Finalized Vulnerability Report for Valorant: Color Detection and Automated Actions Exploit  
  
1. Executive Summary

Title: Automated Mouse Triggering via Screen Capture and Color Detection

Date Discovered: November 25, 2024

Date Reported: November 26, 2024

Reporter Information:

Name: Kemono

Email: [REDACTED]

Phone: [REDACTED]

Severity: Critical

Overview: A script leveraging Windows APIs and Python libraries detects specific in-game colors and automates mouse clicks. The script allows for manual adjustments and randomization of delays to evade Vanguard detection, posing a severe threat to the competitive integrity of Valorant.

2. Detailed Description

Vulnerability Type: Unauthorized Gameplay Automation (Cheating Tool)

Affected Component: Gameplay fairness and integrity.

Technical Description:

The script monitors a specific region of the screen and detects a pre-configured target color (e.g., ([REDACTED])). Default red highlight may differ from pc graphics setting. You can take a screenshot and put it in an colorpicker and find the hex code for the highlight for eg: red, purple, yellow.

Upon detecting the color within the specified tolerance, the script simulates a left mouse click using the Windows API.

Customizable Timing: Users can manually adjust the delay (time.sleep) and introduce randomized intervals ([REDACTED](0.24, 0.36)) to mimic human-like reaction times.

Evasion of Vanguard Detection: By varying timing and simulating natural inputs, the script operates covertly and avoids triggering Riot’s anti-cheat system, Vanguard.

The [REDACTED] library enables real-time analysis of in-game visuals, capturing screen regions undetected by standard anti-cheat mechanisms.

3. Steps to Reproduce

Install the required Python libraries: Pillow, keyboard.

Save the provided script and run it on a Windows system for example VScode.

Launch Valorant and position the in-game crosshair over an object highlighted by the target color ([REDACTED]).

Press and hold the Shift key while running the script also put fire on H in-game setting.

Observe the automated mouse clicks triggered by the detection of the target color.

Manually adjust the delay (time.sleep) or randomize it using [REDACTED] to prevent detection by anti-cheat mechanisms.

4. Impact Analysis

Potential Damage:

Competitive Integrity: Allows players to automate actions like aiming and shooting, granting an unfair advantage.

Reputational Risk: Widespread use of such tools could damage Riot Games’ reputation and compromise Valorant’s competitive environment.

Player Base Dissatisfaction: Trust in matchmaking and gameplay fairness is eroded.

Exploitability:

Customizable delays and randomization make the script difficult to detect.

The reliance on external Python libraries and the Windows API makes it accessible to a broad audience.

5. Mitigation Recommendations

Color Randomization: Randomize in-game highlight colors dynamically to prevent static detection.

Input Verification: Implement server-side checks for unusual input patterns, such as unnaturally consistent click timing or excessive precision.

Anti-Cheat Enhancements:

Enhance Vanguard to detect patterns of external screen capture tools like [REDACTED].

Monitor processes that utilize [REDACTED] or other libraries to manipulate input.

Behavioral Analysis: Detect suspicious gameplay behavior, such as abnormally high accuracy or reaction times that exceed human limits.

6. Supporting Evidence

Attached Files:

Script: [REDACTED]

Debug Image: debug\_region.png

Logs: None provided but can be generated upon request.

7. Disclosure Timeline

Date of Initial Discovery: November 25, 2024

Date of Report Submission: November 26, 2024

Public Disclosure: No public disclosure planned unless permitted by Riot Games.

8. Conclusion

This script demonstrates how minor modifications, such as randomizing delays, can allow it to operate covertly under Vanguard’s radar. Addressing this issue promptly is critical to maintaining the integrity of Valorant’s competitive environment and protecting Riot Games' reputation.