Abstract: Protecting a digital asset once it leaves the cyber trust boundary of its creator is a challenging security problem. The creator is an entity which can range from a single person to an entire organization. The trust boundary of an entity is represented by all the (virtual or physical) machines controlled by that entity. Digital assets range from media content to code, and include items such as: music, movies, computer games and premium software features. The business model of the creator implies sending digital assets to end-users – such that they can be consumed – in exchange for some form of compensation. A security threat in this context is represented by malicious end-users, who attack the confidentiality or integrity of digital assets, in detriment to digital asset creators and/or other end-users. Software obfuscation transformations have been proposed to protect digital assets against malicious end-users, also called Man-At-The-End (MATE) attackers. Obfuscation transforms a program into a functionally equivalent program which is harder for MATE to attack. However, obfuscation can be use both for benign and malicious purposes. Malware developers rely on obfuscation techniques to circumvent detection mechanisms and to prevent malware analysts from
understanding the logic implemented by the malware. This chapter presents a tutorial of the most popular existing software obfuscation transformations and mentions published attacks against each transformation. We present a snapshot of the field of software obfuscation and indicate possible directions, which require more research.

**Dewey Dezimalklassifikation (Liste):**
000 Informatik, Wissen, Systeme

**Kongress- / Buchtitel:**
to appear in advances in computing

**Jahr:**
2017

**Revied:**
ja

**Occurences:**
Einrichtungen > Fakultäten > Fakultät für Informatik > Lehrstühle der Informatik > Informatik 4 - Lehrstuhl für Software & Systems Engineering (Prof. Pretschner)

- Hochschulbibliographie > 2017 > Fakultäten > Informatik > Informatik 22 - Lehrstuhl für Software Engineering (Prof. Pretschner)