Making Reverse-Engineering Harder

Dr. Robert W. Baldwin
Plus Five Consulting, Inc.
Baldwin@plusfive.com

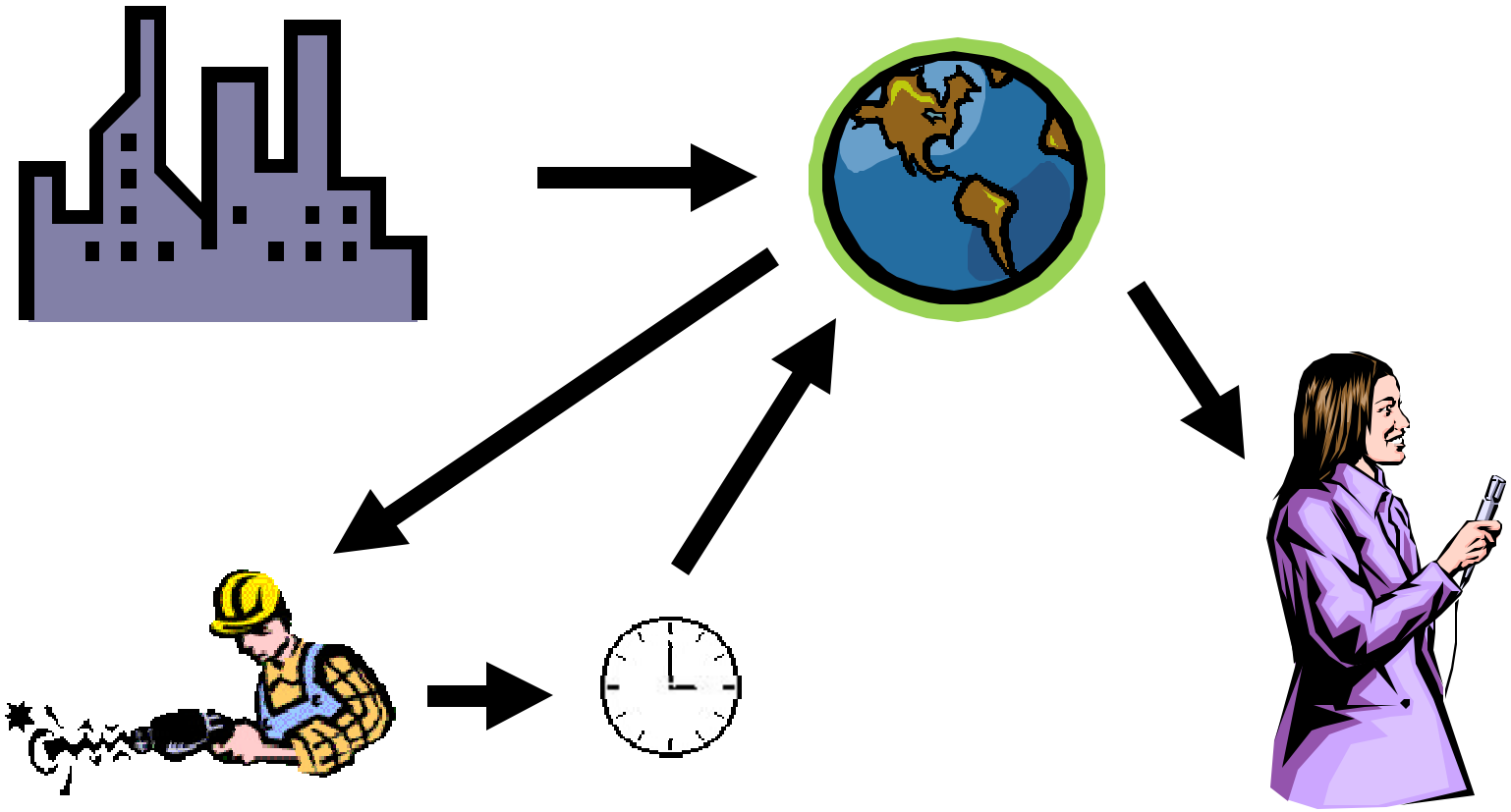
Outline

- Goals & Process for Crackz
- Nag Screen & Registration Code
- Measures and Counter Measures
- Cracker Tools
- Recommendations

Cracker Goals

- Personal Fame
- Fight Evil Capitalists
- Free Entertainment
  - End-Users of Crackz

Process for Crackz



Observations

- End Users Need No Talent
  - Special Search Engines
- Crackz Files are Tiny
  - Easy to Host. Lots of places to Host.
- Vendors Want to Make Effort Large
  - Even hardware hacking is possible
    Ex: Cable Modems, Cell Phones.

Choke Points

- Work Required Per Release
  - Old Version Might be OK
- Trick Patch Tool
- Scare End Users

Outline

Registration Nag Screen

- Program Splash Screen Nags About Unregistered Version
  - WinZip
- Cracked Version Could
  - Remove Nag Screen
  - Provide Valid Registration Codes

Code For Registration Check

```
Do_Init();
if (Is_Registered()== False)
    Show_Nag();
Do_Real_Work();
```

Asm For Registration Check

```
Call 040100     // Do_Init
Call 040480     // Is_Registered
Cmp eax, eax    // if non-zero
Jne $123        // skip next
Call 040600     // Show_Nag
$123:
Call 041000     // Do_Real_Work
```

Patch For Registration Check

```
Call 040100
Call 040480
Cmp eax, eax
Jne $123
NOP
$123:
Call 041000
```

Registration Generator

- Publish Program to Generate Valid Registration Codes.
  - Usually Function of User Name
  - Sometime PC Info → Problems with Upgrades
- Step One: Find Code That Checks
  - Locate Based on Dialog Box Resources
- Step Two: Copy Code to New Program.

Finding Code

- Run Executable Through Disassembler
  - Dialog Box Strings Identified
  - Uses of These Strings Indexed
- Breakpoint & Stack Trace Finds:

```
expected = Gen_Code (Name)
if (!strcmp (expected, actual))
   Show_Bad_Code_Msg ();
```

Copy Code

- Disassembler Produces Valid Code
- Download Framework for Code Generator
- Cut & Paste Gen_Code Routine
  - Need to get Subroutines & Globals
  - This can be hard.
- Test & Publish.

Outline

Protection Measures

- Encrypt Code to Thwart Disassembly
- Bootstrap Code Cannot be Encrypted
- Code must be self-relocatable
  - Loader/Linker Cannot Relocate Encrypted References

Counter Protection Measures

- Cracker Tool Dumps Memory of Running Process After Decryption.
  - Part of Run Time Debugger
- Counter-Counter Measure:
  - Decrypt Sections As Needed,
    Re-Encrypt When Not Needed.
- Scan for All Self-Decryption Code.

Protection Measures

- Detect Runtime Debugger
  - SoftIce
  - IDA
- Found on Disk
- Found Active in Memory
- Problems For Paying Customers

Counter Measures

- Avoid Debugger Detectors:
  - Modify File Names
  - Modify Call Table Names
  - Append Garbage to Change Checksum
- Standard Tools Available for These Tricks

Outline

Disassemblers

- Static & Runtime Disassembly
- Knowledge of Executable Formats for Different Compilers
- Displays OS Calls & Names Parameters
- Cross Reference GUI Resources
- Rapid Testing of Patches

Disassemblers

- Conditional Breakpoints
  - Tenth Call, Called when x = y+1
- Break on Memory Read or Write
  - Finds Code Using Global Variable
  - Ex: Separate Reading & Checking Clock Tics
- Dump Regions of Data & Code
- Produce Listing to Allow Re-Assembly

Process Monitoring

- Log All File Access (FMON)
- Log All Registry Access (RMON)
- Log All OS Calls (WinMON)
- Log Network Traffic (tcpDump)
- Logs Find Persistent Storage Used by Program (e.g., Key and Configuration)

Legitimate Tools

- CleanSweep – Norton Tool to Properly Remove Programs.
  - Thwarts Demo Protection Based on Secret File or Secret Registry Entry.
- Search Engines
  - “+Crack +RealMedia”
- Free Web Hosting

Recommendations

- Design-In Protection From Start
  - Non-Modular Code: Inline Coding of Checks
  - Global Variables As Subroutine Parameters
  - Redundant Variables and Copy Operations
  - Unnecessary Multi-Threading & Processes
  - Compute Obvious Constants (Crypto, Dialog)

Recommendations

- Don’t Include All Code in Demo Version
  - Make Registration Key Decrypt Code
- Make Patching Harder:
  - Vary Checksum of Each Downloaded Version
  - Vary Do-Nothing Code of Each version
    - Especially in Security Routines
- Separate Detection of Problem From Complaint By Several Seconds

Recommendations

- Read Cracking Tutorials
  http://www.home.aone.net.au/byzantium/tools/tools.html
  http://www.instinct.org/fravia/
  http://www.bokler.com/bsw_crak.html
  http://freeweb.coco.cz/Berserka/tutorials.htm
- Use Cracker Search Engines
  http://astalavista4.box.sk/

Conclusions

- Most Programs Easy to Crack
- Tutorials Available to Teach How To Make Reverse-Engineering Harder
- Decreasing Returns With Time
- Designed-In Security Better Than Add-On