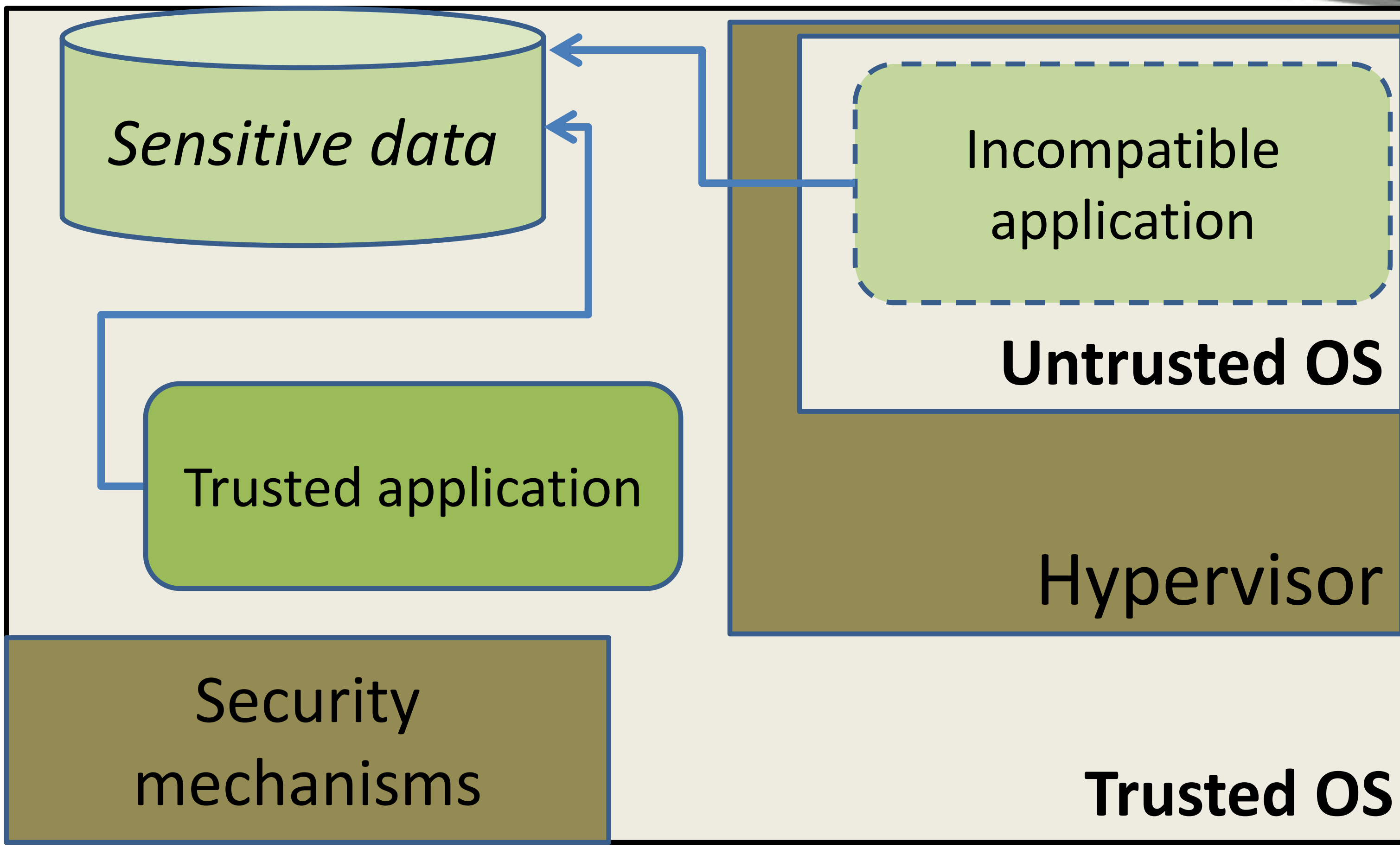
VIRTUAL ENVIRONMENT SECURITY MODELING

Dmitry Zegzhda, Ekaterina Rudina
Saint-Petersburg State Polytechnic University

Our goal is the well-grounded use of hybrid systems



➢Trusted OS design concepts

Background

Peter D. Zegzhda, Dmitry P. Zegzhda

MMM-ACNS 2001

Secure system design based on consistent and correct implementation of information flows and flow control

Dmitry P. Zegzhda, Pavel G. Stepanov, Alexey D. Otavin.

MMM-ACNS 2001

Principles, models and the formally-proven architecture of secure operating system

➢ Formal definition of security policy for real operating system and resolution about security

Background

Peter D. Zegzhda, et al.

MMM-ACNS 2003

The approach for testing security policies enforcement and weakness

Dmitry P. Zegzhda, Maxim O. Kalinin.

EIWST-07

A logical processor for verification of operating systems security

➢ Formal methods of the vulnerabilities detection

Background

Peter D. Zegzhda, et al.

MMM-ACNS 2005

Approach to discover vulnerabilities of the operating systems by logical processor

Peter D. Zegzhda, et al.

MMM-ACNS 2005

The generalization of the formal verification procedure

➢ Application behaviour and assurance evaluation techniques

Background

Dmitry P. Zegzhda, Maxim O. Kalinin.

EIWST-07

Verifying security assumption for the evaluation of solution assurance

Peter D. Zegzhda et al.

MMM-ACNS 2007

The roadmap for the security evaluation based on security attributes analysis

Research purpose

Use trusted OS with untrusted applications without loss of secure properties

Dmitry P. Zegzhda, Alex M. Vovk.

Secure Hybrid Operating System

“Linux over OSMOS”

presented at **MMM-ACNS 2005**

Was proposed the design of trusted systems based on the hybrid OS technology that is similar to virtualization technology

Was proven the adequacy of models of the system states for the problem of modeling hybrid systems

Virtualization in computer security: some examples (1/2)

Chen P.M., Noble B.D.
8th Workshop on HTOS 2001

Is asserted that some applications to make them trusted should relocate into a virtual environment, but the formal substantiation of that approach is absent.

Garfinkel T., Rosenblum M.
NDSSS 2003

Is presented an architecture that retains the visibility of a host-based IDS, but pulls the IDS outside of the host for greater attack resistance.

Background

Virtualization in computer security: some examples (2/2)

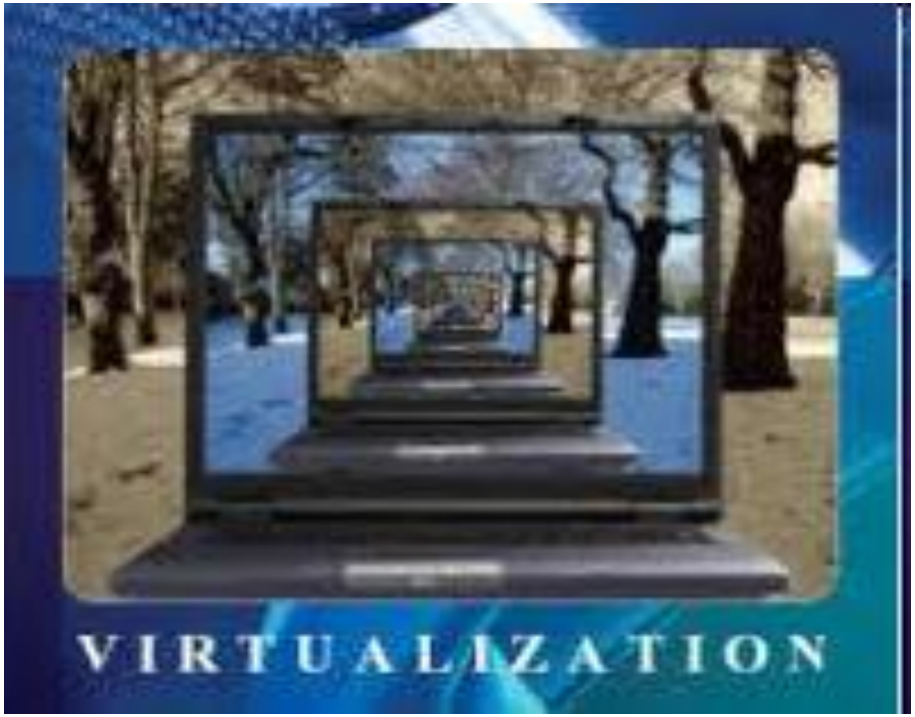
Background

Liang Zhenkai, et al.
ACSAC 2003
The technique analogous to the virtualization is used to isolate the effects of untrusted program execution from the rest of the system.

Goldberg I., et al.
6th Usenix Security Symposium
The approach to program isolation is offered. The declared advantage is to reduce the risk of a security breach by restricting the program's access to the operating system.

Using virtualization allows to get a new solutions in computer security scope

BUT



is NOT only the isolation

We want to use other virtualization properties to secure information processing

What conditions?

Let’s formulate the formal conditions saving the necessary properties of the data processing

Hypervisor properties [Popek, Goldberg 1974]

- Equivalence

of the virtual environment and the non-virtualized system

- Full control of resources

by the hypervisor (including resources allocation and reallocation hypervisor initiated)

- Efficiency

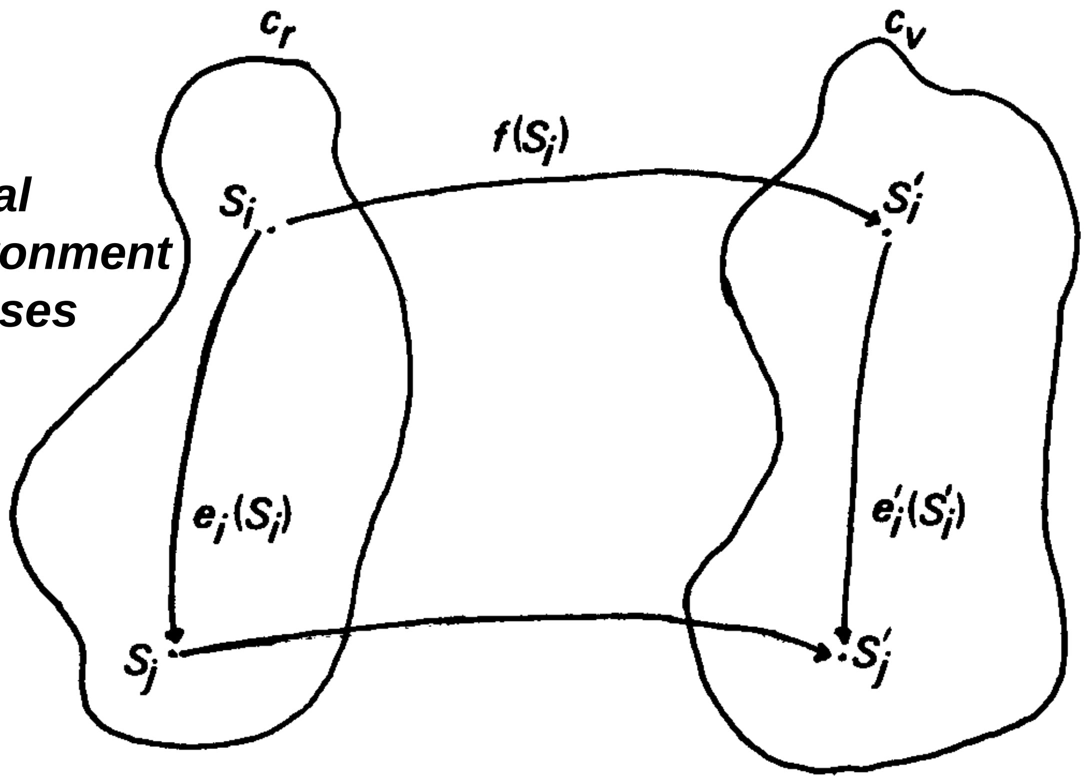
of the data processing in the virtual environment

Equivalence property

$f\colon C_r \rightarrow C_v,\ e_i \in I$

for each $S_i \in C$ *and* e_i *exists* e_i'*:* $f(e_i(S_i)) = e_i'(f(S_i))$

Virtual environment statuses



Non-virtualized system statuses

Resource control property

- it is not possible for a program running under it in the virtual environment to access any resource not explicitly allocated to it

- it is possible under certain circumstances for the hypervisor to regain control of resources already allocated.

What about model?

We need the model

- Adequate to modern complex computing systems

- powerful to express the mentioned properties (considering all modern virtualization techniques)

The model of the hybrid system (1/2)

$$\boldsymbol{M}=(P, R, TR, D, \tau, \delta, F, Prg, \varphi)$$

The key feature of the model is

resources typification

that makes the model powerful and expressive

Resource typification is the idea used in **SPM, ESPM, TAM** and some other models allowing to make some problems resolvable

(*see papers of R.Sandhu about these models*)

The model of the hybrid system (2/2)

$S=(P,R,p)$ state of the system

$C=\{S\}$ set of the possible states

$F=\{fi\}i \in 1:n$ transition functions set

F^* set of the sequences of functions

$Prg \subset F^*$ set of programs

Virtualization modeling

Non-virtualized system model

$$\boldsymbol{M}^{\boldsymbol{A}}=(P^A, R^A, TR^A, D, \tau^A, \delta, F, Prg^A)$$

Goal system model

$$\boldsymbol{M}^{\boldsymbol{V}}=(P^V, R^V, TR^V, D, \tau^V, \delta, F, Prg^V)$$

so as $R^A \subseteq R^V$

and *D* representation is the same in both models

Generalized security property VER

$$\forall r \in R(\tau(r) \in CR \Rightarrow VER(\delta(r)))$$

$CR \subseteq R$

types of sensitive resources

$VER: D \rightarrow \{true, false\}$

predicate describing security property

Assumptions

- Identical representation of data *D* for each model
- Sensitive resources should be virtualized $CR^A \subseteq VR$
- Hypervisor behavior answers to security condition

Theorem

If the resource typification function is mapped from M^A to M^V homomorphically

$$\exists \chi : TR^A \rightarrow TR^V, \forall r \in R^A \subseteq R^V (\tau^V(r) = \chi(\tau^A(r)))$$

and subset of the sensitive resources of the initial system is appropriate to the subset of the sensitive resources of the virtual environment

$$\forall t \in TR^A : t \in CR^A \Leftrightarrow \chi(t) \in CR^H$$

then the secure execution of any program of the system A is provided.

It is mean that…

When the given conditions are met, any program’s behavior **will be changed** by the virtualization hypervisor and security mechanisms according to the security requirements

Remarks

- Declared conditions are sufficient

- These conditions can be satisfied easier if some best practices of computer security are provided

Conclusion

- Sufficient conditions of inheritance of the security properties by untrusted applications run in virtual environment were defined and proved
- These conditions can be used to build a formal proven trusted system handling sensitive data properly without verifying of untrusted applications

Perspectives

Further we have to

- clarify defined conditions for some special cases:
  - Thin hypervisor
  - Application virtualization
  - …
- use approach based on these conditions to design trusted systems
- create an technique of verifying systems safety

THANK YOU FOR YOUR ATTENTION!

Dmitry Zegzhda, Ekaterina Rudina
Saint-Petersburg State Polytechnical University