle[donglecontext[rootkit[0]]]]] | cafeteria[0] | room2[employee[laptop[laptopcontext[appcontext[0]]]]]]

Transformations:

$$\frac{room1 \xrightarrow{adversary} building;building \xrightarrow{adversary} cafeteria}{\dfrac{room1 \xrightarrow{adversary} cafeteria}{\eta[room1] \xrightarrow{adversary} \eta[cafeteria]}}$$

$$\frac{adversary \xrightarrow{dongle} cafeteria}{\eta[adversary] \xrightarrow{adversary} \eta[cafeteria]}$$

$$\frac{cafeteria \xrightarrow{adversary} building;building \xrightarrow{adversary} room1}{\dfrac{cafeteria \xrightarrow{adversary} room1}{\eta[cafeteria] \xrightarrow{adversary} \eta[room1]}}$$

$$\frac{room2 \xrightarrow{employee} building;building \xrightarrow{employee} cafeteria}{\dfrac{room2 \xrightarrow{employee} cafeteria}{\eta[room2] \xrightarrow{employee} \eta[cafeteria]}}$$

$$\frac{cafeteria \xrightarrow{dongle} building;building \xrightarrow{dongle} laptop}{\dfrac{cafeteria \xrightarrow{dongle} laptop}{\eta[cafeteria] \xrightarrow{dongle} \eta[laptop]}}$$

$$\frac{donglecontext \xrightarrow{emit(rootkit)} laptopcontext;laptopcontext \xrightarrow{recieve(rootkit)} appcontext}{\dfrac{donglecontext \xrightarrow{rootkit} appcontext}{\eta[donglecontext] \xrightarrow{rootkit} \eta[appcontext]}}$$

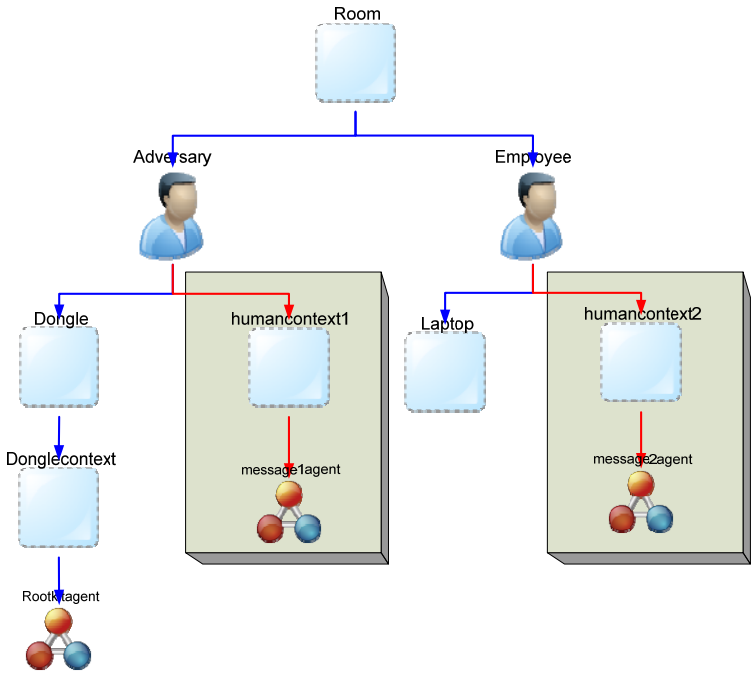
Result:
building[room1[adversary[0]] | cafeteria[0] |
room2[employee[laptop[dongle[dongle context[0]] |
laptopcontext[appcontext[rootkit[0]]]]]]]

(YES)

Road apple with direct interaction between the adversary and the employee:

Although the model can present the physical/digital transitions, the model can not present the interaction employee-adversary. A similar approach as in ambient calculus will require generating new rules besides *pick up/leave down*, generalizing the model up to ambient calculus level.

First, we will need to add context to people. Second, we can use special agents which will present messages and reside in the context in people. The world for the road apple attack with direct interaction between the employee and the adversary would look like this:



room[adversary[usb[usbcontext[rootkit[0]]] |humancontext1[message1agent[0]] |
employee[laptop[laptopcontext[0]] | humancontext2[message2agent[0]]]

Next, we need to add rules for exchanging messages between people, such as:

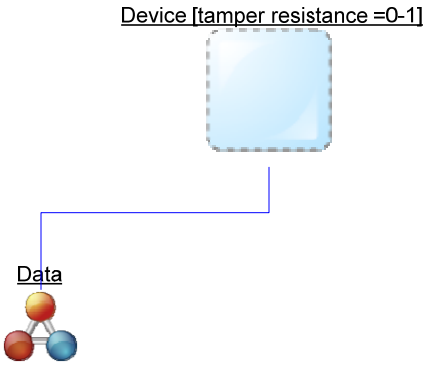
$$\frac{humancontext1 \xrightarrow{emit(message1agent)} employee; employee \xrightarrow{recieve(message1agent)} humancontext2}{humancontext1 \xrightarrow{message1agent} humancontext2}$$

$$\frac{humancontext1 \xrightarrow{message1agent} humancontext2}{\eta[humanlcontext] \xrightarrow{message1agent} \eta[humancontext2]}$$

(NO)

Appendix C
Model of Dragovic

Tampering:

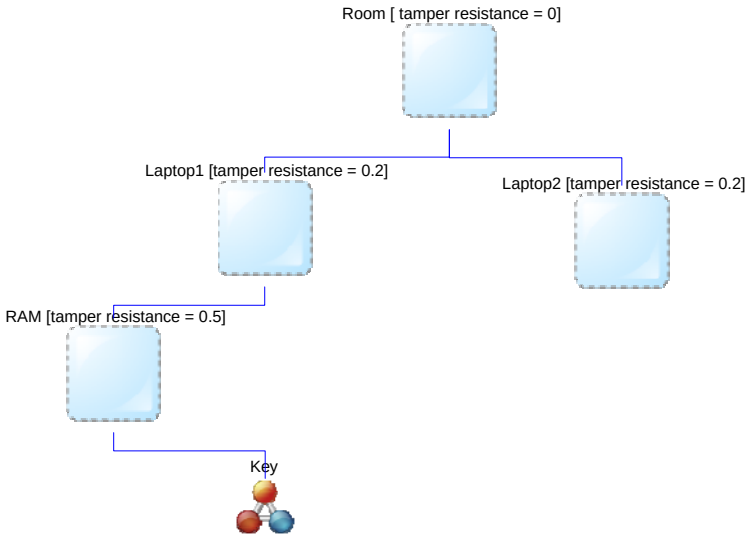


leave(Data)

The model of Dragovic, besides providing attributes that define the tamper resistance of a device, provides information for the sensitivity of the data inside the device, the level of exposure of this data, as well as additive protection of multiple types of containers. The additional information is not provided because it is out of the scope of the paper. For further information refer to [20,32].

(YES)

Coldboot attack:



Initial:
tr = tamper resistance
room:tr=0;[laptop1:tr=0.2;[RAM:tr=0.5;[key:tr=0[0]]] | laptop2:tr=0.2;[0]]

Transformations:

leave(Room/Laptop1/RAM)

update(RAM, tamper resistance = 0.8)
enter(RAM, Room/Laptop2)
update(RAM, tamper resistance = 0.2)

Result:
room:tr=0;[laptop1:tr=0.2; | laptop2:tr=0.2;[RAM:tr=0.5;[key:tr=0;[0]]]]

Besides providing information about the movement of the RAM with the data, the model provides information for the degradation of the data inside the RAM when the RAM is removed from the laptop.
(YES)

Road apple:
Initial:
For brevity, we assume the containers to have no attributes.

building [room1[adversary[dongle [rootkit[0]]]] | cafeteria[0] |
room2[employee[laptop[0]]]]

Transformations:
migrate (building/room1/adversary, building/cafeteria)
migrate (building/cafeteria/adversary/dongle, building/cafeteria)
migrate (building/cafeteria/adversary, building/room1)
migrate (building/room2/employee, building/cafeteria)
migrate (building/cafeteria/dongle, building/cafeteria/employee/laptop)
migrate (building/cafeteria/employee/laptop/dongle/rootkit,
building/cafeteria/employee/laptop)

Result:
building[room1[adversary[0]] | cafeteria[0] | room2[employee[laptop[dongle
[rootkit[0]]]]]
(YES)

Road apple with direct interaction between the adversary and the employee:

Initial:
room[adversary[dongle[rootkit[0]]] | employee[laptop[0]]]

Transformations:
migrate (Room/Adversary/Dongle, Room/Employee/Laptop)
migrate (Room/Employee/Laptop/Dongle/Rootkit, Room/Employee/Laptop)

Result:
room[adversary[0] | employee[laptop[dongle[0] |rootkit[0]]]]

This model also does not present interaction between people. A workaround will be similar as in modeling the road apple example with Scott's model.
(NO)