Leveraging 3D Game Engines (L3DGE)

Novel techniques for anomalous traffic detection and collaborative network control

Overview

- Precursors to L3DGEWorld
- Cisco URP
- What is L3DGEWorld?
- Evolution of L3DGEWorld
- L3DGEWorld's Metrics
- Demos
Precursors 1
- OpenGL application
  - 3VEN (3D Visualisation Environment for NIDS)
- GLUT based
  - Primitive
  - Hard work
- ATNAC 04

Precursors 2
- Cube engine
  - Very early version – contained 'monsters' as avatars
Precursors 3

- Cube engine (not Quake III)
  □ Many hacks to get greynet info into a cube map / actions out

- VizSec06

URP

- Awarded June 2006,
- "Anomalous Traffic Detection and Collaborative Network Configuration Using 3D Multiplayer Game Engines"
- ...or L3DGE (Leveraging 3D Game Engines)
- Cisco Champion: Fred Baker
- Grenville Armitage, Warren Harrop
- Money bought on Lucas Parry
- Allowed app development to expand in scope
What is L3DGEWorld

- Data visualisation and Control tool
- Based on GPL'd game OpenArena (http://openarena.ws)
- Designed to be Modular
  - L3DGEWorld Server & Client
  - Input Daemon(s)
  - Output Daemon
- Primarily developed for monitoring networks

Input and Output Layers
OpenArena Networking

- Server keeps “gamestate”
  - everything players need to know
- Connecting client sent entire “gamestate”
- From then on, incremental diffs are sent
- If client fails to ack an update, server diffs current update to last ack'ed update
**LTMON 1.0**

- Server side modification for Q3A
- No changes on the client side
- represented traffic with “columns”

---

**L3DGEWorld 1.0**

- Q3A mod, client and server side
- manipulates entities based on statistics
- Textual labels on entities
- Persistent host postitions
- Primitive greymatter and monitorhosts.sh
- using the file system for abstraction layers... BAD!
L3DGEWorld 1.0

- Why client side?
  - Normally entities are controlled by the clients independently
  - Good for network efficiency
  - Bad for us
- Found an unused part of entity state structure
- manipulate entity based on this field
L3DGEWorld 1.5

- Found OpenArena
  - Based on IOQuake3 (improved Q3A engine)
  - Fully GPL'd resources (textures, models, etc)
- Updated L3DGEWorld to use only OA provided resources
- Allowed us to distribute a complete standalone product

L3DGEWorld 1.5
L3DGEWorld 2.0

- Administrative weight for actions
- Detailed information window
- Configurable interaction with hosts
- Greatly improved greymatter
  - proper averages
  - more statistics generated
- Requires custom executables
L3DGEWorld 2.0

- Needed more fields to get data to clients
- Had to modify entitystate
- Greatly improved code readability

L3DGEWorld 2.1

- 9 generic metrics. easy to re-purpose L3DGEWorld. see: LCMON
- Each metric has a “name”, “value” and “rate”
- UDP based input abstraction layer, based on RCON
- More generic outputted actions
- Ran into many problems behind the scenes
L3DGEWorld 2.1

- UDP input required modification to the server portion of the code
- Storing all the metric data as configstrings
- Hit many limits
  - Total maximum string data
  - Maximum initial gamestate size before fragmenting
  - Had to minimize the amount of unnecessary stuff stored in gamestate
UDP Input Protocol

- Based on RCON
  - Used by Q3A for remote administration
  - Human readable ASCII messages

- Added very basic authentication
  - Daemon sends “gettoken” request
  - L3DGEWorld replies with “token XXX” (16 character token)

UDP Input Protocol

- Daemon includes the token in update messages
  - “l3dge XXX ~a~b~c~d~”
    a) host number
    b) metric number
    c) metric field (name/value/rate)
    d) what to set it as

- If L3DGEWorld server receives an update with an invalid token sends “invalidtoken” message to sender.
LCMON 1.0

- Super computer monitoring software
- Utilises L3DGEWorld 2.1
- Created in a few short weeks!

L3DGEWorld 2.2(?)

- Not yet completed
  - UDP based output layer
  - greymatter improvements
  - New LRCD output daemon
Metrics of L3DGEWorld

- Metrics
  - Spin
  - Scale
  - Bounce Height
  - Bounce Rate
  - Roll
  - Colour
  - Alternate Skin
  - Alternate Model
  - Sound

Metrics Demo
nmap demo

- Spin rate = PPS
- Bounce height = No. of attackers
- Scale = No. of unique ports
- Colour = Traffic % (TCP/UDP/ICMP)