DESIGN PRINCIPLES OF TRUSTED SYSTEMS

Article · February 2014

Running Head: Design Principles of Trusted Systems

DESIGN PRINCIPLES OF TRUSTED SYSTEMS

In American Psychological Association's Style

Jiwan Pokharel

Loyola University of Chicago

Abstract

This paper is written as a part of PROJECT-1 for COMP 448, Spring 2014. The paper entails about some design principles involved in designing trusted systems. Also, it tends to explain the security concerns, mechanisms to enforce security in the system and various loopholes that prevail. Moreover, the paper is an analytical evaluation, but not an expert opinion.

1. INTRODUCTION

As opposed to the centralized architectures before 1970s, distributed architectures have rapidly evolved since 1990s ("Abrahams and Joyce"). With this evolving distributed computing environment, there arose a need for ***trust*** based scenario for easier and convenient access control, authorization, and information security. This paper intends to discuss this trusted system architecture and policies pertaining to their design and implementation, thereby providing a basis for the usefulness of trusted systems. Also, the paper shall, in brief, exploit the variation of these systems in relation to secured systems. In the latter sections of the paper, we shall also involve in detailed analysis of the policies that influence the architectural design. Moreover, the example of "*Reference Monitor"* will too be a major point of discussion in the paper.

1.1 What is a trusted system?

A system on which we rely to enforce the security policies and strategies is referred as a trusted system. In the field of computing, "**Trust**" is that entity, on the basis of which a user transfers the information through the communication channel ("Wikipedia.org").

Once a trusted system is breached, it leads to the compromise of security policies governing the whole system setup ("Purdue University", 2010). Thus, a trusted system is the central figure to implement an organization's security policies and provides assurance, trust and security. In quest of achieving the system security, it is seen that the system has to implement certain kind of layered architecture which renders eavesdropping ineffective.

When we define a trusted system, it is essential to know the difference between the trusted system and trustworthy system. A trusted system is the one whose failure breaks the entire security architecture, and also it is the centralized figure which is designated the word "*trusted"* due to its role, but a trustworthy system is the one which can be trusted, if at all, it is implemented correctly. Thus, a trustworthy system can move towards the prominence of being trusted, should it be implemented properly and should it satisfy all the security policies ("Purdue University", 2010).

1.2 Why Trusted System?

Advancements in technologies bring forth numerous challenges with it. In quest of developing time sharing scenario, it was essential to address the issue of trust. As a user could derive numerous services from different non-local applications or resources, there was a need to control, authenticate and monitor the access ("Searchsecurity.com"). Henceforth, a basis for developing trust in computing became the need. There are various necessities of trusted system in present day. Trusted systems are necessary to provide enhanced security for the continually time sharing machines that work in shared environment. With increasing number of users, there too has been rise in the number of notorious ones, thus trusted system shall help safeguarding the memory of the concerned users. Also, in the time when spyware are everywhere around, trusted

computing mechanism shall provide security to various input and output operations. With trusted systems, encryption has been the primary mechanism of communication and security. This feature ultimately aids in protecting the user from unauthorized penetration. Moreover, in the scenario of remotely held systems, trusted mechanism provides a framework of safety from unwanted intruders.

2. TRUSTED SYSTEM AND SECURE SYSTEM

When designing a trusted system, it is essential that the development team aptly knows the thin line of difference existing between the trusted system and secured system. We can certainly create trusted system architecture, but for it to be secured, it needs to qualify the two pronged test of rendering the result in the form of "Yes" or "No" ("UC Santa Cruz"). We can elaborate more on the differences between these two kinds of systems as depicted below.

Concept of secured system is dichotomous in nature i.e. a system can either be secured or not. But, trust does not associate with mere dichotomy. Trust rather is a degree, up to which we can rely. Also, a secured system is the product characteristic, whereas trusted system is perceived in relation to appropriate analysis of evidences ("Edwardbosworth"). Moreover, secured property is of absolute nature where the result is defined by simple *'Yes'* or *'No'*, and there prevails no consideration of the facts about how, where, when and who used the system. But trusted system concept is relative and viewed contextually ("Ola Flygt", "Linnaeus University").Least not to say, developing a secured system is actually a goal and being trusted or trustworthy to a certain degree is the characteristic of the system in context.

3. TOWARDS DESIGNING A TRUSTED SYSTEM

Whenever we are designing a system, it is essential to identify the needs, goals as well as available resources. A careful understanding of the motive behind developing a system can only allow producing the system compliant with the requirements ("Roger S. Pressman", 2010). Therefore, this paper identifies some of the design steps as illustrated in the figure below:

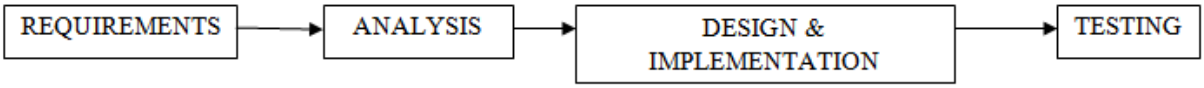

Figure 1. Steps in developing a system

3.1 Requirements

This is the first step in designing a trusted system. It is essential to understand the goal and the available resources when developing a system. When the developer knows the specification and pre-designated purpose of any system, developing it becomes an easier task. Thus, in requirements phase, various information required to carry out the development process are collected.

3.2 Analysis

Analysis phase is the other important step when developing the trusted system. This is the time when the developer analyzes or carefully reviews the customer's requirements and expectations from a particular trusted system. Also, at this point the available information of the project and the resource availability are taken into consideration. Moreover, a detailed plan to implement the required system is sketched here.

3.3 Design and Implementation

This very stage is the most scrutinized one. At this stage, the layout of the trusted system to be developed is made and appropriate components are modeled and designed. Also, the various components of the system are then coupled together to satisfy the requirements set up during the requirements phase. Eventually, a fully fledged system that is comparable to the designated goal is developed.

3.4 Testing

Once a project or a system is developed, there remains an integral part of the overall project development process known as testing. This highly essential stage in the development cycle is required to check the consistency of the components developed. Also, in this testing phase the unit level testing, component testing, integration testing and overall system testing is undertaken to verify the adherence of the system with the required functionalities and features.

Testing can be taken concurrently from the initial step of requirements collection too ("Roger S. Pressman", 2010). When, each smaller parts of the system are being tested one-after-other, it is known as unit testing. Similarly, if a particular component concerned with fulfilling a specific functionality is being checked for its consistency, it is known as component testing. Also, when the combination of two or more components to mutually satisfy functionality is being tested, it is referred to as integration testing. Eventually, when a full-fledged system, ready for deployment, is being tested for the overall functionality of the trusted system, it is known as system testing. Thus, the main goal of testing phase is to verify that the system being developed is consistent with the requirements and ready to be deployed in the real world scenario.

4. DESIGN CHARACTERISTICS OF TRUSTED SYSTEM

When designing any system it is necessary to keep customer's perspective and security policies in view. More than the design features of the ordinary system, a trusted system needs to incorporate some additional features in it. The major function of a trusted system is to provide security to the users and their resources. Thus, it is essential that every program and the user work with ***least privilege*** possible ("Ola Flygt", "Linnaeus University"). For example, if it suffices to work in a system using non-administrative privileges, we should better not invoke administrative capabilities. Doing so, might at times provide a vulnerable site for eavesdroppers to intrude into the system.

When designing a system it is essential to keep ***cost*** into great deal of consideration. As almost everything is governed by cost, lower cost and simpler design could mean a better environment for the development of a system viable in the real world scenario. For a trusted system, it is important that its mechanisms are easily testable. The notion of "security from secrecy" shall nowhere help to instill trust in the customers about the effectiveness of the system. Therefore, the more the system comes under scrutiny, the better it can be perceived by the customers. Trust shall then only be created, when the system can control each access requests. It should be able to restrict all unauthorized or illegal access requests. Another important aspect is ***privilege separation***. It is always better that privilege are separated ("UC Santa Cruz"). For instance, should there be any intrusion into the system; the intruder should not have easy access to all the system functionalities, should he have access to one. Moreover, attaching various services together makes it more complicated architecture as well as extremely vulnerable.

Also, a system should be ***easy to use*** by a customer. Should a user feel uneasy achieving his or her requirements from the system, there no longer remains the usefulness of the system nor there prevail a sense of protected-ness. To secure a system and make it remain un-breached, the first and foremost mechanism of access control is feature of ***no default access***. The mechanisms being used in the trusted system should vary as per the users ("Purdue University", 2010). The more common the mechanisms are amongst the users, the more probability of leakage of security policy. So, as far as possible, the mechanisms should be implemented per user.

5. SECURITY FEATURES IN DESIGNING TRUSTED SYSTEM

An ordinary system too considers the aspect of the security, but a trusted system has to think and implement beyond the ordinary system. It is evident that an ordinary system provides services like user authentication, memory protection, file access control, resources sharing fairness in service, etc ("Columbus State University"). Beyond these features, the trusted system should also implement the various security features.

Access control mechanism is highly essential with the objects of the trusted system. This prevents any unauthorized intrusion and secures the environment, thereby living up to the expectation of being trusted or trustworthy. Thus, mandatory as well as discretionary access control mechanisms should be implemented by the system. It too has to incorporate the technologies that address assurance and security. A trusted system should have its memory aptly separated from the users, data and library entities. Also, object reuse protection feature shall be implemented so as to prevent the leakage of the reuse of objects. To provide trusted computing, special paths of communication i.e. ***Trusted path*** needs to be enforced. This is essential for the

critical operations like user login and top secret communications like that in military purposes ("Edwardbosworth"). Moreover, it should have the security feature for detecting intrusion. For this, appropriate normal usage patens are to be created and the alarms or triggers are to be activated if any deviation from these pre-defined patterns is observed. Finally, the system should be capable to construct the audit log. This log should contain all the information regarding security related events and has to be tamper proof.

6. SECURITY POLICY MODELS

There are various security policy models prevalent till date, to govern the trusted system environment. We shall basically classify them as *Military Policy* and *Commercial Policy.* The military policy model is more stringent in terms of access control and security.

6.1 Military Policy

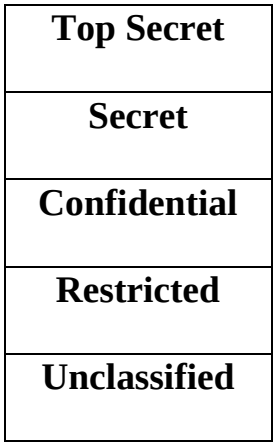

Figure. 3. Military Security model

In a military policy model, the information is ranked by its level of sensitivity. The sensitivity level of information is depicted in figure 3 in a top to bottom approach of decreasing level of sensitivity. Here, the information access is provided on a "*need-to-know"* basis ("Edwardbosworth"). Thus, access to them is done through the clearance procedure, which is

governed by dominance. The classification is shown by: **<rank, compartment>**. So, as per the dominance theory, any rake that has the clearance to the higher level of compartment can also access to the lower sensitivity level compartments ("Purdue University", 2010). For example, if a *Colonel* is allowed to access "**Top Secret"** compartment, certainly he can access lower sensitivity level compartments like "**Unclassified**"**.** Conversely, if *Sergeant* is permitted access only up to the compartment level "**Restricted",** he shall not have any access to higher level compartments like **"Top Secret**"**.**

6.2 Commercial Policy

This policy is employed in most of the other areas except military. This security model is more flexible than the military security model. Here the data items in a system are kept in different levels, but have different level of sensitivities like public, internal or proprietary ("Ola Flygt", "Linnaeus University"). This model does not use formalized notion of clearances and probably no dominance is employed in most of the cases.

7. REFERENCE MONITOR IN TRUSTED COMPUTING

A reference monitor is a non-by-passable, tamper-proof and analyzable component, which is primarily concerned with implementing trust in the system ("Trent Jaeger"). In other words, it is a media or key component responsible to create trust between the various components of the trusted system. It possesses the characteristics of ***complete mediation***, ***isolation*** and ***verifiability*** ("William Stallings"), ("Illinois.edu", 2011).

Complete mediation feature of the reference monitor implies enforcement of security rules on every access. Also, a reference monitor needs to be isolated from the unauthorized access, such that the database and the whole system are protected. Eventually, it is essential that appropriate evidence can be provided to prove the verifiability of the reference monitor. Thus, there should be possibility of mathematical demonstration to prove the isolation and complete mediation.

A reference monitor is a part of Trusted Computing Base (TCB) ("Illinois.edu", 2011). A TCB is the combination of hardware, procedural components and software to enforce security policies ("Purdue University", 2010).

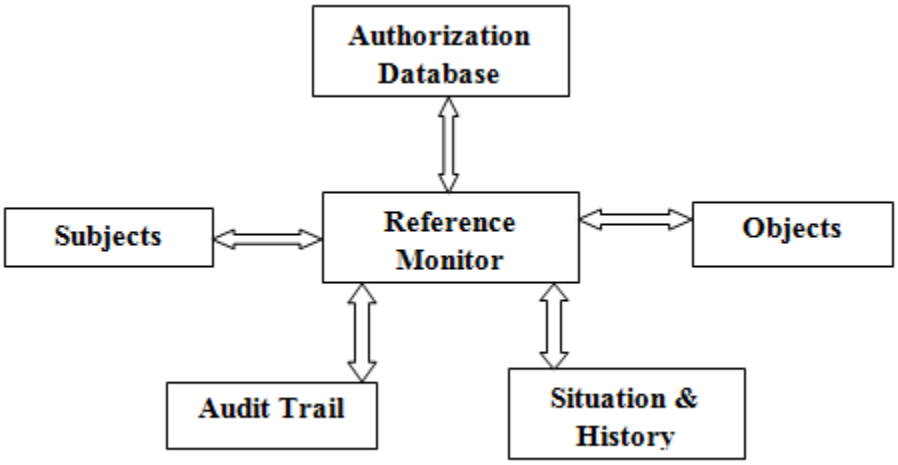

Figure 2. Reference Monitor in a system

Figure 2. represents the interconnection of a reference monitor with various components in a trusted system. It is the necessity of a reference monitor to mediate the access at every instance to the system. For the purpose, *authorization database* stores security attributes of the subjects and objects ("Iqware.com"). Here, *subjects* are the active entities and user processes that try to gain access to the resources through the system, and *objects*

are the resources like files, directories repositories and devices. Thus the reference monitor enforces the security policies among these and stores the record of all the activities through *audit trail*. Moreover, various deployment specific history, evaluation rules and real time data feeds are kept by the *situation & history* component ("Iqware.us")

8. CHALLENGES AND THREATS IN TRUSTED SYSTEM

The foremost challenge with trusted computing is creation of chain of trust. It is essential to select the appropriate ally who can be trusted to a given extent so as to perform the secured computing. Also, in this era of increasing technologies, internet can be a big challenge of trust. Copy righted materials that are transmitted through the trusted system, can at times be encroached; ultimately leading to the leakage of the copyrighted materials through the internet. Moreover, numerous business operations are done through the online mechanism involving the trusted system, so should there be any loop hole in the chain of trust; the whole financial activities go into jeopardy. For example: a customer performs his credit card transaction with a financial institution like bank, but should there be a compromise of trust among these two, there can be huge financial loss ("Jason et. Al.", 1998).

Spywares, dDoS attacks, bot-nets and many other hacking attacks are nemesis of trust. Should any of these be able to intrude into any system, there is a high risk that the whole system could be compromised. Similarly, there is a notion that trusted systems have often led to the copying of materials through internet. So, it is an issue to be addressed presently ("Jason et. Al.", 1998). Though communication and resource transfers are done in a secured channel using encryption privileges, there have already occurred instances

where a small slack in the design of the system has led to bigger rather negative impacts. Therefore, with technology and power there certainly come challenges, and they need to be duly taken care of.

9. CONCLUSION

Trusted system is the very basis of present day distributed real time computing. Designing a trusted system is to be taken a very good care of; else it can lead to hazards. As numerous organizations, financial institutions, military scenario and highly confidential works are done using trusted system, it is important to keep in view all the required security policies and implement them accordingly ("Searchsecurity.com").

Therefore, I believe, with advancements in technologies, there have certainly sprouted numerous challenges, but trusted system concept is one of the need of today's world. It allows the ease of communication and sharing with minimal overload on user's side. So, design of these kinds of important systems should be done with great care and by meeting the set standards.

REFERENCES

[1] William Stallings, Cryptography and Network Security, Principles and Practices, 4th Edition

[2] Edwardbosworth, www.edwardbosworth.com/CPSC6126/Lectures/CPSC6126_Ch05.htm?

[3] Jason et. Al., 1998, Digital Rights Architectures for Intellectual Property Protection, MIT and Harvard Law School

[4] Gang Xu, A Policy Enforcing Mechanism for Trusted Ad Hoc Networks

[5] Trent Jaeger, Reference Monitor, Systems and Internet Infrastructure Security Lab, Pennsylvania State University

[6] G. Xu, Borcea and Iftode, Trusted Application-Centric Ad Hoc Networks, Rutgers University, NJIT and AT & T

[7] Purdue University, 2010, Computer Security, CS 426, Trusted Operating Systems and Assurance

[8] G. Qu and Min Wu, Information Hiding Based Trusted Computing System Design, University of Maryland

[9] UC Santa Cruz, Trusted Operating Systems, CMPS 122, UC Santa Cruz

[10] Dr. William Stallings, Use Fundamental Security Design Principles to Design or Evaluate Security Products, www.networking.answers.com

[11] Wikipedia.org, en.wikipedia.org/wiki/Trusted_system?

[12] Illinois.edu, 2011, Trusted System Elements and Examples, wiki.engr.illinois.edu

[13] Trusted Lecture, www.cse.unr.edu/~mgunes/cs450/cs450sp11/Lect22_TrustedOS2.ppt?

[14] Abrahams & Joyce, Trusted System Concepts, the MITRE Corporation

[15] Tobe Leibert, 2000, Features – Should We Trust "Trusted Systems?" University of Texas at Austin

[16] Pfleeger et. Al, 2007, Designing Trusted Operating Systems, Pearson Education

[17] Searchsecurity.com, http://searchsecurity.techtarget.com/definition/trusted-computing

[18] Ola Flygt, Designing Trusted Operating Systems, Linnaeus University, Sweden

[19] Columbus State University, csc.columbusstate.edu/summers/NOTES/**6126**/notes/**6126-ch5.ppt**

[20] Iqware.us, http://www.iqware.us/markets-infra.php

[21] Roger S. Pressman, 2010, Software Engineering A Practitioner's Approach. 7th Edition