
Software Piracy Overview of Anti-Tampering Technologies

**Scott Baeder
Sr. Architect
Cadence Design Systems
baeder@cadence.com**

Agenda

- **Quick Review of “Piracy”**
- **Binary Hacking**
 - Crackers Workflow
- **First Impressions**
- **EDA Requirements**
- **Next Steps**
- **Contact Information**

Quick Review of “Piracy”

- **Three basic forms of “Piracy”**
 - **Counterfeit keys**
 - » (make your own keys)
 - **HostID spoofing**
 - » (clone the server)
 - **Binary hacking**
 - » (modify the executable to remove licensing checks)

Counterfeit Keys

- This is the easiest because the FLEXIm key generation algorithm was cracked way back in V6.x
 - <http://www.woodmann.com/crackz/index.html>
- FLEXIm added “increased” encryption
 - Tamper resistant licensing (TRL)
 - » Licensed from Certicom
 - To my knowledge no one has completely cracked ECC algorithm and generated counterfeit TRL enabled keys

Tamper Resistant Licensing

- **TRL is not a part of the basic licensing “package” from Macrovision.**
- **Everyone in the EDA space should move to implement this in their licensing systems.**
- **BUT, the quality of the result depends on the implementation!**

HostID Spoofing

- A HostID is the unique identifier used to lock a license file to a server.
- Can be a dongle or other HW based identifier
- Very easy to “spoof” (clone) PC hardware IDs
- Home routers can clone an Ethernet HW identifier (MAC address)
 - Commonly used ID on Linux and Windows based PC hardware
- FLEXIm option to “increase” security of hostID called Tamper resistant binding (TRB)

Tamper Resistant Binding

- **Can use additional “items” to create a “stronger” hostID**
 - Can be supplied by vendor
 - Can also make use of trusted storage
- **Some forms of this now included in the basic “packaging” from Macrovision**
- **As with TRL, quality of results when using TRB depend on the implementation**

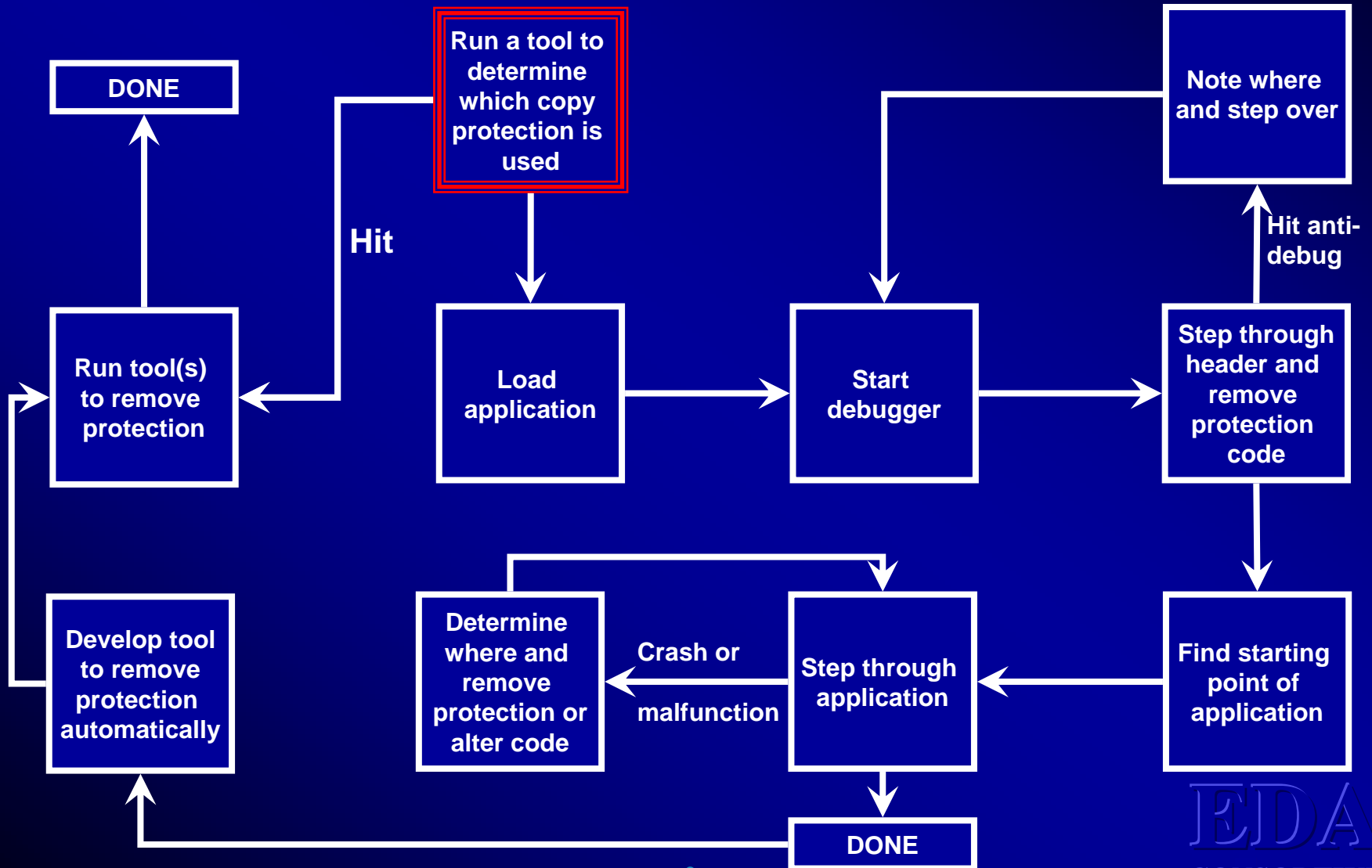
Binary hacking

- **Process of modifying the executable to remove or bypass licensing checks**
- **Has been the method of choice in the PC Software space.**
- **Now that we are closing off the “easy” ways to steal EDA software, more attention needs to be placed here.**

Binary hacking

- **Preventing binary hacking is a very young field**
 - related to DRM (Digital Rights Management)
- **EDAC has had discussion with three major players (so far), plus a few other related companies...**
 - Some came from PC Game industry, others from protecting Military Software, etc.

Cracker's Workflow



Multiple Technologies Are Necessary To Provide Truly Effective Protection...

- **Maximize the work effort to remove “licensing”**
 - Variety of protection installed in “Thousands” of places in the code
- **Impede reverse engineering and automated “licensing” removal**
 - Use anti-disassembler & anti-debug technologies
 - Use encryption, obfuscation, spoofing and authentication
- **Maximize user friendliness**

An example (taken from a vendor web site)

- **XXXXXXXXXXXX** is comprised of three components:
 - **Post Processor** – which is an application that automatically encrypts a software application, embeds a **Secure Execution Monitor** (described below), and adds pre-defined application and security extensions.
 - **Secure Execution Monitor** – which is a set of security functions for encrypting and decrypting application subroutines, ensuring application integrity, and monitoring the run-time environment for malicious activity and unauthorized access.
 - **Application and Security Extensions** which are custom extensions that support software authorization frameworks, enhanced security through the integration of external key management devices/systems, and alternate encryption algorithms.

First Impressions

- **Today, all are PC/Windows based, but moving into Linux and other platforms.**
 - **But unless we cover all platforms, the crackers will just move to the ones that are unprotected.**
 - **Some only protect .exe's and not shared libraries**
 - **Some originally viewed their own technology as copy protection**
 - **BUT anti-tampering technologies are similar.**

First Impressions

- **Macrovision has trusted storage, but no one has really evaluated it yet.**
- **To be really safe you probably need to install “guards” into the source code which requires modifying application**
- **Tools that protect the Militaries embedded software (smart bomb control software, etc.) is the most comprehensive (and the most expensive)**

First Impressions

- **Some use “exotic” means. For example, one uses “imperfections”**
 - All hard drives have highly unique damage areas
 - Same for the silicon in CPUs, RAM, Video Chips etc.
 - Generate a “Device Fingerprint” by using over 10,000 non-user-configurable samples from a typical PC
- **Bottom line – no “Silver Bullet” (yet)**

EDA “Requirements”

- **A comprehensive solution that covers secure hostid, anti-tampering and reverse engineering protection.**
- **Want a solution that works as part of the backend process without source code changes**
 - even though that offers the best protection and could eventually be required.

EDA “Requirements”

- **Doesn't rely on any “phone home” mechanisms**
- **So far, all the vendors we have talked to only cover part of the solution**
 - but any of them may be an important part of a comprehensive solution.

Next Steps

- **EDAC's Anti-Piracy committee will continue to follow trends**
- **Some of us are already engaged in evaluations and trial deployments**
- **If you want more information on what we are doing (and your company is an EDAC member), contact me to get involved.**

Contact Information

- **Scott Baeder**
baeder@cadence.com