

as an answer key; but if you do, please don't feel limited to or constrained by them. There are other example threats.) In contrast this section presents a model without an answer key. It's a lightly edited version of a class exercise that was created by Michael Howard and used at Microsoft for years. It's included with their kind permission. I've personally taught many classes using this model, and it is sufficiently detailed for newcomers to threat modeling to find many threats.

Background

This tool, named iNTegrity, is a simple file-integrity checking tool that reads resources, such as files in the filesystem, determining whether any files or registry keys have been changed since the last check. This is performed by looking at the following:

- File or key names
- File size or registry data
- Last updated time and date
- Data checksum (MD5 and/or SHA1 hash)

Architecturally, the tool is split into two parts: a host component and an administrative console. As shown in Figure E-4, one client can communicate with multiple servers, rather than running the tool locally on each computer.

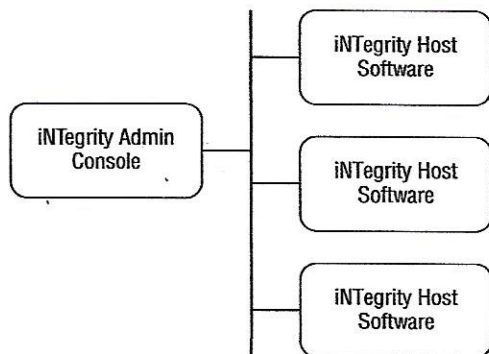


Figure E-4: The networked host/admin console nature of the iNTegrity tool

In another operational environment, it might be known that a machine has been compromised and can no longer be trusted, and the server and client software can be run off, say, a bootable CD or USB drive. In this case, the integrity-checking code is running under a trusted, read-only Windows environment, and the host and admin components both read data from the compromised machine,