TECHNISCHE UNIVERSITÄT DRESDEN

Department of Computer Science, Institute of System Architecture, Operating Systems Group

Distributed Operating Systems

Exercise 4: Trusted Computing

In the tutorial, all solutions will be presented by students. Please be prepared for all questions as the exercise will focus on discussion, not on understanding the question and gathering the knowledge.

Secure and Authenticated Booting

This exercise practices secure booting, employing the example of an Internet-connected game console.

1) The game console runs the following software stack. Is the given reply within the challenge response protocol correct? Justify your answer.

   Application: *A*

   Operating System: *OS, ID, OSrunning*

   OS vendor: *OSV*

   Platform: *CPU, TRB with EK, SRK, AIK*

   Platform Vendor: *TVK, certifies {"good EK",EK^{pub}}TVK^{priv}*

   Certification Authority: *CAK, checks and certifies that AIK belongs to TRB with EK (not shown here)*

   Challenge: *nonce*

   Response: *{nonce}$OSrunningAuthK^{priv}$, {IDOS, $OSrunningAuthK^{pub}$, $OSVK^{pub}$}AIK^{priv}, {"good AIK", AIK^{pub}}CAK^{priv}*

2) Describe how trusted-computing technology can be applied to protect the integrity of game-related data (e.g., the integrity of files describing avatar properties).

3) Describe in detail how application A can check whether it has been securely booted, or why this is impossible.

4) The first version of the game console should implement a closed system. Design a boot protocol which allows only licensed applications to be started. Describe the limitations of the resulting boot protocol respectively the hard- and software properties required to overcome these limitations.

5) An extended version of the console should be produced as an open system. This console should allow online players to play together, provided each player has a private legal copy of the game. How does secure booting help to prevent

   a) cheating

   b) the use of illegal copies?

Hardware and Software Requirements for Trusted Computing

6) Discuss how the cryptographic primitives and protocols needed for "secure/authenticated booting" and "remote attestation" can be integrated into the hardware and software stack, such that the desired protection goals can be met. Consider the following aspects and design questions:

   a) What are the protection goals?

   b) Who's the attacker? What can be assumed about her capabilities and means of access?

   c) What has to be built into hardware? What can be done in software?

   d) Which security properties must be enforced by hardware components, firmware, operating system, and applications?