

# DEFEATING ANTI-CHEAT WITH HARDWARE

Trapaceando em jogos sem ser (facilmente) detectado

Ricardo Gomes da Silva

DEFCON Porto Alegre 2019



# \$ whoami



rgsilva.com



debugweshell

# Disclaimer

As opiniões são minhas, sem relação com o meu empregador.  
Não apoio uso de cheating :)



# O que é anti-cheating

- » Implementações específicas ou software de terceiros
  - » Validação server-side
  - » Ofuscação de memória
  - » Supervisão do jogador
  - » ... e outras técnicas variadas

# O que é anti-cheating

## » Ferramentas de terceiros

- » PunkBuster

- » Valve Anti-Cheat (VAC)

- » EasyAntiCheat 

## » Limitado apenas a monitoramento do software



# Muitos anos atrás...



# Ragnarök Online



## » Anti-cheat via GameGuard

- » Bloqueia macros de teclado e mouse por software
- » Bloqueia ferramentas de automação (eg. AutoHotkey)
- » Migrou para o EasyAntiCheat em 2019-06-18 (RIP VMs)

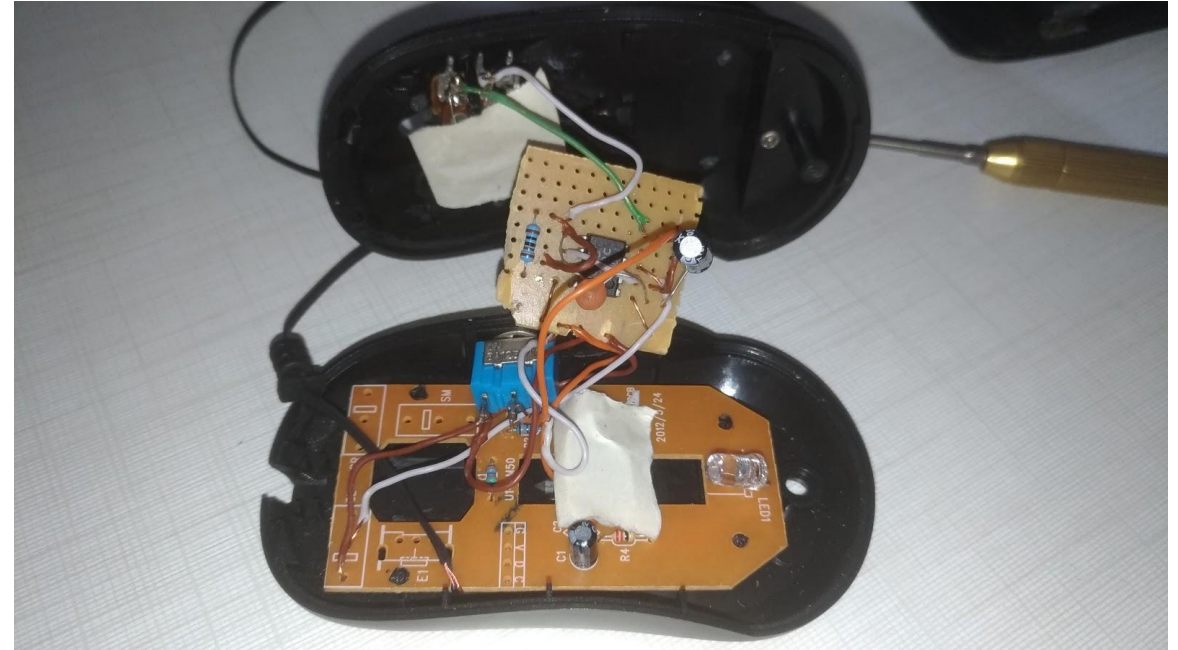
## » E como fica o hardware?

# The Auto-Clicker ©

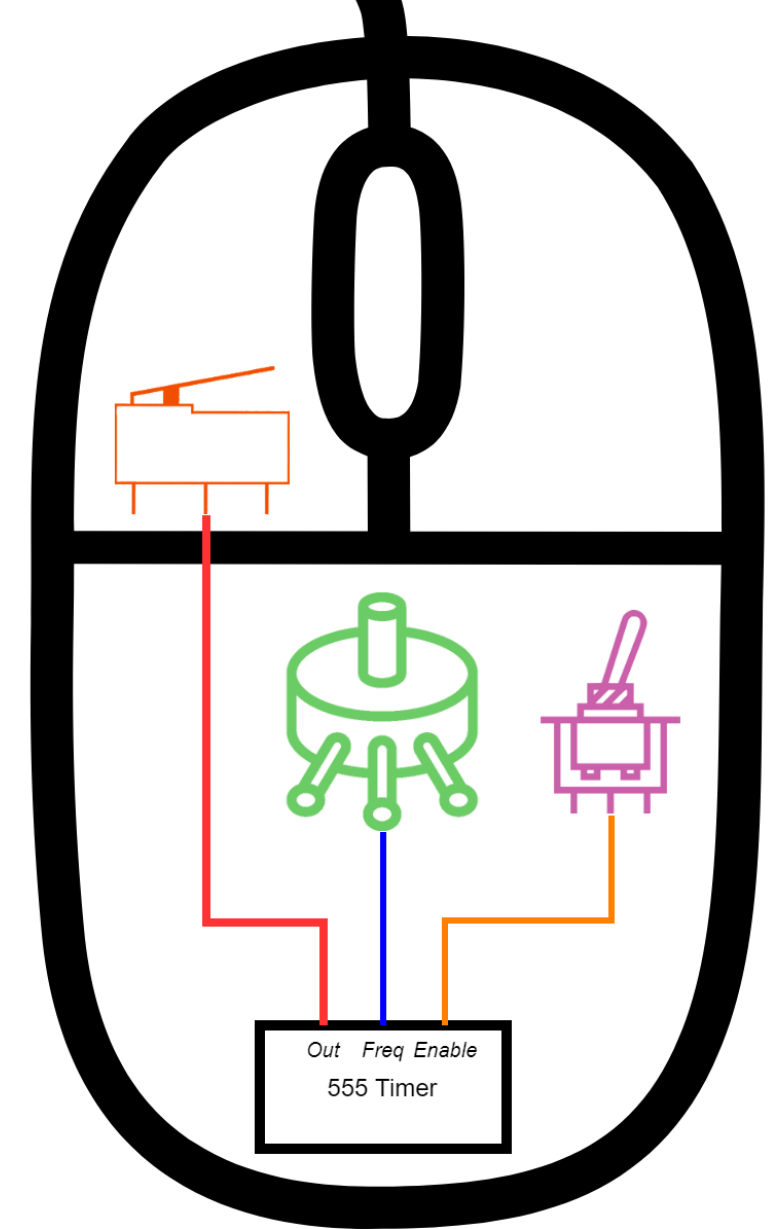
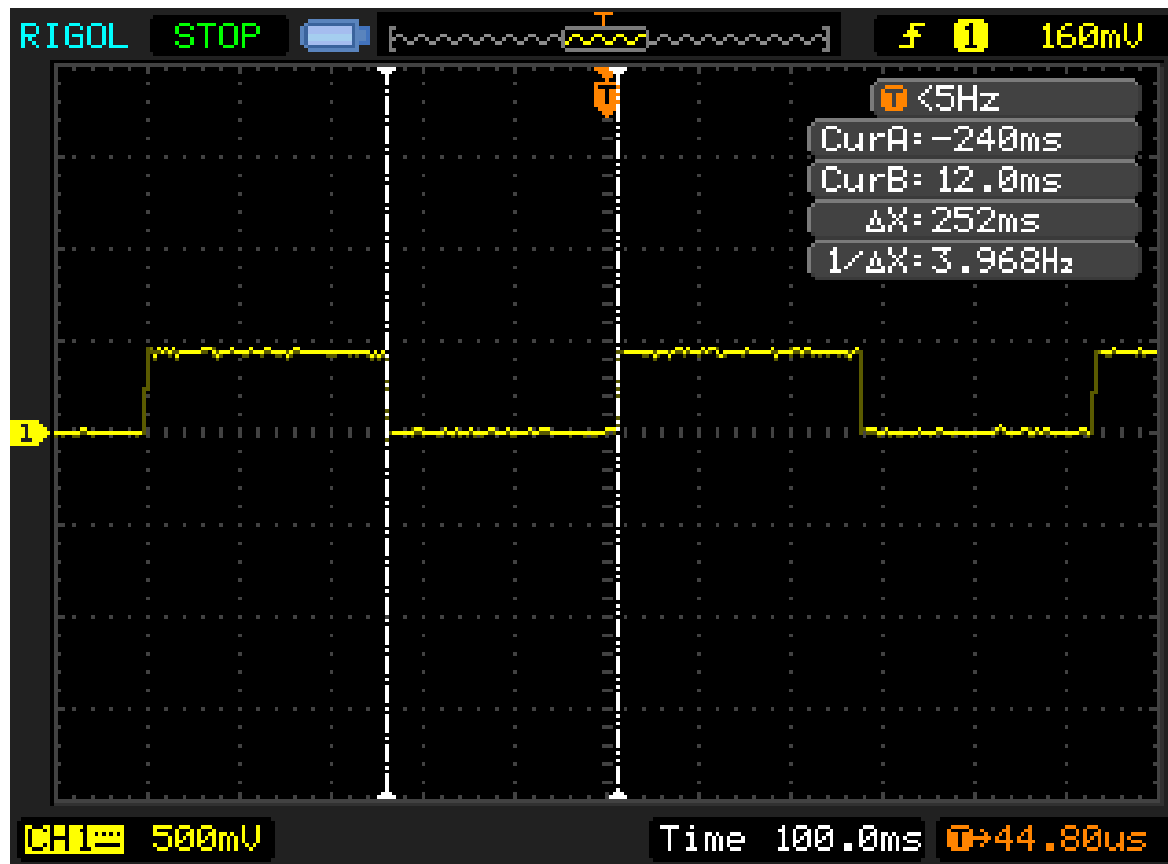
- » Botões são apenas (micro)switches!
- » Utiliza um 555 (timer) para fechar o circuito do botão
- » Frequência pode ser ajustada por um potenciômetro
- » Switch para ligar/desligar
- » RIP mouse



# The Auto-Clicker ©



# The Auto-Clicker ©





\$ ./demo

Em vídeo para garantir que vai dar certo 🍀

# But why?



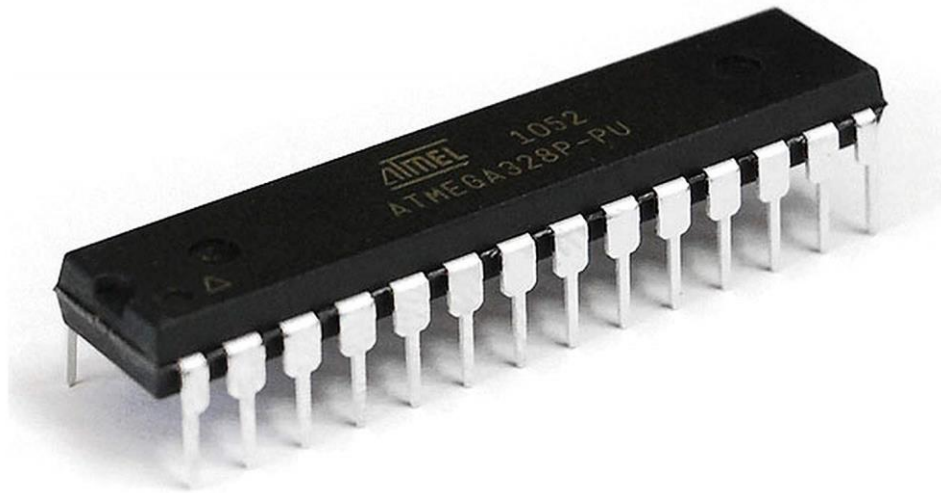
## » Vantagens no jogo

- » Clicar automaticamente em NPCs por (muitas) horas dava dinheiro 💰
- » Ferramenta anti-cheat não tinha (e ainda não tem) como detectar
  - » Foco é unicamente em hacks de software e não em nível de hardware

**Até onde o ~~jogo~~ sistema operacional sabe, é apenas um mouse!**

**Mas a gente *realmente*  
precisa de um *mouse*?**

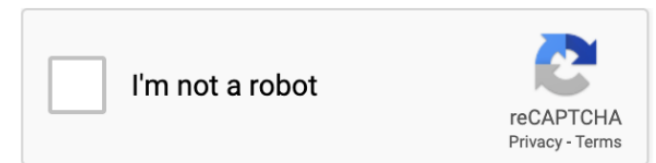
# Outros “mouses”





# Bug's Fake Mouse ©

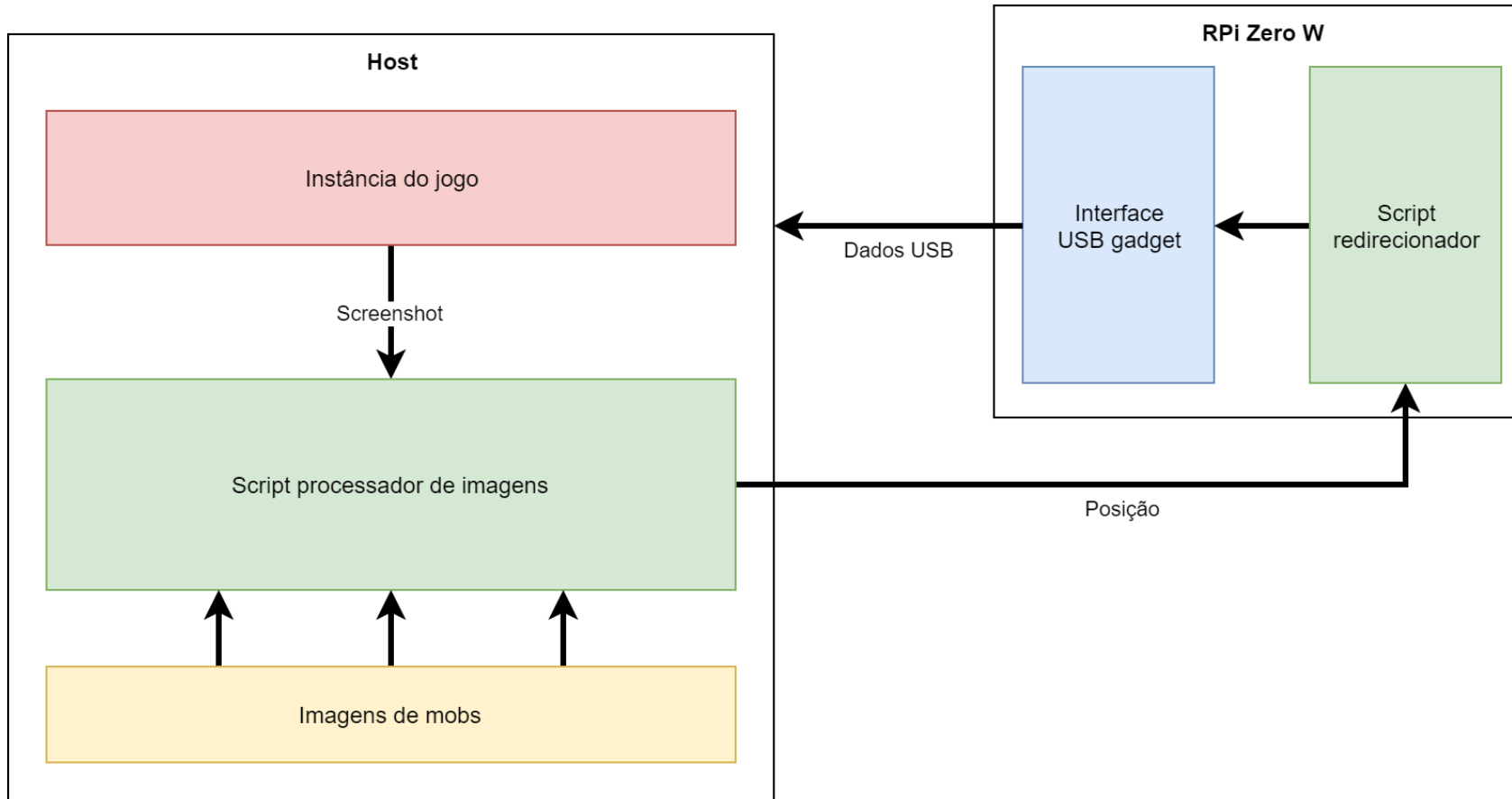
- » Baseado em uma Raspberry Pi Zero W
- » Modo USB gadget
  - » Permite tornar a RPi um dispositivo USB qualquer
  - » Script recebe dados pela rede e envia para a USB
- » Para o host é tudo apenas um mouse USB que se move *bem* rápido



# Bug's Fake Mouse ©



# Bug's Fake Mouse ©





\$ ./demo

Acharam que não ia ter demo, né?



# Como se proteger?

(se você for o dev, claro)

# Detecções nível I

## » Análise simples das entradas do jogo

- » Cliques com frequência ou duração constante
- » Teclados e mouses com entradas excessivamente rápidos

## » Soluções triviais:

- » Introdução de fator aleatório nas entradas
- » Controlar o *timing* das entradas baseado em comportamento real



# Detecções nível II

- » Dinâmica de digitação (*keystroke dynamics*)
  - » Biometria baseada em “como” digitamos
  - » Identificaria jogador a níveis individuais
- » Uso improvável ainda
  - » Biometria oscila muito durante o dia, além de fatores externos
  - » Complexidade alta para implementação em um simples jogo

# Detecções nível III

- » Comportamento anômalo do jogador
  - » Alterações nas horas e duração das partidas
  - » Anomalia de progresso (*level* alto)
- » Relativamente trivial de burlar
  - » Requer modelagem do comportamento real como base da automação

# Detecções nível IV

## » Controles dedicados (a la *consoles*)

- » Uso de técnicas anti-tampering para (tentar) impedir modificações
- » Uso de comunicação criptografada/autenticada para (tentar) impedir MITM

## » A não ser que teu jogo seja *muito* legal, eu não jogaria :)

- » Custo do hardware se tornaria alto
- » Base de jogadores ficaria limitada a quem pode comprar o hardware

# Detecções nível...V?

## » Combinação de técnicas

- » Técnicas tradicionais de detecção
- » Técnicas para detectar “anomalias” em hardware
- » Supervisão de jogo em caso de alta chance de trapaça



Tá, e agora?

(já tá acabando galera)

# Evoluindo o ataque



- » Processamento de imagens em FPGA
  - » Captura do sinal HDMI
  - » Alta performance, mas alta complexidade
- » E se o hardware fosse apenas software?
  - » Dispositivos podem ser virtuais (drivers!)
  - » Máquinas virtuais permitem qualquer dispositivo nelas




# Evoluindo ainda mais?

## » Streaming de jogos!

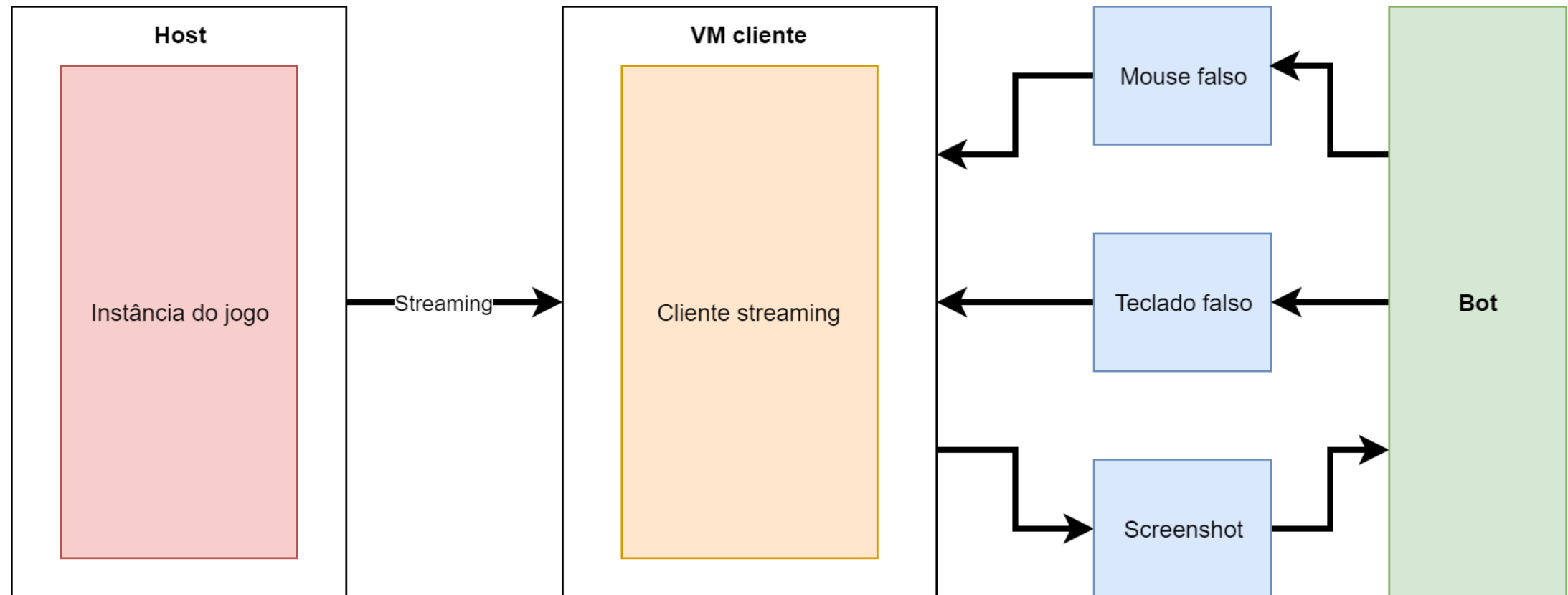
- » Steam In-Home Streaming
- » NVIDIA Shield

## » Simplificação da máquina virtual

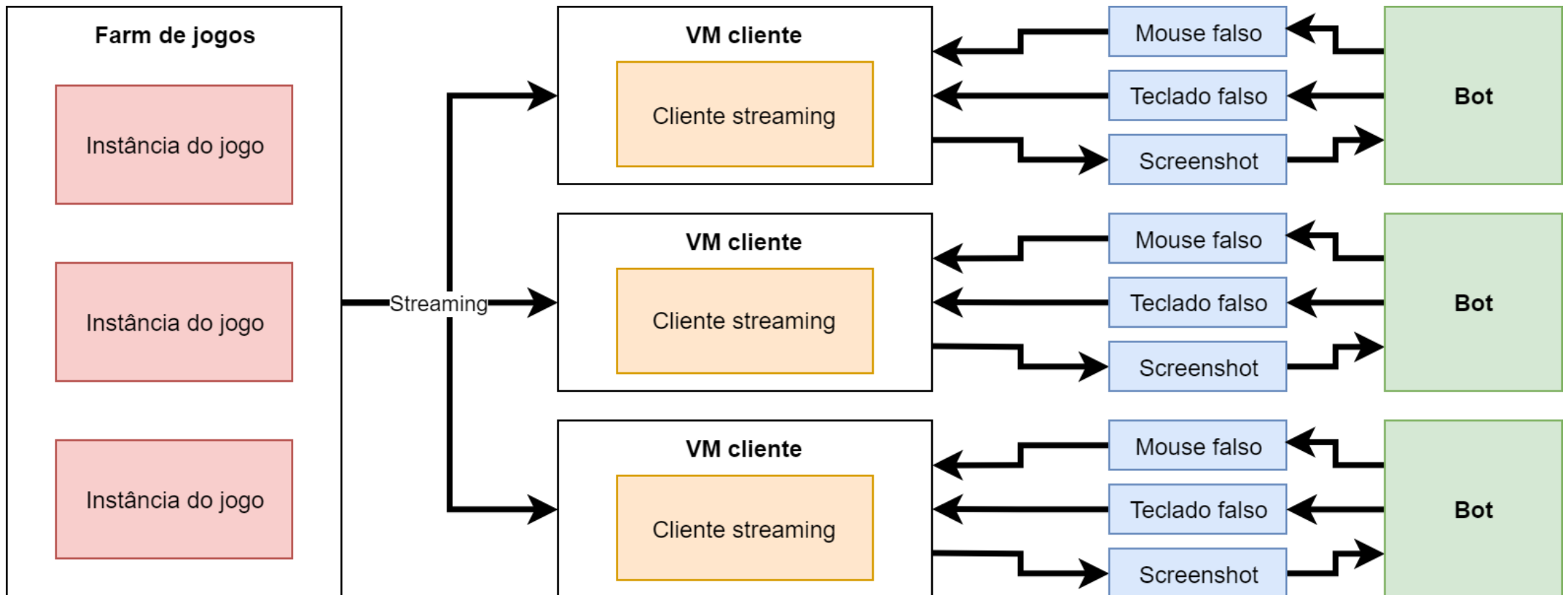
- » VM não precisa rodar o jogo e o ataque mantém sua simplicidade
- » Streaming assume dispositivos “locais” como reais 
- » Detecção de hardware modificado se torna mais complexo



# Proposta de ataque “remoto”

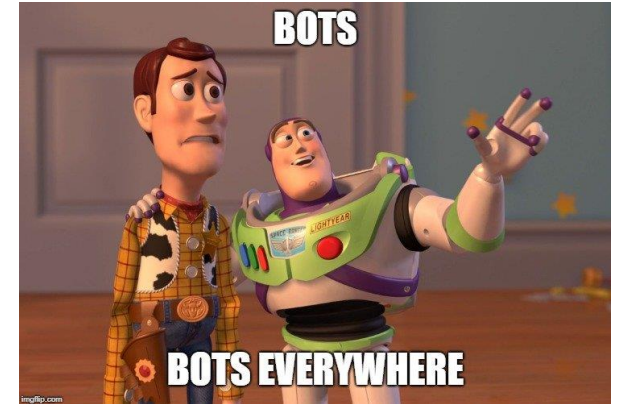


# Fazendinha Feliz



# Concluindo

- » Ataque não muito trivial de se proteger
  - » Sempre confiamos que nosso hardware nunca nos trairá
- » Necessário definir até *onde* queremos nos proteger
  - » Introdução de custos e complexidades adicionais
  - » Não dá pra se proteger 100% :)
- » Automação em geral
  - » Nada impede isto de ser expandido para um Selenium versão hardcore



# Pera, mas o quão eficiente é isso tudo?

- » Poucos meses de pesquisa e sabe o EasyAntiCheat que falei antes?
  - » 110 jogos “protegidos” (Fortnite inclusive)
  - » Não detectou nenhum dos mouses até agora



# DEFEATING ANTI-CHEAT WITH HARDWARE

Trapaceando em jogos sem ser (facilmente) detectado

Ricardo Gomes da Silva

DEFCON Porto Alegre 2019

