

A Generic Attack on checksumming-based Software Tamper Resistance

Wurster, Van Oorschot and Somayaji

*Presented by
Sandra Rueda*

Software tamper-resistance:

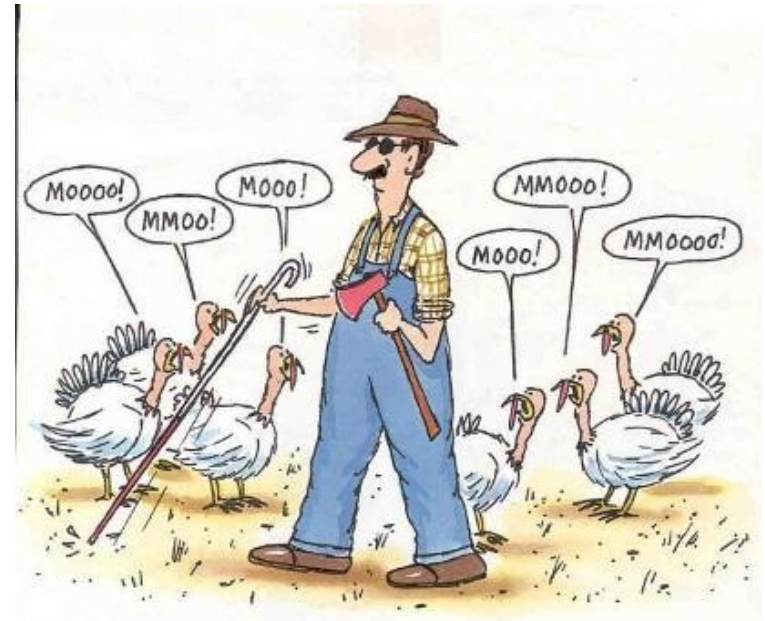
if program is modified, the action must be detected

- To prevent circumvention of :
 - Copyright protection (DRM)
 - Security mechanisms

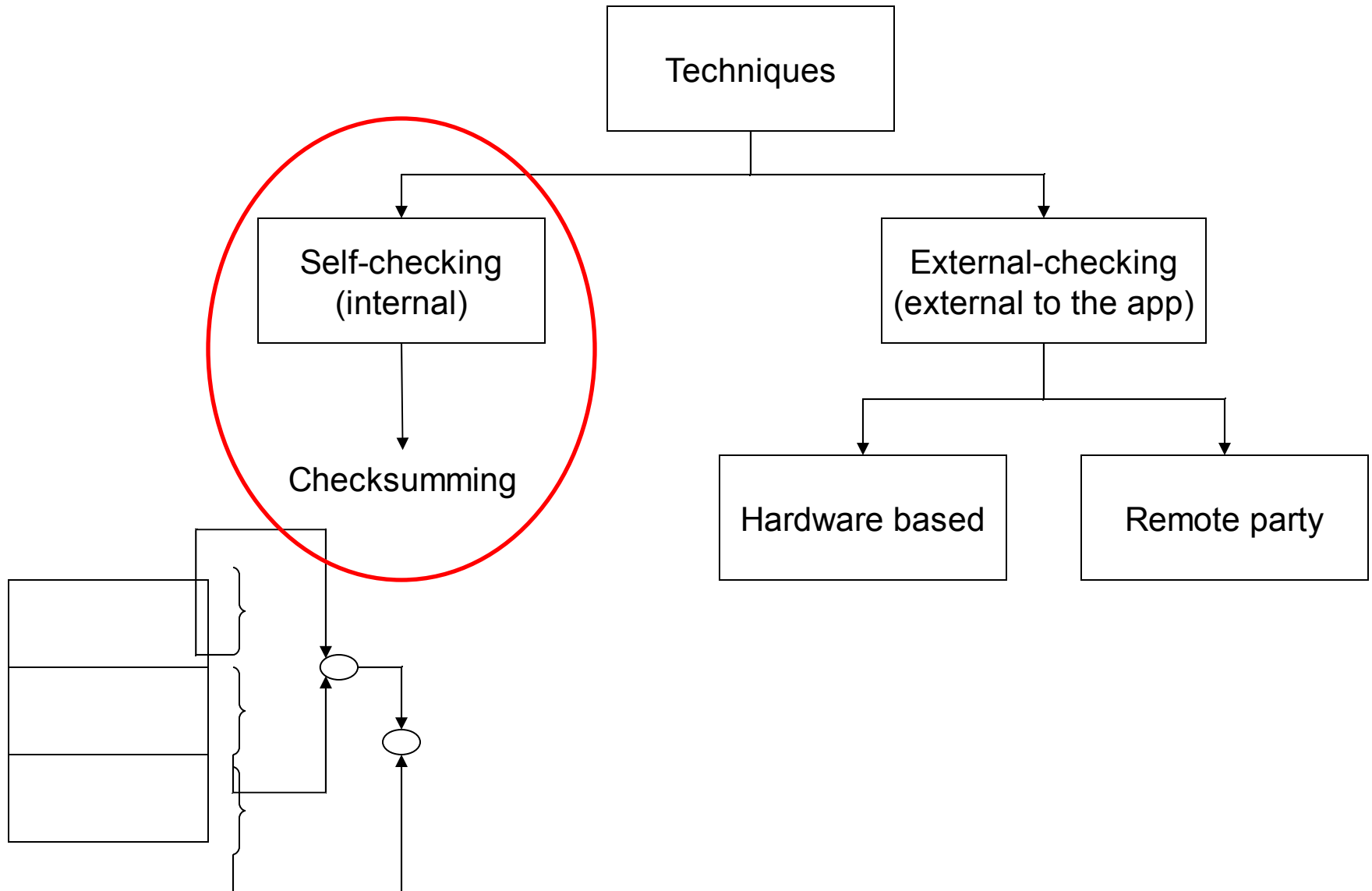


Threat Model

- Cryptographic hashes are used to guarantee integrity in multiple cases, but they are not enough in this case ...
- Hostile Host Model
 - Host system is untrusted
 - Host administrator is untrusted
 - OS may be manipulated
 - State of resources may be manipulated
 - Other applications may be manipulated



Tamper Resistance Techniques



Attack

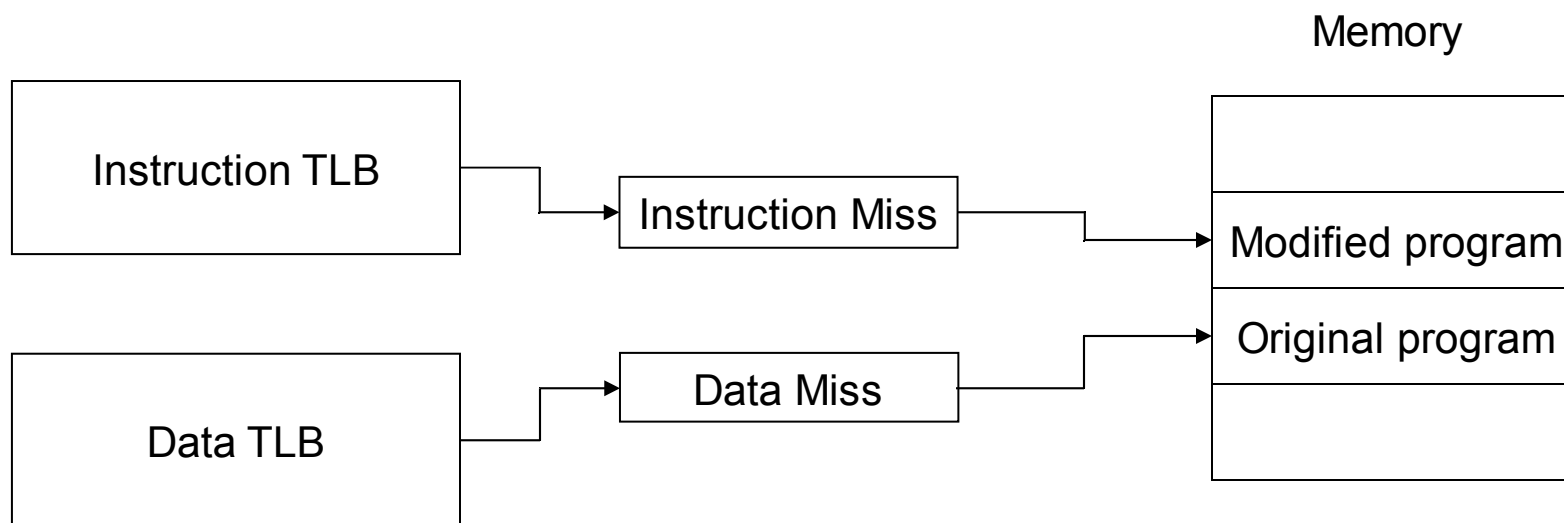
- Copy the original program
- Modify the copy
- Modify the kernel
- Run the modified code under the modified kernel



- *The program should believe that the original version is running although the modified one is the one that is actually running*

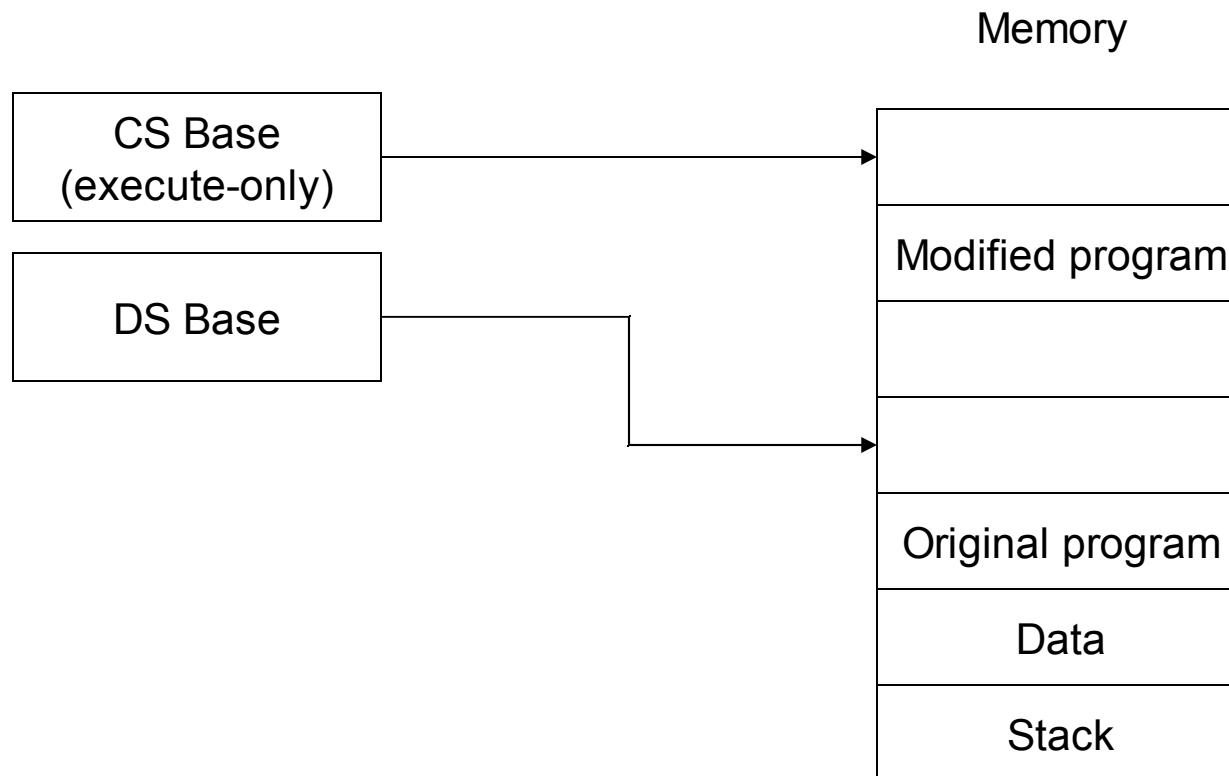
Kernel Modification

- UltraSparc
 - Modification depends on the architecture

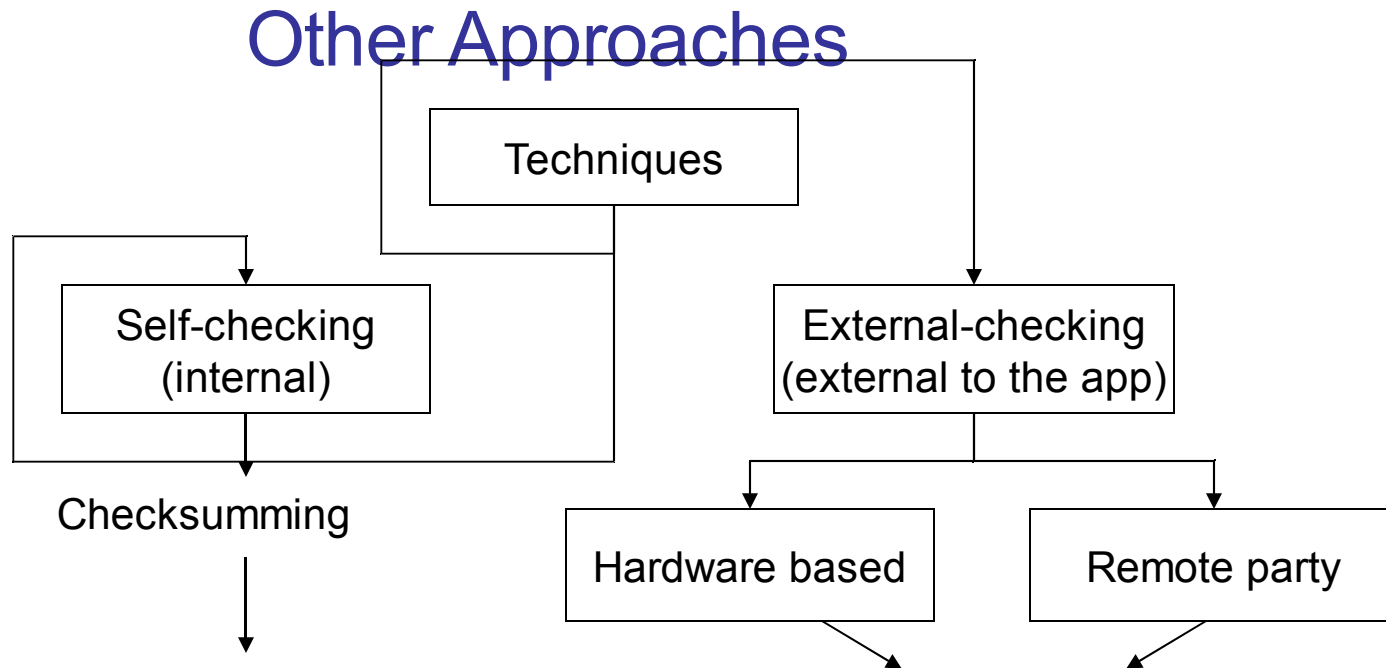


Kernel Modification

- x86
 - Modification depends on the architecture



Tamper Resistance Techniques



- Intermediate computations
- Encryption/Decryption
- TPM
- IMA
- Secure Hardware

- Do they apply in every case ?
- Do they apply at boot time and also at run time ?
- Are there assumptions about trusted components ?

Tamper Resistance

- From the program point of view:
 - Processes do not have direct access to the memory pages nor to other structures in the kernel in which they are running. How to differentiate real from fake ?
 - Would external devices have advantages in this case ?
- From the attacker point of view:
 - Could virtual machine technology be used ?
 - Could emulators be used ?
 - The attack may increase program run-time but .. is performance important in every case ?



Take Away

- Hostile-host problem is difficult because programs have no direct access to data
- Any architecture that make it possible to differentiate between code and data is susceptible to the proposed attack

