This documentation is geared towards HLGuard v1.8

Welcome to HLGuard, the server-side anti-cheat system for Half-Life MODs!

Even with the limitations working with a very old Half-Life engine, HLGuard continues to improve techniques in detecting and blocking cheats.

Here's a list of features that are found in HLGuard:

Wallhack Block
Rapid/Burst Detection
Alias & Cvar Detections
Name Violation Detections
SkyWalkGuard
Blacklist checking against Steambans.com

Wallhack Block

A wallhack is a program which makes the walls of a map appear transparent. This allows the player to see his enemy anywhere in the map. Obviously this destroys all gameplay and fair play within a game.

This is where HLGuard's WallHack Block comes in!

HLGuard WHBlock tackles this problem at source by stopping the data which details the enemy's position being sent to a player if there is a wall between the player and his enemy. Because of this method, and the fact that no data is actually sent, there is no possible way ANY wallhack can be used effectively on a HLGuard WHBlock enabled server.

Now there has often been some confusion about the WHBlock because of the way it works. It doesn't stop the walls becoming transparent. It doesn't stop the player seeing his own team mates. It doesn't stop soundhacks which take data from the sound a player makes. It doesn't stop a wallhack working on a player while he is spectating. It does, however, stop all alive players seeing all alive enemies, which is the only important bit.

So why don't we stop all players from seeing any player if there's a wall between them, regardless of whether it's team mates or if they are spectating? The simple answer here is resources.

The WHBlock takes up a lot of processing power. It's without doubt the most CPU intensive feature of HLGuard. To reduce the overheads it requires we only apply the

WHBlock exactly where it’s needed and that’s between a player and his enemy. The server has to take into account all the players’ positional data and enable the WHBlock on a player every time a wall comes between him and his enemy. It also has to disable the WHBlock just before the player comes around a corner. That requires quite a lot of processing, hence the fact that WHBlock works only on alive players.

You'll notice I said the block is disabled just before the player reaches a corner. This is done in this way because if we disabled the WHBlock at the exact moment the player came around the corner and either the player or server lagged, the player would appear to just popup out of nowhere as if he was teleported there. The downside to this is that wallhackers can see players who are very close to the corner of a wall. But this isn't a huge downside as it only gives them a spilt second notice of a player, as opposed to knowing their exact position the whole time.

F.A.Q about WH Block

Q - Hey! This thing doesn't work, I can still see players! This wall hack beats the wall hack block!

A - You'll find that is probably a sound hack, I think you have a wallhack & sound hack mixed up.

A wallhack makes walls transparant so that the player can see the enemy - this works because the HL engine sends info to the player regarding enemy position even when the enemy is behind a wall. So what HLG does is check if the player can actually see then enemy, if no then it stops the enemies positional data being sent from the server to the player. There is no way this can be bypassed/hacked because the positional data is never sent, so there is nothing to hack.

A sound hack takes the enemies positional data from the sounds the enemy makes e.g. jumps, footsteps, gunfire, buying weapons, reloads etc and then draws the players image where the sound came from. While this does give the impression of a working wallhack, it is not. If you were to disable footsteps on the server the soundhack would become much less effective and in alot of cases not work at all.

So the facts are: Wallhacks are blocked, soundhacks are not. While we are working on trying block soundhacks it is not a easy thing to do without completely ruining a players gaming experience and its something that looks more and more difficult. But we will keep trying.

Rapid/Burst Detection

The Rapid/Burst detection monitors the speeds at which a player fires his gun and detects the use of anything which allows a player to fire faster than *should* be possible. This detection is great for detecting multihacks, rapidfire scripts, burstfire scripts and also mousewheel triggers.

An unintelligent newbie once explained very well what this system can do:

QUOTE (ASDFASDFASDF)
Me and a couple of my buddies were playing with duelies in a public server with HLGuard

on it. We were like, "Dudes, we should use mousewheel as our fire button and make our duelies shoot super fast." So we did. About a second later, we were all banned for "Rapid/Burstfire Hacks/Scripts"...

So remember: Using a mousewheel as fire **WILL** be detected, as it allows players to fire at an inhumanly rate, especially with certain guns.

Alias/Cvar Detections

When a cheat script or cheat program is used in HL based mods, it sometimes creates an alias or Cvar (Command Variable) which is used to control the cheat. HLGuard offers a feature which can scan all players in the game for any of these known alias/cvars.

Inside HLG's encrypted checks configs are lists of hundreds of these alias/cvars. As soon as a player connects to the server HLG starts scanning them. If they are found to be using a cheat, HLGuard will then announce that a cheat was found, display their name and a brief description of the cheat found. HLGuard will then kick or ban that player depending on the severity of the cheat found.

Once HLGuard has detected and dealt with the cheating player, it will log the details including player name, IP address, Steam/Won ID number and the cheat found. This info can be found in the logs directory in the hlglist.htm & cheaterlist.txt files.

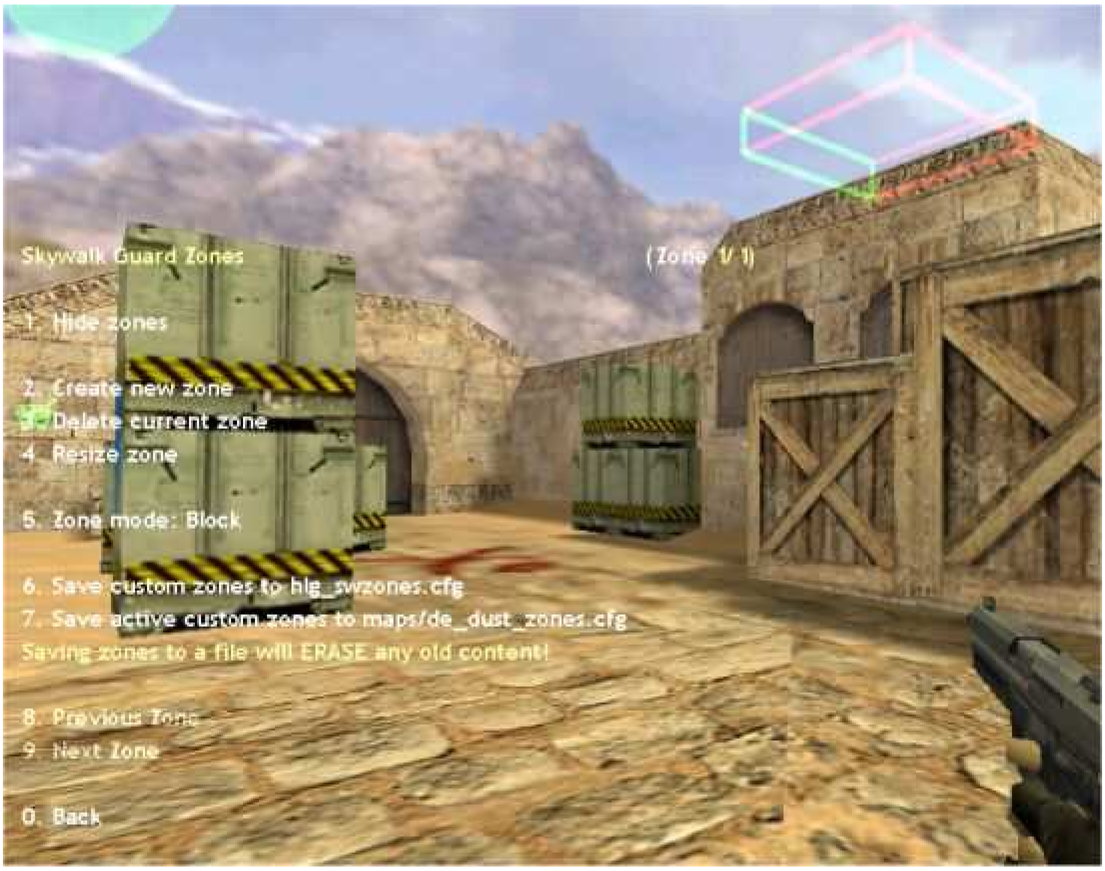
Name Violation Detections

HLGuard has a very versatile method of handling players names. It allows the admin to set rules for the use of certain names e.g. offensive names, name exploits, cheating clans etc. It also monitors the rate at which a player changes his name to stop the use of name spam and fast name changes scripts which attempt to make it hard for a admin to ban a player. All these actions can be configured in the hlguard configs.

SkyWalk Guard

Skywalk Guard basically prevents various kind of map exploiting. HLGuard ships with pre-defined zone files containing data of known exploit locations in a number of default maps. Custom zones can be added as well using the advanced in-game zone editor. This way skywalking can be prevented, and the "jump-through-ceiling" bug detected. If a player tries to enter a SkyWalk zone, he will be pushed back. If he tries to use the "jump-through-ceiling" bug, he will be punished according to the server configuration.

Preview of a SkyWalk Guard zone and the main SkyWalk guard menu:



Zone Mode explanation:

* None -> Zone will do nothing, and will turn invisible when the SWG Zone menu is closed
* Block -> Zone will block players from moving through it, and punish players using the "Jump-Through-Ceiling" bug against it. This zone will fit most uses.
* Block + Force no beams -> Zone will act like "Block", but when the menu is closed, the zone will turn invisible. This is useful for zones placed inside ceilings to prevent the clients rendering the zones (each zone uses 12 .

Notes about SWG:

* Boxes (the zones) cannot be rotated.
* Even though you can edit the zones live in-game, we recommend creating a test server for the purpose, as the gameplay will continue even in the zone edit mode.
* You can place zones inside small ceilings to detect the "jump-through-ceiling" bug.
* Zones can be displayed in-game, but the HL engine limits the number of beams (which the zones are displayed with) in one view, so too many zones in one area might cause the last-added to turn "invisible".
* Zones are saved by HLG to a .cfg file in the HLGuard folder. Zones can therefore be loaded via normal console commands, or by executing a .cfg file with the neccesary commands.
* When a player tries to enter a SWG Zone, he will be teleported back to his previous position, however, if a player in the mean-time moved there, there is a chance they will get stuck in eachother.
* The system is designed to run with players of normal HL size. HLDM, TFC, CSTRIKE and the like should run fine, but if a mod uses players of a different size, HLG will still calculate collisions with boxes according to the default HL model sizes. Support for other mods might come later.

Blacklist checking against Steambans.com

All players joining a HLGuard protected server can be checked for previous cheating offenses against the steambans.com blacklist. For more information regarding steambans.com and their methods for compiling their blacklist visit: www.steambans.com

How to install HLGuard

HLGuard is a metamod plugin, so you need to have metamod installed on your server. For more information on how to install metamod see here:
http://metamod.org/metamod.html

Once metamod is installed you can then install HLGuard.

To install HLGuard you need to unpack the release into your mod folder, the correct path for hlguard should be: ***MOD/addons/hlguard/***
After you have unpacked the files in the correct place, you then need to edit the metamod/plugin.ini file.If you do not already have a plugins.ini file in the MOD/addons/metamod/ directory, you need to create one yourself.

In linux, simply type

touch plugins.ini

In Windows, go Start -> Run... Then type in the text box

notepad

In the notepad window that comes up, click the File menu and choose Save As... Navigate to the MOD\addons\metamod\ folder, and in the File Name text box type: "plugins.ini"

WITH the quotation marks. This will save the file as plugins.ini and not plugins.ini.txt

Linux Server

1. Edit the file plugins.ini (mod/addons/metamod/) by using a text editor (e.g. 'ee' or 'vi')
2. Add this line after any existing plug-ins:

linux addons/hlguard/dlls/hlguard_mm_i686.so

3. Save and exit the text editor.

Win32 Server

1. Edit the file plugins.ini in (mod\addons\metamod) by using a text editor (e.g. notepad)
2. Add this line after any existing plug-ins:

win32 addons/hlguard/dlls/hlguard_mm.dll

3. Save and exit the text editor.

After installation

After installing HLGuard you then need to have a look at the config files and set them up the way you want. If you have never used HLGuard before then we recommend you leave the settings alone for a few days and use the default settings. Then once you have a better idea on how things work you can go over the HLGuard config files and make any changes you need.

Try to read over ALL config files, most have comments describing what they do and how they need setting.

Support

If you have any questions about HLGuard or need help with anything relating to it, feel free to visit our support forums and post any questions you have and We will do our best to help.

You can find us here: The Z Project's HLGuard forums

Credits.

OLO - Orignal creator of CSGuard, the name which HLGuard started out under.

The UnitedAdmins HLG Team - They turned CSGuard into HLGuard, making it available to many more people.
Special thanks must go to: !2SX!, Hullu, Bob, Wtaag and all the many HLG team members too numerous to name.

The Z Project - The current team developing and maintaining HLGuard.

There have been many many more people who have all given their time free of charge to develop and work on HLGuard, all of these deserve a huge thank you.

HLGuard Docs - Last Updated 16:21 27/10/2005 - [EEV]ZepheR.