A NEW VILLAIN: INVESTIGATING STEGANOGRAPHY IN SOURCE ENGINE BASED VIDEO GAMES
Table of Contents

- History behind platform
- Impact of platform
- Creating game levels with hidden data
- Investigating these levels to recover information
- Conclusion
- Future Work
The Source Engine

- Created by Valve
- Two ex-Microsoft Employees started in 1996
- Began with the release of Half Life in 1998
- Originally a modified version of the Quake gaming engine
- Known initially as $Gldsrc
- Modified further into Source engine
The Source Engine – Cont’d

- More commercial success
- Counter-Strike released in 2000
  - Most actively played online game in the world
- Need to aggregate and control game patches
- Steam was released in 2003
One of the leading game engines in the world

Released titles such as:
- Half Life 1 & 2
- Portal 1 & 2
- Left 4 Dead 1 & 2

Ongoing constant development
What is Steam?

- PC based gaming solution
- Store
- Game Management
- Statistic Aggregation
- Patch Aggregation
- Social network

- Currently in Development – Steamworks API
The Steam Interface
Steam Usage

- 1523 games available
- 40 million active user accounts
- 5 million concurrent players on January 2, 2012
- 70% of the digital distribution market in 2009
- Continual growth
Hammer

- Official level (map) creation tool
- Used on all Source games
- Free with Source games
Tools Within Hammer

- Hammer is a set of tools to create, develop, and publish Source maps
- Main game creation interface
- Game logic
- Tools to compile map data into playable levels
Exploiting the Source Engine

- Main focus of this project
- Use video game files to hide data
- Text Messages
- Images
- Steganography
What is Steganography?

- Hiding Data Within Data
- Security Through Obscurity
- Only Sender/Receiver Recognize Data
- Advantages Over Encryption
Why Video Games?

- Size – Plenty of room to hide data
- Common – Video game installations are not out of place on computer systems
- Dynamic – Video game files are intended to change repeatedly
- Untraceable Information – Data hidden in these files cannot be viewed on a dead system
- Open Source Files - Source specific
Embedding Text With Brushes

- Brushes are main level geometry
- Brushes can be manipulated to form words and messages
- Most basic data hiding technique
- Easy to accomplish
- Tedious to execute
- Impossible to detect on disk
Embedding Text with Overlays

- In-Game messages
- Physical locations
- Implemented with Entities
- Env_instructor_hint
- Info_target
- Relatively easy to implement and use
- Detectable on disk by investigator
Embedding Images with Textures

- Developer jargon for images
- Image handling by Source - VTF
- Size considerations
- File format
- Metadata file
- VTFEdit
Embedding Images with Textures

- Once images are converted, they can be added to the map
- Face Edit tool
Map Distribution

- VPK File
- VPK File Contents
  - Level Data
  - Textures
  - Assets
- VPK Tool
- Distribution
- Installation
Demonstration!
Investigating Source Games

- Source games can be used to hide data
- Investigators must have a way to recover this data
- Forensic Toolkit (FTK) used for investigation
Issues Facing Investigators

- Multitude of game files
- Size of game file installations
- No native support in investigative software
- Reliance on non-forensic level tools
- Viability in court
The First Step: Finding Game Files

- The first step in the investigative process is to identify and locate game files
- Two main approaches
- Game directory structure
  - Steam\steamapps\common\gamename\addons
- File header
  - 0x55aa 1234
Finding Game Files – cont’d

- Once a VPK has been found, it must be decompressed and unpacked
- GCFScape Tool
- Allows users to view and extract files from a VPK
- Used by an investigator to work with data
Investigating Data Hidden with Brushes

- Impossible to do
- Cannot be detected in disk
- Only visible when game is played
Investigating Data Hidden with Overlays

- Data hidden in overlays can be recovered on disk
- VPK file must be decompressed
- Data resides in `mapname.bsp` file
- Stored in “entity lumps”
- Search for keywords
- “`hint_caption`” followed by message
- "`hint_caption" "Malicious information here!""

Malicious information here!
Entity Lump
{
  "world_maxs" "480 480 480"
  "world_mins" "-480-480 -224"
  "maxpropscreenwidth" "-1"
  "skyname" "sky_wasteland02"
  "classname" "worldspawn"
}
{
  "origin" "-413.793 -384 -192"
  "angles" "0 0 0"
  "classname" "info_player_start"
}
Investigating Data Hidden with Overlays - cont’d
Investigating Data Hidden with Textures

- Identification
  - File System structure
  - Header
  - 0X5654 4600 0700 – VTF\0

- Once identified, textures can be investigated

- VTFEdit may be used
Conclusion

- Data privacy is a right of every individual
- Sometimes this right can be abused
- Data can be hidden in Source game files
- Investigators have ways to recover this data, albeit rudimentary
- The widespread impact of data hidden in this way drives demand for solutions on both sides
Future Work

- New methods of data hiding
- New methods of data recovery
- Development of investigative tools
  - Support for Source files in FTK and others
  - Forensic verification
- Expansion to other game engines
- Expansion to other platforms
References


References cont’d.


References cont’d.


References cont’d.

Questions?