CS 592: Security Practicum

Lecture 1

Introduction

Instructor

- Wu-chang Feng
  - Office: FAB 120-17
  - Office hours: After class
- Course information
  - Join pdx-cs592@yahoogroups.com
  - Goto http://groups.yahoo.com/group/pdx-cs592
  - http://www.thefengs.com/wuchang/work/courses/cs592

Good news, Bad news

- Bad news
  - This is not the course described in the catalog
- Good news
  - You can still drop the course if you do not like what we’ll be doing

What is this course?

- Precursor to a research project with Intel
- Hardware mechanisms to detect execution of undesirable software
  - Focus on cheating in on-line games

Course objectives

- Examine Intel’s AMT platform
- Survey a variety of cheats across game genres
- Understand the software mechanism used by cheat software
  - How it hooks into OS and game
  - How it hides from detection
- Understand anti-cheat software and design anti-cheat mechanisms based on Intel’s AMT platform

Why take this course?

- Latch onto an on-going research project that is being pursued in conjunction with Intel’s research laboratory
- Potential for research assistantship position ($) if project funded by Intel
- Potential for a summer internship if your project is successful
- Inside track to a future job at Intel

Course organization

- Teams of two will tackle a particular genre
- Genres to choose from
  - Cheats for real-time strategy (RTS) games
    - Maphack for Warcraft 3
  - Cheats for massively multiplayer on-line role playing (MMORPG) games
    - WoWGlider for World of Warcraft
  - Cheats for first-person shooters (FPS) games
    - OGC for Half-Life games
  - Anti-cheat systems
    - Warden for WoW, PunkBuster for Unreal Tournament

Phase I (1 week)

- Introduction to course
- Introduction to Intel’s AMT (Active Management Technology) platform

Phase II (4 weeks)

- Survey the cheats of your chosen genre
- Give a slide presentation summarizing results
  - In class on 5/1/2007
  - For each cheat, have the following bullets on your slide
    - Web link for cheat
    - What game does it modify?
    - How does it give a player an advantage?
    - What is its software mechanism?
- What to turn in:
  - Tarball that includes slides and software for each cheat (if possible)
  - Grading will be based on thoroughness of survey

Phase III (3 weeks)

- Analyze software architecture of a chosen cheat/anti-cheat
  - Cheats
    - How does it modify the game?
    - How does it avoid detection?
  - Anti-cheats
    - How does it perform detection?
    - What does it measure and when does it measure it?
    - How can it be subverted?
- Give a slide presentation summarizing results
  - In class on 5/22/2007
- What to turn in:
  - Tarball that includes slides and software for chosen cheat

Phase 4 (2 weeks)

- Apply the AMT system to your chosen cheat/anti-cheat
- Emulate AMT using a multi-core or multi-processor
  - Create a monitoring process that detects the presence of the cheat
  - Examine how to adapt the anti-cheat to the AMT platform
- Give a slide presentation summarizing results
  - In class on 6/5/2007
- What to turn in:
  - Tarball that includes slides and/or code that demonstrates the ability to use the AMT approach to properly detect chosen cheat

Evaluation

- Phase 2 (Survey) = 40%
- Phase 3 (Disassembly) = 30%
- Phase 4 (AMT counter-measure) = 30%