กลวิธีต่อต้านการโกงเกมออนไลน์

ONLINE GAMING ANTI-CHEAT METHODOLOGIES

วรนิษฐา ไกรสิทธิพงศ์

วิฒัฐ จิรฤกษ์มงคล

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2559

กลวิธีต่อต้านการโกงเกมออนไลน์

ONLINE GAMING ANTI-CHEAT METHODOLOGIES

วรนิษฐา ไกรสิทธิพงศ์

วิณัฐ จิรฤกษ์มงคล

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2559

ปริญญานิพนธ์ปีการศึกษา 2559

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง กลวิธีต่อต้านการโกงเกมออนไลน์

ONLINE GAMING ANTI-CHEAT METHODOLOGIES

ผู้จัดทำ

1. นางสาววรนิษฐา ไกรสิทธิพงศ์ รหัสนักศึกษา 56011055

2. นายวิณัฐ จิรฤกษ์มงคล รหัสนักศึกษา 56011127

..... อาจารย์ที่ปรึกษา

(ผู้ช่วยศาสตราจารย์อัครเดช วัชระภูพงษ์)

กลวิธีต่อต้านการโกงเกมออนไลน์

นางสาววรนิษฐา ไกรสิทธิพงศ์ 56011055
นายวิณัฐ จิรฤกษ์มงคล 56011127
ผู้ช่วยศาสตราจารย์อัครเดช วัชระภูพงษ์ อาจารย์ที่ปรึกษา
ปีการศึกษา 2559

บทคัดย่อ

การโกงเกมออนไลน์ เพื่อเอาเปรียบทั้งผู้เล่น และผู้ประกอบการที่ให้บริการเกมออนไลน์ ได้ส่งผลเสียต่อผลประโยชน์ที่บริษัทควรจะได้รับ โดยปัจจุบันยังไม่มีการรวบรวมกลวิธีต่าง ๆ เพื่อต่อต้านการโกงเกมออนไลน์ในรูปแบบต่าง ๆไว้ เพื่อเป็นแบบให้กับผู้พัฒนาเกม หรือผู้ประกอบการใช้เป็นแนวทาง โครงงานนี้จะประกอบไปด้วยยกกรณีศึกษาจากเกม และการใช้เครื่องมือในการโกง การทดลองจะใช้โปรแกรมช่วยเล่น จากสมมติฐานที่ว่า พฤติกรรมของตัวละครที่มีความสามารถมากกว่ามนุษย์ จะถูกระบบตรวจจับและลงโทษอัตโนมัติ โดยมีกลุ่มทดลองเลียนแบบผู้เล่น, มีพฤติกรรมที่ผิดปกติ, ไปจนถึงมีพฤติกรรมท้าทายการโดนระงับบัญชี จากนั้นทำการเก็บข้อมูลที่ได้จากการใช้โปรแกรมช่วยเล่นของเกมเพื่อนำมาวิเคราะห์ และผลลัพธ์จากข้อมูลดังกล่าวเป็นข้อสรุปเกี่ยวกับระบบการตรวจจับการโกงอัตโนมัติของเกม Pokémon Go ซึ่งขึ้นอยู่กับจำนวนครั้งการเข้าสู่ระบบ และจำนวน Pokémon ที่จับได้ในแต่ละวัน โดยมาตรการตรวจจับนั้นไม่ทรงประสิทธิภาพมากนัก แต่การอัพเดทเกมอย่างสม่ำเสมอ ทำให้การใช้งานโปรแกรมช่วยเล่นนั้นมีอุปสรรคมากขึ้น

Online Gaming Anti-Cheat Methodologies

Ms. Worranitta	Kraisittipong	56011055
Mr. Winut	Jiraruekmongkol	56011127
Asst.Prof. Akkradach	Watcharapupong	Advisor

Academic Year 2016

ABSTRACT

Cheating for advantages over players or providers causes damages directly to providers. Damages including lost benefits, and worse player society. In present, there are no collected methodologies to counter cheating as a guide to provide checklists of methods to prevent cheating for online game providers or game maker. This project is meant to be a guide for all people with tested methodologies. Pokémon Go is chosen as a case study for an experiment we conducted. The assumption is when a character or an avatar behaved unlike human or player, the cheat detector implemented should detect and act properly. We categorized our bot configuration into three; bots behaved like players, bots behaved like abnormal players, and bots behaved unlike players at all. While bots are running, we collect logs from the program and afterward analyzed them. The results showing that Pokémon Go's cheat detecting system uses number of logins and number of Pokémon caught within 24 hour. Although the detecting methodologies are not so effective but constantly update client software create obstacles for cheaters.

กิตติกรรมประกาศ

โครงงาน และปริญญานิพนธ์ฉบับนี้เสร็จสมบูรณ์ได้ เนื่องจากได้รับคำแนะนำ และคำปรึกษาที่ดีอย่างต่อเนื่องจากท่านอาจารย์ที่ปรึกษา ผู้ช่วยศาสตราจารย์อัครเดช วัชระภูพงษ์ ที่คอยให้คำแนะนำและชี้แจงอย่างเสมอ ทั้งในเวลา และนอกเวลา รวมไปถึงการแนะนำ ชักจูงเพื่อให้โครงงานนี้เกิดขึ้น และบรรลุผลได้ อย่างที่คาดหวังไว้ ทางคณะผู้จัดทำขอขอบคุณอาจารย์ที่ปรึกษาเป็นอย่างสูง

ขอขอบคุณกลุ่มชุมชนบน Github และทีม PokemonGoF ที่ร่วมกันสร้างโปรแกรมช่วยเล่น PokemonGo-Bot ขึ้นมา เพื่อที่จะใช้ในการทดลอง

ขอขอบคุณกลุ่มชุมชน PokemonGoDEV บน www.reddit.com ที่รวมตัวกันสร้างโปรแกรมช่วยเล่น, แคร็ก, และสร้าง pgoapi สำหรับติดต่อกับทาง Pokemon Go servers

ขอขอบคุณห้องวิจัย ISAG ที่เอื้อเฟื้ออุปกรณ์ สำหรับการทดลองตลอด 24 ชั่วโมง

ขอขอบคุณคณะเทคโนโลยีสารสนเทศ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ สำหรับโอกาสในงานการประชุม National Conference of Computer and Information Technology ครั้งที่ 13

วรนิษฐา ไกรสิทธิพงศ์

วิณัฐ จิรฤกษ์มงคล

สารบัญ

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญตาราง

ตาราง **หน้า**

สารบัญรูป

สารบัญรูป (ต่อ)

สารบัญรูป (ต่อ)

บทที่ 1
บทนำ

1.1 ความเป็นมาของปัญหา

โลกดิจิตอลทุกวันนี้มีบทบาทมากขึ้นกับชีวิตประจำวันของผู้คน จากอดีตที่เกมเป็นแค่เครื่องมือเพื่อความบันเทิง จนกระทั่งอินเทอร์เน็ตก่อให้เกิด "เกมออนไลน์" เกิดเป็นอีกสังคม ซึ่งแยกจากโลกความจริงอย่างสิ้นเชิง ในช่วงยุคต้นของเกมออนไลน์ ผู้ประกอบการเปิดให้บริการเกมออนไลน์นั้นมีรายได้หลักจากการจ่ายเงินเพื่อเล่นเกม หรือรู้จักกันในชื่อ "Air time" ตัวอย่างเกมที่ใช้ระบบนี้เช่น Ragnarok Online โดยภายหลังการเข้ามาของเกมประเภท Free to play ทำให้เกมออนไลน์ต่าง ๆที่ใช้ระบบ air time มีรายได้ลดลง ผู้ประกอบการต่าง ๆ จึงมีความสนใจในรูปแบบ free to play และเปลี่ยนช่องทางรายได้จาก Air time เป็นระบบ Item mall หรือ Cash Shop กล่าวโดยง่าย ผู้เล่นที่ทำการ "เติมเงิน" จำได้รับเงินดิจิตอลในระบบของเกมนั้น ๆ เพื่อนำไปใช้แลกไอเทม หรือสิทธิพิเศษ เหนือผู้เล่นที่ไม่ได้ทำการเติมเงิน (Free to play; But pay to win) ในส่วนของผู้ไม่ประสงค์ดี หรือผู้ที่ต้องการความได้เปรียบเหนือผู้เล่นอื่น โดยไม่ต้องเติมเงิน ได้หาช่องทาง ในการใช้วิธีทางต่าง ๆเพื่อให้ได้มาซึ่งความได้เปรียบเหนือผู้เล่นอื่น เป็นผลเสียต่อผู้ประกอบการอย่างมาก ทั้งทางตรงและทางอ้อม; สูญเสียรายได้ที่ควรจะได้รับจากการเติมเงิน และสังคมในเกมที่ไม่สมดุล ก่อให้เกิดผลเสียอื่นต่อไป

1.2 วัตถุประสงค์ของโครงงาน

1. เพื่อศึกษากลวิธีป้องกันการโกงเกมออนไลน์
2. เพื่อรวมรวบกลวิธีที่ใช้ในการป้องกันเกมออนไลน์จากผู้ให้บริการต่าง ๆ
3. เพื่อศึกษาหลักการทำงานของโปรแกรม และการตรวจจับการโกงเกมออนไลน์

1.3 ประโยชน์ที่คาดว่าจะได้รับ

องค์ความรู้การต่อต้าน การโกงเกมออนไลน์ เบื้องต้น หรือที่มีประสิทธิภาพ เพื่อปกป้องผลประโยชน์ของผู้ประกอบการ

1.4 ขอบเขตของโครงงาน

ศึกษาผลกระทบการโกงเกม ที่ทำให้ผู้ประกอบการสูญเสียรายได้ ที่ควรจะได้รับ

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

2.1 Cheat software

การโกง (Cheat) คือการใช้กลวิธีใดก็ตามเพื่อเพิ่มเติมผลประโยชน์ให้แก่ผู้เล่นไม่ทางใดก็ทางหนึ่ง ซึ่งบางครั้งผู้พัฒนาเกม (Game developer) สามารถใช้ได้ในเกมที่มีผู้เล่นคนเดียว (Single player) แต่ไม่ได้รับอนุญาตสำหรับเกมที่มีผู้เล่นหลายคน (Multiplayer games)

Cheat software ที่จะกล่าวถึงต่อไปนี้เป็นการปรับแต่งสภาพแวดล้อมของเกมซึ่งเอื้อประโยชน์แก่ผู้เล่นให้มากกว่าผู้เล่นคนอื่น ๆ ซึ่งผลประโยชน์เหล่านั้นจะมีรูปแบบต่าง ๆเช่น

- แสดงข้อมูลบางส่วนที่ผู้เล่นไม่ควรทราบ
- ปรับแต่งเกมเพื่อให้ผู้เล่นสามารถกระทำบางสิ่ง (action) ที่ปกติไม่ได้รับอนุญาต
- การเลียนแบบพฤติกรรมที่กำหนด โดยอาศัยการจดจำการเคลื่อนไหวของเมาส์และคีย์บอร์ด

2.1.1 Cheat software in Windows operating system

การทำงานของ Cheat software หลักการการเข้าถึงหน่วยความจำ (Memory) จะคล้ายกับพวก Malware ในการใช้ Windows API สำหรับ Debugger เพื่อเข้าถึงส่วนของความจำ (Address space) ของเกมที่ต้องการโกง

โดย Cheat software ใน Windows operating system แบ่งเป็น

External vs Internal cheats การโกงประเภทนี้ส่วนมากจะสร้าง Dynamic-link library (DLL) และบังคับให้เกมโหลด DLL นั้น โดยการโกงที่สามารถเข้าถึงโครงสร้างของข้อมูล (Data structure) ปรับปรุงแก้ไข และเรียกใช้ Function ของเกมได้โดยตรง จะเรียกว่า Internal cheat และ External cheat จะทำการ run แยกกับตัวเกมและมีส่วนของความจำ (address space) ของตนเอง และใช้การ Remote ผ่าน API function ของ Windows

User mode vs Kernel mode ผู้พัฒนาโปรแกรมโกงจะอาศัยประโยชน์จากโค้ดที่รันโดยใช้ Kernel mode ซึ่งทำให้สามารถเข้าถึงหน่วยความจำทั้งหมดและสามารถปรับเปลี่ยนพฤติกรรมของ Windows API functions และป้องกันการตรวจจับการโกงจาก User mode

2.1.2 Cheat software tools

การทำงานของเกมทั่วไป ผู้เล่นจะไม่สามารถเข้าถึงโค้ดได้โดยตรง ทำให้ผู้พัฒนาโปรแกรมโกงต้องใช้วิศวกรรมผันกลับ(reverse engineer) เพื่อเข้าถึง Location และ Structure ของข้อมูลแต่ละตัว โดย tools ที่พบได้ทั่วไป มีดังนี้

Cheat Engine เป็น open-source tool ที่มีจุดประสงค์เพื่อเปลี่ยนแปลงพฤติกรรมของเกม โดยสามารถเปลี่ยนค่าตัวแปรและโครงสร้างภายในของ game memory เช่น ค่าเลือด, ความเร็ว เป็นต้น ซึ่งถูก

ออกแบบมาสำหรับ Single player game แต่มีการดัดแปลงเพื่อใช้ใน online game ยกตัวอย่างเช่น เกม Team Fortress2 ซึ่งแก้ไขเพียง 1 byte ก็สามารถเข้าสู่ developer mode และมองเห็นผู้เล่นอื่นทะลุกำแพง

รูปที่ 2.1.1 ตัวอย่างการใช้ Memory patching

โดยการทำงานของโปรแกรมอาศัยหลัก Memory scanner, Memory viewer และ debugger เพื่อแก้ไขหน่วยความจำของเกม และ debugger จะตรวจสอบ function ที่เข้าถึงเกมนั้น ๆ

รูปที่ 2.1.2 โปรแกรม Cheat engine

โดย Cheat engine สามารถใช้เป็น DLL Injector ได้ ตัวอย่างเช่น ใช้ในการโกงความเร็ว (Speed hack) ซึ่งทำให้ผู้เล่นสามารถเคลื่อนไหวในเกมได้รวดเร็วมากขึ้น

IDA (Interactive Disassembler) เป็น Disassembler และ Debugger ซึ่งสามารถใช้ได้ทั้ง Static และ Dynamic โดยเป็นโปรแกรมเป็นเชิงพาณิชย์ ส่วน Disassembler ของ IDA จะสามารถสร้างการแสดงกราฟิกฟังก์ชั่นในภาษาแอสแซมบลีจากคำสั่งที่เก็บไว้ภายในไบนารีไฟล์ โดยผู้ใช้จะสามารถเปลี่ยนชื่อฟังก์ชันและตัวแปรเพื่อทำให้อ่านง่ายมากขึ้นได้ ซึ่งสามารถสร้างโค้ดที่คล้ายกับภาษาซีได้ ผู้ใช้สามารถกำหนด Breakpoint ที่ Assemble code ได้ ซึ่ง IDA ทำให้การทำวิศวกรรมผันกลับ (Reverse engineering) สามารถทำได้สะดวกมากขึ้นและทำให้ผู้พัฒนาโปรแกรมโกงใช้เพื่อ Reverse engineer Game และสร้างโปรแกรมกันโกงเกมต่าง ๆ

"ReClass" เป็นเครื่องมือแบบ Open source เพื่อทำวิศวกรรมผันกลับ (Reverse engineering) สำหรับข้อมูลที่ไม่ทราบโครงสร้าง (Unknown Data Structures) และ Class ในหน่วยความจำของเกม ซึ่งเมื่อผู้ใช้ทำการ "ผนวก" โปรแกรมเข้ากับโปรเซส และป้อนตำแหน่งที่อยู่ของโครงสร้างของข้อมูลเจอแล้ว เขาสามารถเข้าถึงสมาชิกของโครงสร้างนั้น ๆ ได้

รูปที่ 2.1.3 โครงสร้างข้อมูลของเกม

โปรแกรมช่วยให้ผู้ใช้ที่จะขยายขนาดของโครงสร้างข้อมูลที่ไม่รู้จัก (Unknown Data Structures) เปลี่ยนประเภทของสมาชิกในโครงสร้างจนแสดงให้เห็นถึงโครงสร้างข้อมูลที่เหมาะสม จะมีการอธิบายโครงสร้างเป็นโครงสร้างข้อมูลที่ใกล้เคียงกับภาษาซี (C-like structure)

2.1.3 การดำเนินการทางด้านกฎหมาย

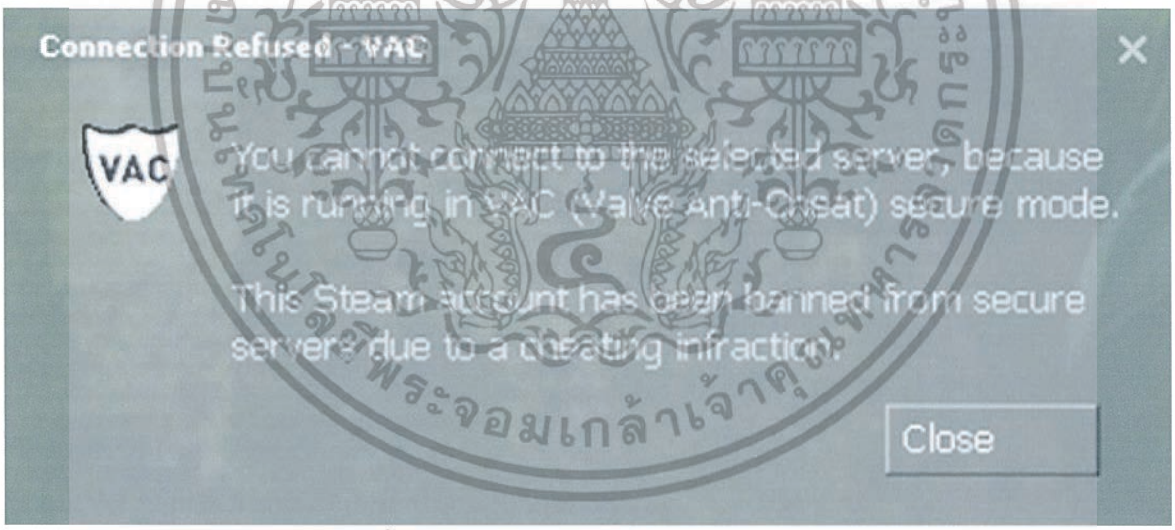
โดยทั่วไปการดำเนินการทางกฎหมายเพื่อปกป้องเกมจากผู้พัฒนาโปรแกรมโกงโดยผ่านทางข้อตกลงอนุญาตให้ใช้สิทธิของผู้ใช้ (EULA : End User License Agreement) เพื่อลงโทษผู้เล่นโดยการจำกัดสิทธิ์ของผู้เล่นในเกม (Ban)

บริษัทบลิซซาร์ดเอ็นเตอร์เทนเมนต์ (Blizzard Entertainment, Inc.) เป็นบริษัทที่มีชื่อเสียงในการใช้กฎหมายเพื่อเผชิญหน้าและกำจัดโปรแกรมโกงเกม ในปี2008 บริษัทบลิซซาร์ดได้ทำการดำเนินการทางกฎหมายกับ MDY Industries ซึ่งทำการจำหน่ายโปรแกรมโกงเกม World of Warcraft (WoW) ซึ่งเป็นเกมที่เป็นที่นิยมของบลิซซาร์ด ศาลสรุปว่าโดยการให้โกง MDY เป็นความผิดของ การแทรกแซงละเมิดสิทธ์ และช่วยในการดึงดูดผู้เล่นได้เปรียบที่ไม่เป็นธรรมมากกว่าผู้เล่น WoW คนอื่น ๆ และหลีกเลี่ยงการตรวจสอบโดยบลิซซาร์ด อีกกรณีหนึ่งเมื่อปี 2013 โดยบลิซซาร์ดได้ทำการฟ้องร้องและชนะบริษัท Ceiling Fan Software LLC(CF) ที่พัฒนาและขายโปรแกรมโกงเกมของโดยบลิซซาร์ด

ปัญหาที่แท้จริงคือการต่อสู้กับโปรแกรมโกงผ่านกฎหมายซึ่งแตกต่างกันไปในแต่ละประเทศ และผู้พัฒนาโปรแกรมโกงปิดตัวโปรแกรมลงเพื่อป้องกันการถูกตรวจสอบ และเปิดใหม่อีกครั้ง ทำให้ต้องมีการต่อสู้ทางกฎหมายไม่สิ้นสุด

2.2 การแบน (Banned)

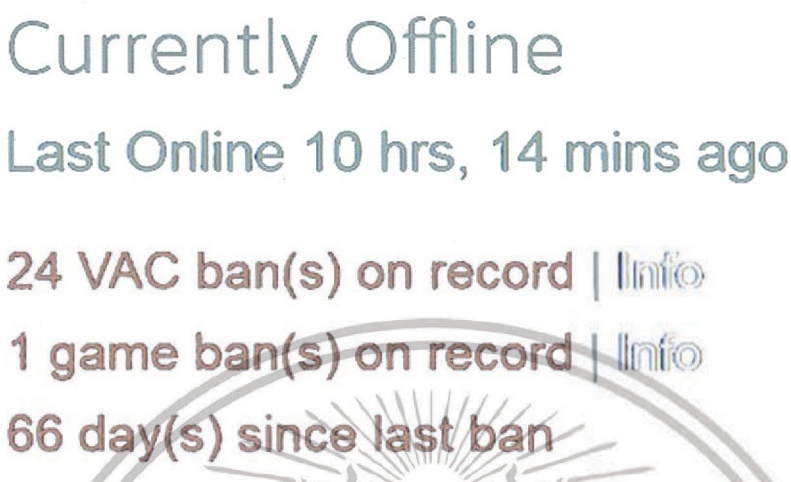
2.2.1 ระบบแบน VAC (Valve Anti-Cheat System)



รูปที่ 2.4 ตัวอย่างของการโดนแบนจาก VAC

Valve Anti-Cheat System หรือ VAC เป็นระบบแบนอัตโนมัติของ Stream ซึ่งมีไว้เพื่อป้องกันการโกงเกมจากโปรแกรมช่วยเล่น หรือโปรแกรมโกงเกม โดยที่จะตรวจสอบเฉพาะเกมที่ใช้งาน VAC เท่านั้น โดยจะทำการแบนทันทีที่ตรวจพบ แต่จะมีการตรวจสอบเพื่อความแน่ใจ โดยจะใช้เวลาประมาณ 1-3 สัปดาห์หลังจากที่ตรวจพบ และผู้เล่นจะไม่สามารถใช้งานเกมบนเซิร์ฟเวอร์ที่ใช้งานระบบ VAC ได้อีก ซึ่งบางเกมจะทำการแบบเฉพาะเกมนั้น ๆ แต่บางเกมจะทำการแบนหลายเกมด้วยกัน

โดยการแบนจะมีแสดงอยู่บนหน้าโปรไฟล์ของ Stream ด้วยเช่นกัน



รูปที่ 2.5 หน้าโปรไฟล์ที่ถูก VAC ban

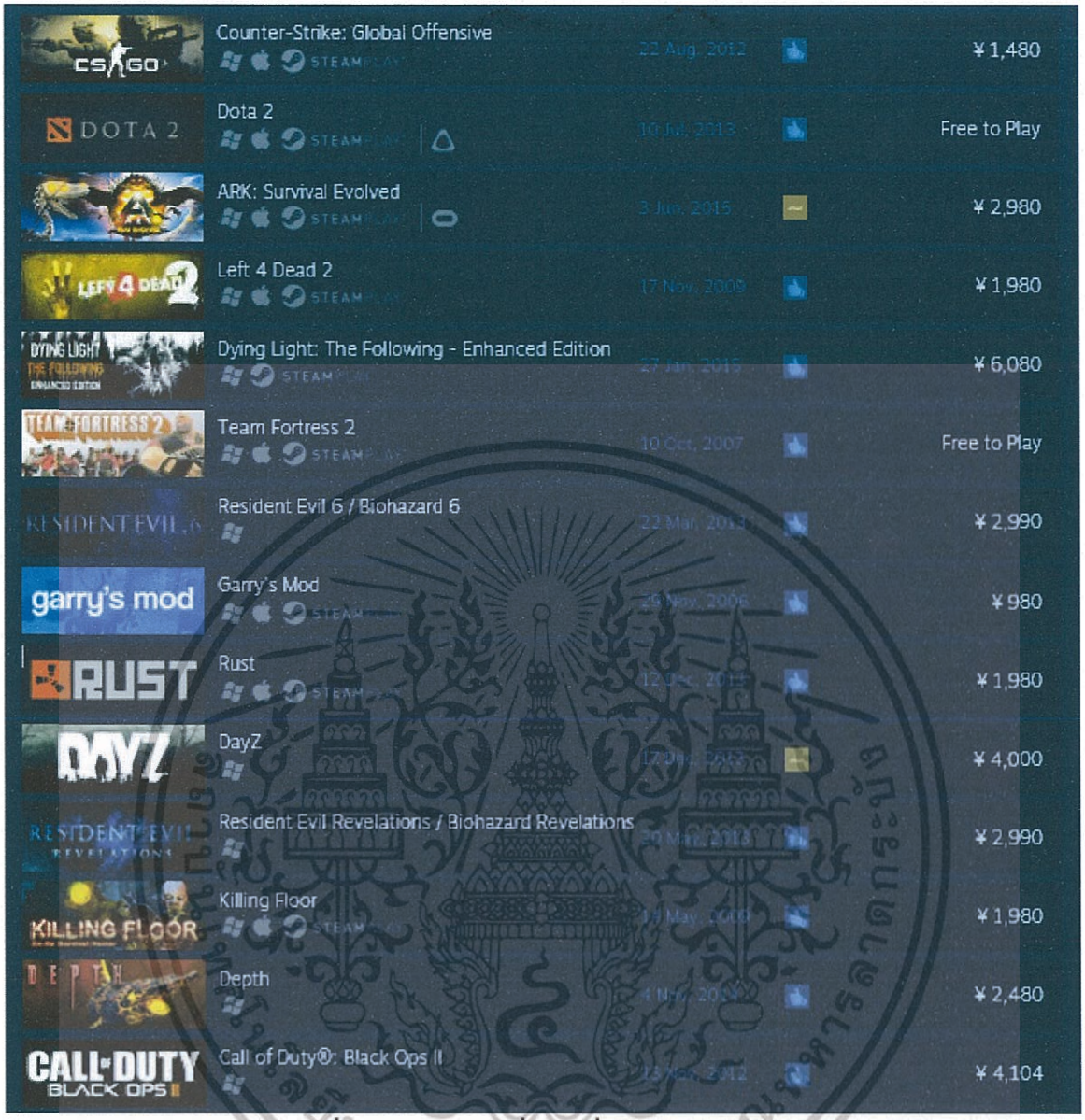
ซึ่ง VAC จะทำการแบนแบบถาวร ผู้เล่นจะไม่สามารถย้าย Item หรือเกมไปแอคเคาท์อื่น ๆได้ รวมถึงแอคเคาท์ที่ใช้เบอร์โทรศัพท์เดียวกันก็จะถูกแบนตามไปด้วย ซึ่งผู้เล่นจะยังสามารถเล่น Singer-player-game และ Multiplayer game ที่ไม่มี VAC ได้

รูปที่ 2.6 โปรแกรม Cheat Engine

การใช้งานโปรแกรมโกงเกม โดยเฉพาะโปรแกรมที่เข้าถึงหน่วยความจำของคอมพิวเตอร์ เช่น Cheat Engine จะทำให้มีโอกาสสูงมากที่จะถูกแบน เพราะฉะนั้นเพื่อความปลอดภัยจึงไม่ควรติดตั้งและใช้งานโปรแกรมเหล่านี้ขณะใช้งาน Stream เพราะบางครั้งโปรแกรมอาจจะทำงานขึ้นมาเองโดยไม่ตั้งใจ หรือเครื่องที่ใช้งานรันโปรแกรมเหล่านี้อยู่ ก็อาจส่งผลให้ถูกแบนได้

รูปที่ 2.7 โปรแกรมแอนตี้ไวรัส Avast

นอกจากนี้ การติดไวรัส (Virus) หรือมัลแวร์ (Malware) ก็มีผลต่อโอกาสการโดนแบนเช่นกัน รวมไปถึงโปรแกรมแอนตี้ไวรัส(Antivirus) บางตัว ถ้าไม่ตั้งค่าให้ดี ก็อาจส่งผลให้ถูกแบนได้ เพราะโปรแกรมแอนตี้ไวรัสอาจมองระบบป้องกันการโกงเกมเป็นไวรัส เนื่องจากลักษณะการทำงานของระบบป้องกันใกล้เคียงกับไวรัส กรณีศึกษาที่เคยเกิดขึ้นเช่น มีผู้เล่นเคยถูกแบนเนื่องจากโปรแกรมแอนตี้ไวรัส Avast ไปบล็อกการทำงานของ VAC ส่งผลให้ผู้เล่นถูกแบนจาก Stream นอกจากนี้การใช้งาน Mod(Modification) ก็อาจส่งผลให้ผู้เล่นถูกแบนได้เช่นกัน เพราะ Mod ที่ติดตั้งลงไป อาจมีการทำงานที่เข้าข่ายการโกงหรือดัดแปลงตัวเกม ซึ่งเกมที่มีผู้เล่นคนเดียว (Single player) แต่มีการใช้ VAC ก็สามารถถูกแบนเช่นกัน แต่จะมีบางกรณีที่เป็นการตรวจสอบที่ผิดพลาดของระบบ VAC ซึ่งมีโอกาสน้อยมาก ซึ่งสามารถติดต่อ Stream Support ได้ แต่อาจจะต้องส่งข้อมูลและหลักฐานเพื่อยืนยัน



รูปที่ 2.8 ตัวอย่างรายชื่อเกมที่ใช้งานระบบ VAC

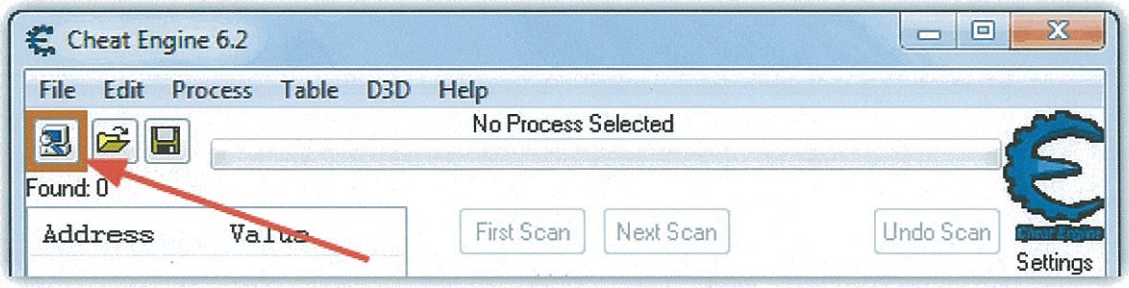
รายชื่อเกมทั้งหมดที่ใช้งานระบบ VAC นั้นสามารถตรวจสอบได้จากตัวเว็ปของทาง Stream

2.3 Cheat Engine

2.3.1 การใช้ Cheat engine กับ Flash game บน Web browser ต่าง ๆ

การใช้ Cheat engine กับ Flash game บน Web browser ต่าง ๆ มีวิธีการดังนี้

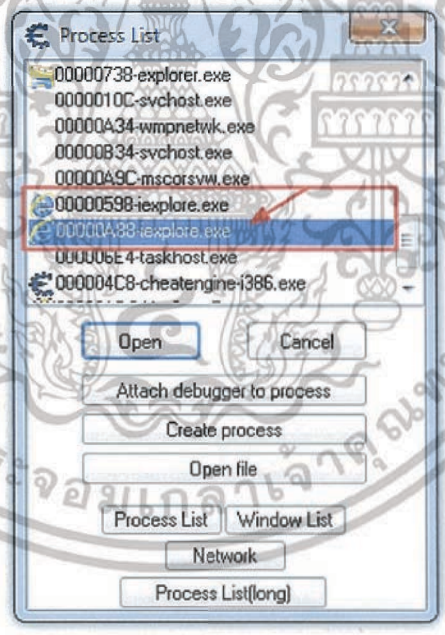
1. ดาวน์โหลดและติดตั้งโปรแกรม Cheat Engine
2. เปิดเกม Flash บน Web browser
3. เลือก Open process เพื่อเลือก Process ที่จะใช้



รูปที่ 2.9 ปุ่ม Open process

Internet Explorer

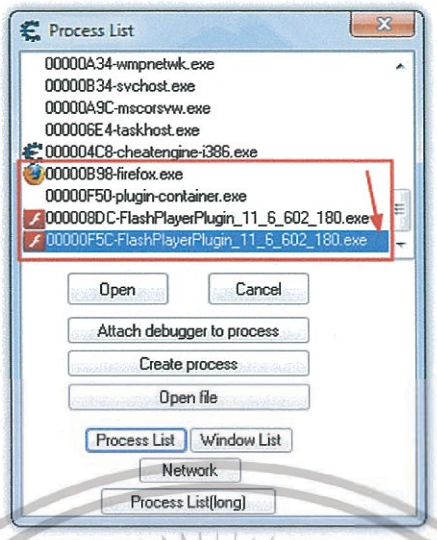
Internet explorer มักจะสร้าง 2 หรือ 3 Process ให้ทำการเลือก Process ที่มีค่าน้อยที่สุด ในกรณีที่ไม่สำเร็จ ให้เลือกค่าที่น้อยรองลงมา



รูปที่ 2.10 Process list ของ Internet explorer

Mozilla Firefox

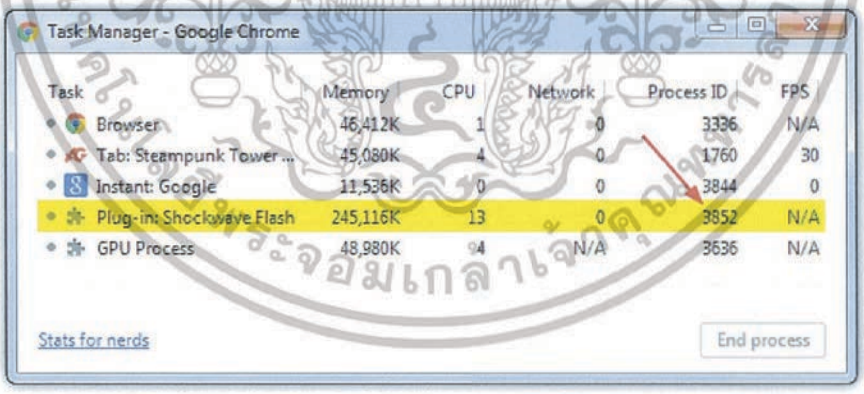
Firefox ใช้ Flash player จาก Plugin ภายนอก และจะมีรายชื่อ Flash 2 processes โดยประมาณ ให้ทำการเลือก Process ที่มีค่าน้อยที่สุด



รูปที่ 2.11 Process list ของ Flash ของ Firefox

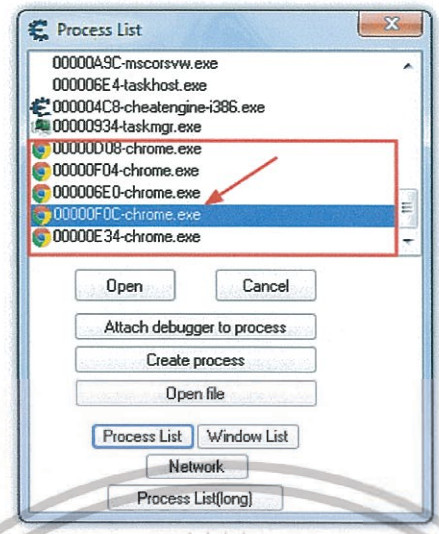
Google Chrome

เนื่องจาก Google Chrome มีหลาย Processes ทำให้ค่อนข้างยากที่จะหาในโปรแกรม Cheat Engine เพราะฉะนั้นให้ทำการเปิด Task manager ของ Chrome โดยการกด Shift และ Escape เพื่อทำการหา Process ID



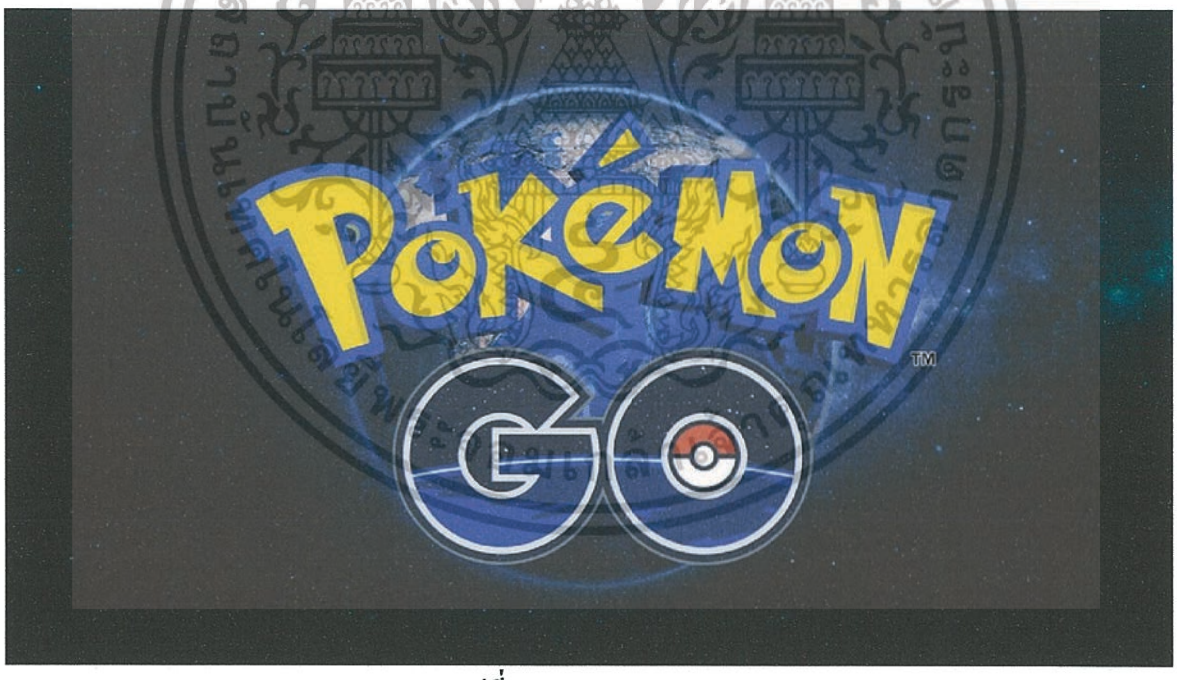
รูปที่ 2.12 Task manager ของ Google Chrome

จากนั้นให้ทำการแปลง Process ID เป็นเลขฐาน 16 และเปิด Open process ของ Cheat Engine และเลือก Process ที่มีหมายเลขนั้น ๆ



รูปที่ 2.13 Process List ของ Google chrome

2.4 โปเกมอน โก



รูปที่ 2.14 Pokémon Go

โปเกมอน (Pokémon) ถูกสร้างขึ้นในปี 1995 โดยซาโตชิ ทาจิริ นักพัฒนาชาวญี่ปุ่น โดยเริ่มต้นซาโตชิเป็นนักสะสมแมลงตั้งแต่เด็ก และโตมากับแนวคิดเรื่อง กระเป๋ามอนสเตอร์ (Pocket Monster) ซึ่งต่อมาถูกพัฒนามาเป็นโปเกมอน โดยเกมโปเกมอนเริ่มต้นด้วยวีดีโอเกมซึ่งบริษัทของซาโตชิเป็น

ผู้พัฒนา และมีนินเทนโด(Nintendo) เป็นผู้เผยแพร่ โดยโปเกมอนประสบความสำเร็จอย่างมากและทำให้นินเทนโดขึ้นครองตลาดในช่วงปลายทศวรรษ 90

ต่อมา ภายใน นีแอนทิค แลป (Niantic labs) นีแอคทิคเป็นบริษัทซึ่งถูกสร้างขึ้นมาภายในกูเกิล (Google) ซึ่งนีแอนทิคได้ให้ความสำคัญเกี่ยวกับ เกมมือถือแบบการรวมสภาพแวดล้อมจริง กับ วัตถุเสมือนเข้าด้วยกัน หรือเรียกว่า เออาร์(Augmented reality) โดยนีแอนทิคได้สร้างเกมแรกขึ้นมาคือ Ingress ซึ่งเผยแพร่เมื่อปี 2012 และประสบความสำเร็จอย่างมาก หลังจากนั้น นีแอนทิคก็แยกตัวออกจากกูเกิล ต่อมาในเดือนกันยายน 2015 นีแอคทิคประกาศว่าได้ร่วมพัฒนาเกมโปเกมอนโก (Pokémon Go) ร่วมกับนินเทนโด และบริษัทโปเกมอน โดยบริษัทโปเกมอนเป็นเพียงกิจการที่ถูกสร้างขึ้นมาเพื่อใช้จัดการใบอนุญาตของตัวละครโปเกมอน

โปเกมอนโก (Pokémon Go) เป็นเกมออนไลน์ประเภทที่อาศัยตำแหน่งทางภูมิศาสตร์ (Geolocation based) โดยเป็นเกมออนไลน์บนมือถือที่มีการรวมสภาพแวดล้อมจริง กับวัตถุเสมือนเข้าด้วยกัน หรือเรียกว่า เออาร์ (Augmented reality) ผู้เล่นต้องใช้มือถือสมาร์ทโฟนเพื่อใช้ในการออกค้นหาโปเกมอนจากสถานที่ในโลกจริง โดยการใช้กล้องมือถือในการส่องหาโปเกมอนในที่ต่าง ๆ และค่อยใช้มือถือในการจับโปเกมอน ฝึกฝนโปเกมอน แลกเปลี่ยน และต่อสู้กันกับผู้เล่นคนอื่น ๆ

จุดเด่นของโปเกมอน โก ที่ทำให้ได้รับความนิยมก็คือ การให้ผู้เล่นเดินทางไปตามสถานที่ต่าง ๆ ในโลกจริง เพื่อทำการ จับ โปเกมอน โดยประเภทของโปเกมอน ก็จะแตกต่างไปตามสถานที่จริง เช่น ถ้าเป็นโปเกมอนสายพันธุ์น้ำ ก็ต้องไปหาจับตามที่ใกล้แหล่งน้ำ หรือหากเป็นโปเกมอนที่อาศัยอยู่ตามทุ่งหญ้า ผู้เล่นก็จะต้องไปตามจับในที่รกร้างหรือที่กว้าง และการฟักไข่โปเกมอน ผู้เล่นจะต้องทำการเดินหรือวิ่ง เพื่อเร่งการฟักไข่ด้วย

2.5 การโกงเกมออนไลน์และความปลอดภัย

เนื่องจากการพัฒนาของมัลติมีเดียภาพเคลื่อนไหวและ Network bandwidth เกมออนไลน์ได้กลายเป็นอุตสาหกรรมที่ประสบความสำเร็จมากและมีความโดดเด่น โดยเฉพาะอย่างยิ่งในภูมิภาคเอเชียแปซิฟิก แต่เนื่องจากการขาดการพิจารณาการรักษาความปลอดภัยการควบคุมทางกฎหมาย การจัดการการตรวจสอบและการออกกฎหมายที่เกี่ยวข้องจึงต้องมีมากขึ้นและผู้เล่นมีการละเมิดกฎหมายหรือกลายเป็นเหยื่อในโลกออนไลน์มากขึ้น และในตลาดโลกแห่งความจริงในการค้าหรือแลกเปลี่ยนของคุณสมบัติเสมือนระหว่างผู้เล่นได้กลายเป็นเป็นที่แพร่หลาย แต่น่าเสียดายที่การใช้ที่ผิดกฎหมาย การหลอกลวงหรือโปรแกรมที่จะสามารถใช้ User ID ของคนอื่น ๆ หรือรหัสผ่านที่เพิ่มขึ้นเช่นกัน ซึ่งหัวข้อนี้จะพูดถึงภัยคุกคามการโกงเกมออนไลน์และความบกพร่องด้านความปลอดภัยตรวจสอบผลของมันสำหรับธุรกิจออนไลน์และแนวทางแก้ปัญหาด้านเทคนิคและผลกระทบ

2.5.1 ความหมายของการโกงเกมออนไลน์

การโกงโดยทั่วไปแล้วจะหมายถึง การแก้ไข ดัดแปลง หรือ พฤติกรรมที่ผู้เล่นใช้เพื่อได้มาซึ่งความได้เปรียบอย่างไม่เป็นธรรม หรือเอื้อประโยชน์แก่ผู้เล่นมากกว่าผู้เล่นคนอื่น ๆ เช่น แสดงข้อมูลบางส่วนที่ผู้เล่นไม่ควรทราบ ปรับแต่งเกมเพื่อให้ผู้เล่นสามารถกระทำบางสิ่งที่โดยปกติไม่ได้รับอนุญาต หรือการใช้โปรแกรมช่วยเล่นอัตโนมัติ

ถึงแม้ว่าในปัจจุบัน เกมส่วนมากเปิดโอกาสให้แก้ไขการตั้งค่าบางส่วน เช่น การเปลี่ยนแป้นพิมพ์ เพื่อให้ปุ่มมีความสะดวกในการกดมากขึ้น ซึ่งไม่ถือว่าเป็นการโกง และเป็นที่ยอมรับได้ทั่วไป แต่การเปลี่ยนรูปลักษณ์ตัวละครในเกม หรือเร่งความสว่างในจุดที่มืดนั้นยังคงเป็นที่โต้แย้งอยู่

การโกงเกมออนไลน์สามารถมองได้หลากหลายความหมาย โดยที่ผู้เล่นส่วนใหญ่มองว่า การโกงคือการที่ผู้เล่นได้รับผลประโยชน์ที่ไม่ยุติธรรมต่อผู้เล่นอื่น แต่มีผู้เล่นบางส่วนมองว่า การเปิดคู่มือเกม หรือการให้คนรู้จักช่วยเล่น หรือการตั้งแค้มป์ (Camping) เพื่อรอรับของ ก็ถือเป็นการโกง ดังนั้นจึงยากที่จะระบุว่าอะไรคือการโกงที่แท้จริง โดยต้องขึ้นอยู่กับกฎระเบียบต่าง ๆภายในเกมนั้น ๆ แต่การเข้าถึงส่วนของโค้ดเพื่อแก้ไข (Hack) หรือการใช้โค้ดโกง (Cheat code) นั้น ถือว่าเป็นส่วนหนึ่งของการโกง โดยการโกงจะแบ่งออกได้ 2 แบบโดยลักษณะ คือ การโกงที่มีการเปลี่ยนแปลงค่า หรือเปลี่ยนแปลงการทำงานของเกม และการโกงที่ไม่มีการเปลี่ยนแปลงค่าหรือการทำงานของเกม

2.5.1.1 การโกงที่มีการเปลี่ยนแปลงค่า หรือการเปลี่ยนแปลงการทำงานของเกม

การโกงชนิดนี้มักจะมีการใช้โปรแกรมประเภท Cheat Engine (Hook program) ในการเข้าถึง Memory หรือ Running process ของเกม เพื่อเปลี่ยนแปลงค่าต่าง ๆ เช่น จำนวนเงิน, ค่าความสามารถของตัวละคร, Skill-cool-down time, เป็นต้น ยังรวมไปถึงการเข้าไปแก้ไขฟังก์ชันการทำงานของเกม ในลักษณะต่าง ๆ เช่น ฟังก์ชันลด HP ของตัวละคร เมื่อถูกมอนสเตอร์โจมตี ให้กลายเป็นการเพิ่ม HP แทน, แก้ไขให้ตัวละครสามารถเดินเข้าไปยังพื้นที่ที่ตัวละครทั่วไปไม่สามารถเข้าถึงได้ (เดินบนน้ำ, เดินบนอากาศ) เป็นต้น

2.5.1.2 การโกงที่ไม่มีการเปลี่ยนแปลงค่า หรือการเปลี่ยนแปลงการทำงานของเกม

การโกงชนิดนี้จะไม่มีการเข้าไปแก้ไขการทำงานหรือค่าต่าง ๆ ในตัวเกม แต่จะใช้การเข้าถึง Resources ต่าง ๆของผู้ให้บริการเกม, การใช้ Macros, การใช้ Scripts, และการใช้อุปกรณ์ใด ๆ เพื่อบังคับเกมแทนผู้เล่น เช่นการใช้โปรแกรม Pokemon-Go-Bot หรือ โปรแกรมประเภท บอท (Bot) เพื่อปลอมตัวเป็น Client ติดต่อกับ Server ของผู้ให้บริการ และทำการ “เล่น” แทนผู้เล่นอัตโนมัติ, การใช้ Macros หรือคีย์ลัดที่ตั้งค่าไว้แล้ว เพื่อความรวดเร็วและแม่นยำที่ปุ่มเดียวแทนการกดปุ่มหลายปุ่ม, การใช้ Scripts เพื่อให้เกิด Event ตามเวลาที่ได้วางไว้อัตโนมัติ, หรือการสร้างอุปกรณ์ที่สามารถเล่นกดปุ่มตาม Note ของเกม Guitar Hero โดยการโกงชนิดนี้จะทำให้ผู้เล่นอื่นเสียเปรียบเนื่องจากเสียโอกาสระหว่างที่ไม่ได้เล่นเกม ในขณะที่ผู้เล่นที่ใช้วิธีการโกงจะสามารถเข้าถึงโอกาสต่าง ๆ หลายประเภท (Event ต่าง ๆ

ของเกม, EXP คูณ 2, โอกาสได้รับ Item น้อยกว่า) เนื่องจากผู้ให้บริการมักจะให้ผลตอบแทนผู้เล่นที่ใช้เวลาในการเล่นเกมมากกว่าผู้เล่นที่ใช้เวลาในเกมน้อยกว่า (ยังมีผู้เล่นที่จ่ายเงินให้ระบบมากกว่า สำคัญกว่าผู้เล่นที่จ่ายเงินน้อยกว่า แต่โดยส่วนมากมักจะเน้นอย่างแรกมากกว่า)

2.5.1 หมวดหมู่ของ โปรแกรมโกง

1. Auxiliary Program: เป็นโปรแกรมที่ให้ความช่วยเหลือผู้เล่นที่จะใช้งานเกมออนไลน์

2. Auto-Robot Program: เป็นตัวแทนของผู้เล่น(Bot) สำหรับเล่นอัตโนมัติ เช่นการฆ่ามอน-สเตอร์เสมือนฝ่ายตรงข้าม ฯลฯ

3. Trojan Horse ภายในโปรแกรม: เป็นโปรแกรมเพื่อให้ได้หมายเลขผู้เล่นออนไลน์อื่น ๆ ขอรหัสผ่านหรือข้อมูลส่วนตัว

4. Malicious Program: การโกงหรือการบังคับให้ผู้เล่นในการซื้อขาย เพิ่มโอกาสในการฆ่า (Probability of kill) เพิ่มความเร็วความเคลื่อนไหว การทุจริตการซื้อขาย เป็นต้น นอกจากนี้บางโปรแกรมที่เป็นอันตรายมากเมื่อมีการรวมกับไวรัสหรือเวิร์มและพฤติกรรมบางอย่างที่มีส่วนเกี่ยวข้อง

โดยการเปรียบเทียบถึงความอันตราย (Harmful) และความปลอดภัยที่ต้องคำนึงถึง (Security concern)ได้ตามตาราง 2.1

ตาราง 2.1 ผลกระทบของความอันตราย (Harmful impact) และความปลอดภัยที่ต้องคำนึงถึง (Security concern) ในโปรแกรมโกงเกม

Cheating Program Types	Auxiliary	Auto-Robot	Trojan Horse Inside	Malicious
Harmful Impact	Low	Small	Medium	Large
Security Concern	Low	Low	Large	Large

2.5.2 เป้าหมายของการโกง

หัวข้อนี้จะพูดถึงเป้าหมายของการโกงที่เกี่ยวข้องกับ Game Software, Memory, Hardware และ Internet connection

1. Game Software การใช้วิศวกรรมย้อนกลับ (Reverse engineering) สามารถใช้เพื่อวิเคราะห์ตัวเกม (Original program) ในรายละเอียดต่าง ๆ หรือสร้างโปรแกรมเกมใหม่ และการเก็บสถิติที่เกี่ยวข้องต่าง ๆโดยไม่ส่งผลกระทบกับตัวโปรแกรมหลัก (Main program)

2. Memory การติดตามตัวแปรที่ควบคุมลักษณะบางอย่างของโปรแกรม และเปลี่ยนแปลงข้อมูลนั้น ๆเพื่อเพิ่มผลประโยชน์ให้กับผู้เล่นที่โกงเกม

3. Hardware เพิ่มความสว่างหน้าจอ หรือใช้การ์ดจอกับไดร์เวอร์พิเศษ เพื่อให้ผู้เล่นสามารถมองทะลุกำแพงได้ คือตัวอย่างของการใช้ฮาร์ดแวร์เทคนิค

4. Internet connection ใช้การวิเคราะห์ Packet หรือเนื้อหาที่ Server ส่งไปยัง Client

2.5.3 อิทธิพลของการโกงเกมออนไลน์

ความสำเร็จหรือความล้มเหลวของเกมออนไลน์ ส่วนมากกำหนดโดยสิ่งที่ผู้เล่นจะได้รับจากการเล่นเกมนั้น ๆ โดยประสบการณ์ที่ผู้เล่นได้รับและการป้องกันการถูกโกงก็เป็นแรงผลักดันที่สำคัญของความสำเร็จของเกม และคำถามที่สำคัญสำหรับอุตสาหกรรมเกมออนไลน์คือ “จะทำอย่างไรให้เรามั่นใจว่าผู้เล่นแต่ละคนจะได้รับประสบการณ์ที่สนุกสนานกับเกมโดยเป็นธรรมและไม่มีการโกง” ดังนั้นหัวข้อนี้จึงได้รวบรวม 7 หมวดหมู่ของอิทธิพลที่ได้รับจากการโกงเกมออนไลน์

1. ทำลายการพัฒนาของการเล่นเกมออนไลน์

2. การเพิ่มจำนวนของคดีอาญาที่เกี่ยวข้องกับการทุจริต การปลอมแปลง โจรกรรมข้อมูล และการคุกคาม

3. ทำให้เกิดการแข่งขันที่ไม่เป็นธรรมของผู้เล่น

4. การได้เปรียบผู้เล่นคนอื่นอย่างผิดกฎหมาย หรือ ผู้จัดจำหน่ายโปรแกรมโกงนั้น ๆ

5. เป็นภัยทางการเงินของเหยื่อ ทำร้ายจิตใจ หรือ ความสัมพันธ์ในสังคม

6. เป็นการเพิ่มภาระและสิ้นเปลืองทรัพยากรของผู้จัดจำหน่าย เช่น ค่าใช้จ่ายของ Network bandwidth, Server processing และค่าใช้จ่ายเพิ่มเติมอื่น ๆ

7. ส่งผลกระทบต่อความน่าเชื่อถือของ log บน server ของเกม ในการเป็นหลักฐานดิจิตอล โดยเฉพาะเมื่อมีอาชญากรรมเข้ามาเกี่ยวข้อง

2.5.4 ข้อแนะนำและการป้องกัน

หัวข้อนี้จะกล่าวถึงข้อแนะนำและวิธีการที่ใช้รับมือกับการโกงและอาชญากรรมบนเกมออนไลน์ ได้แก่

1. หากไม่สามารถป้องกันได้ ก็ควรส่งเสริมบริษัทประกันเพื่อทดแทนการสูญเสีย

2. ปรับปรุงกลไกการตรวจสอบการโกงของเกม เพื่อลดการเปลี่ยนแปลงตัวเกม(Modification) เพื่อลดความเป็นไปได้ในการโกงเกม เช่น ระบบที่สามารถตรวจสอบพฤติกรรมที่ผิดปกติหรือการเปลี่ยนแปลง (Modification) ที่เกิดขึ้นและทำการแจ้งเตือน ตัวอย่างผู้ขายที่ใช้ระบบนี้เช่น Joe Wilcox

3. กำหนดรูปแบบการซื้อขายที่ปลอดภัย หรือมีช่องทางซื้อขายที่ชัดเจน ผู้เล่นควรจะระมัดระวังเป็นอย่างมาก ในการซื้อขายของออนไลน์ การแลกเปลี่ยน, ขาย, หรือซื้อของในเกม ผ่านบุคคลที่สามที่ได้รับการเชื่อถือนั้นปลอดภัยกว่าการทำด้วยตัวเอง

4. มีการแจ้งผู้เล่นอยู่เสมอ ว่าข้อมูลใด ๆ ที่ผู้ประกอบการ หรือผู้ขายต้องการ และไม่ต้องการอยู่เสมอ ทั้งบน เว็บไซต์ หรืออีเมล และอะไรที่คาดหวังว่าจะประสบ ในโอกาสต่าง ๆ

5. การสร้างแพลตฟอร์มการเตือนภัยล่วงหน้าของลูกค้าอย่างรวดเร็ว ผู้ขายสามารถให้ข้อมูลที่จำเป็นแก่ผู้ใช้หรือกรณีศึกษาเกี่ยวกับกิจกรรมที่น่าสงสัยหรือความผิดทางอาญาอื่น ๆ

6. สร้างURL ของเว็ปไซต์โดเมนที่ใกล้เคียงกัน เช่น www.googel.com และเชื่อมไปยังเว็ปไซต์ที่แท้จริงคือ www.google.com เพื่อป้องกันการปลอมแปลงเว็ปไซต์

7. มีการลงทะเบียนตัวตนของผู้ใช้ในสภาพแวดล้อมเช่น อินเทอร์เน็ตคาเฟ่เท่าที่จะทำได้ ร้านอินเทอร์เน็ตคาเฟ่ และสถานที่สาธารณะอื่น ๆ ควรให้การเล่นเกมออนไลน์มีการบันทึกตัวตนของลูกค้า เวลาของการใช้งานออนไลน์และข้อมูลอื่น ๆ ที่จะสนับสนุนการสอบสวนของกิจกรรมทางอาญา

8. มีการบันทึกข้อมูลที่มีการตรวจสอบเรียบร้อยและเก็บข้อมูลไว้อย่างน้อยเป็นเวลาสามเดือน และผู้ขายต้องเพิ่มประสิทธิภาพในการตรวจสอบ เช่นเดียวกับการบันทึกและจัดเก็บข้อมูลที่สำคัญ เช่น บันทึกการโอนทรัพย์สิน เพื่อใช้ในการตรวจสอบหรือการสืบสวน

9. การปรับปรุงการศึกษาทางศีลธรรมที่เกี่ยวข้อง เพื่อลดอาชญากรรมที่อาจเกิดขึ้น และพฤติกรรมที่มีผลต่อศีลธรรมของวัยรุ่นที่เล่นเกมออนไลน์

10. ให้ความรู้กับผู้เล่น เพื่อให้ผู้เล่นรักษา User Id และ Password เป็นความลับ

11. การปรับใช้กลไกการจัดการสิทธิสำหรับการใช้งาน โดยเฉพาะผู้ที่ได้รับอนุญาต

12. หน่วยงานทางกฎหมายสามารถใช้ "Honey pots" เพื่อล่อและจับผู้กระทำผิดในการเล่นเกมออนไลน์ โดย Honey pot จะเป็นระบบเกมออนไลน์หลอก เพื่อเก็บและรวบรวมข้อมูลที่เกี่ยวข้องกับกิจกรรมทางอาญา

2.5.5 บทสรุปการโกงเกมออนไลน์และความปลอดภัย

จากความสำเร็จของการเล่นเกมออนไลน์ แต่อิทธิพลเชิงลบของการเล่นเกมออนไลน์และการโกงเกมได้กลายเป็นปัญหาร้ายแรงให้กับวัยรุ่นและสังคม ซึ่งตามมาด้วยปัญหาการต่อต้านสังคม แม้ว่ากิจกรรมการเล่นเกมออนไลน์และพฤติกรรมการโกง มีความผิดทางอาญาไม่ได้ร้ายแรงเท่าอาชญากรรมธรรมดา แต่ผู้คนจำเป็นต้องตระหนักถึงปัญหาที่เกิดขึ้น รวมไปถึงหามาตรการเพื่อลดการโกงเกมและอาชญากรรมออนไลน์ และความบันเทิงควรจะกลับไปเป็นความบันเทิง มากกว่าอาชญากรรมหรือปัญหาทางสังคม โดยอุตสาหกรรมเกมต้องควบคุมเรื่องนี้อย่างจริงจังและคำนึงถึงกฎหมายของแต่ละประเทศ เพื่อศักยภาพในการเติบโตของ อุตสาหกรรมเกม นอกจากนี้ผู้จัดจำหน่ายเกมต้องให้ความสำคัญกับการให้ความรู้ต่อผู้เล่น และปรับปรุงกลไกการป้องกันการโกงเกมออนไลน์ โดยปัญหาการโกงเกมออนไลน์ควรเป็นที่รับรู้ของประชาชนทั่วไป และมีการแก้ไขผ่านการศึกษาหรือข้อกฎหมาย และเทคโนโลยี

2.6 Geolocation-based game

Geolocation-based game เป็นความบันเทิงรูปแบบใหม่ซึ่งเข้ามามีบทบาทในที่ผู้เล่นสามารถเล่นในสถานที่จริง ด้วยใช้อุปกรณ์สวมใส่ต่าง ๆ หรือ Smart phone โดยอุปกรณ์จะส่งค่าต่าง ๆ ของเซนเซอร์ใน

อุปกรณ์ บ่งบอกข้อมูลที่อยู่ปัจจุบัน และพฤติกรรมการเล่นของผู้เล่น เป็นต้น ซึ่งอ้างอิงตามสภาพแวดล้อมของผู้เล่นจริง สำหรับเกมส์ที่ร่วมเล่นกับผู้อื่นข้อมูลนี้จะส่งไปยังผู้เล่นผู้อื่นที่กำลังเล่นอยู่ ซึ่งส่งผลให้ประสบการณ์การเล่นกับเหตุการณ์ในชีวิตประจำวันของผู้เล่นมีความเกี่ยวข้องกันมากขึ้น

ซึ่งเกมส์รูปแบบนี้เป็นที่น่าจับตามองในตลาด ส่วนมากสร้างในระบบสมาร์ทโฟน ซึ่งทางตลาดคาดการณ์ไว้ว่าตลาดนี้จะสามารถเติบโตได้ถึงระดับพันล้านในอีกไม่กี่ปีข้างหน้า โดยเฉพาะอย่างยิ่งส่งผลโดยตรงกับรายได้ของระบบ 3G ตัวอย่างของเกมส์ประเภทนี้ เช่น เกมโปเกมอน โก (Pokémon go) เป็นต้น

2.7 API

2.7.1 หน้าที่ API

API ทำหน้าที่ช่วยในการเข้าถึงข้อมูลต่าง ๆ หรือจะเป็นการนำข้อมูลต่าง ๆออกจากเว็บไซต์ หรือจะเป็นการส่งข้อมูลเข้าไปก็ได้ โดยเจ้าของเว็บไซต์ที่มี API จะกำหนดขอบเขตในการเข้าถึงบริการต่าง ๆของทางเว็บไซต์

2.7.2 ประโยชน์ของ API

- ช่วยในการพัฒนาเว็บไซต์หรือ Application ได้ง่ายและรวดเร็วซึ่ง API จะเป็นตัวช่วยที่นักพัฒนาไม่ต้องเข้าไปแก้ไข Code คำสั่งเลยทำให้สะดวกสบายในการใช้งาน
- ช่วยให้นักพัฒนาเว็บไซต์หรือเจ้าของเว็บไซต์สามารถฐานผู้ชมเว็บไซต์ให้มากขึ้น
- ทำให้ผู้ใช้งานเว็บไซต์ต่าง ๆ ที่มีการติดตั้ง API ของอีกเว็บไซต์หนึ่ง ไม่ต้องเข้าหน้าเว็บไซต์ที่เป็นเจ้าของ API เพียงแต่เข้ามายังเว็บไซต์ที่มีการติดตั้ง API เท่านั้นทำให้การรับรู้ข่าวสารต่าง ๆทั่วถึงกันและสะดวกในการใช้งานของผู้ใช้งานเว็บไซต์
- API สามารถรับส่งข้อมูลข้าม Server ได้
- ในปัจจุบันเว็บไซต์ใหญ่ ๆ หลายเว็บไซต์จะมีการเปิดให้ใช้งาน API ซึ่งเราอาจจะเห็นการใช้งาน API ได้มากขึ้นโดยเฉพาะเว็บไซต์ที่ด้านการติดต่อสื่อสาร Social Network และ E-commerce

2.7.3 ตัวอย่างการใช้งานของ API

โดยส่วนมากแล้วเราจะเห็น API ถูกใช้งานกันอย่างแพร่หลาย ที่เห็นได้กันอย่างชัดเจนก็คือบริการของ Amazon มี API ที่เปิดให้ผู้ที่สนใจที่จะเป็นตัวแทนขายสินค้าหรือเจ้าของเว็บทั่วไป ได้นำสินค้าที่มีขายอยู่ใน Amazon ไปติดไว้ในเว็บไซต์หรือบล็อคของตัวเองได้ โดยเจ้าของเว็บไซต์หรือผู้สนใจจะได้รับคอมมิสชั่นเมื่อมีการคลิกซื้อสินค้าจากเว็บไซต์หรือบล็อคที่นำ API ไปติดตั้ง อีกบริการหนึ่งก็คือบริการของ PayPal API ซึ่งเจ้าของเว็บไซต์ที่ต้องการเพิ่มช่องทางการชำระเงินให้กับลูกค้าก็สามารถนำ PayPal API ไปติดตั้งที่เว็บไซต์ที่ต้องการได้ เพื่อเพิ่มความสะดวกสบายให้กับลูกค้าที่มาใช้บริการในเว็บไซต์นั่นเอง

นอกจากเว็บด้านอีคอมเมิร์ซและยังมีเว็บไซต์ด้านสังคมออนไลน์หรือ Social Network ที่นำ API ไปใช้งานด้วย เช่น Facebook และ Twitter ที่สามารถนำกล่องแสดงความคิดเห็นไปติดในเว็บไซต์ที่ต้องการได้

ในโครงงานนี้จะศึกษามุ่งเน้นไปยัง Google map API เพราะ API ที่ใช้ในการทดลองโดย Google map API เป็นบริการของ Google ที่ไว้ใช้บริการ google map เพื่อที่ดึงข้อมูลแผนที่ ที่ทาง Google ได้เตรียมไว้ ไปใช้งานได้ เป็นต้น

บทที่ 3
การวิเคราะห์รูปแบบการตรวจจับ

เกม Pokémon GO นั้นมีการอัพเดท Client ตัวเกมอยู่เป็นระยะ และทางฝั่ง Server ของผู้ให้บริการ จากการสังเกต และศึกษา พบว่า มีการอัพเดท และเปลี่ยนแปลงค่า Parameter ต่าง ๆของฟังก์ชันต่าง ๆ ดังนั้นโปรแกรมช่วยเล่น จะสามารถถูกตรวจจับได้ ถ้าการติดต่อกับทาง Server นั้นใช้ฟังก์ชันเวอร์ชั่นเก่า

อีกทั้ง ในช่วงหลังความนิยมของเกมลดลง ทำให้ทีมที่ทำการ Reverse engineer ระบบของตัวเกมทำได้ช้าลง

จากการสังเกต ในตัว Client จะมี key เฉพาะ เพื่อตรวจสอบผู้ที่ทำการส่งคำร้อง หรือติดต่อไปยัง Server ซึ่งถ้าไม่ตรงตาม key ปัจจุบัน ก็จะสามารถตรวจจับได้ ว่าไม่ใช่ผู้เล่นที่ใช้ Client จากทางผู้ให้บริการ

การวิเคราะห์โดยการทดลอง ทดลองโดยการใช้งานโปรแกรมช่วยเล่น (Bot) ในการโกงเกม Pokémon Go โดยจะแบ่งกลุ่มทดลองออกเป็น

- เลียนแบบผู้เล่นจริง เพื่อใช้เป็นตัวควบคุม
- เลียนแบบผู้เล่นจริง แต่มีพฤติกรรมบางอย่าง ที่ผิดปกติ
- เลียนแบบผู้เล่นจริง และพฤติกรรมท้าทายการระงับบัญชีผู้ใช้ นอกเหนือจากกลุ่มทดลองข้างต้นแล้ว ยังมีการทดลองอื่น เพื่อทดสอบในกรณีที่ผิดปกติ
- ส่ง Geolocation ไปยัง Server พร้อมกันสองตำแหน่ง
- Login พร้อมกันจาก IP address ต่างกัน

ผลลัพธ์ที่ได้จากการทดลองคือบันทึก หรือ Log ของโปรแกรมช่วยเล่น ซึ่งบอก Action ต่าง ๆ พร้อม Timestamp และนำมาวิเคราะห์ต่อไป ว่าลักษณะใด ที่ทำให้สามารถถูกตรวจจับได้ว่าเป็นโปรแกรมช่วยเล่น ไม่ใช่ผู้เล่นจริง

3.1 Reverse engineering: Pokémon Go

การ Reverse engineering Application Pokémon Go นั้นได้อาศัยเทคนิคการตรวจสอบการทำงานของโปรแกรม และ ตรวจสอบการสื่อสาร เป็นหลัก โดยมีขั้นตอนแบ่งออกได้ดังนี้

การตรวจสอบการทำงานของโปรแกรม

- ระบุไฟล์ Android application package (APK) ของ Pokémon Go
- ใช้เครื่องมือในการ Decompile ไฟล์ Binary เพื่อให้สามารถอ่าน, ศึกษา และทำความเข้าใจ ในภาษาต่าง ๆ (JAVA)
- ตั้งสมมติฐานเกี่ยวกับการทำงานของ function ต่าง ๆ และทำการทดสอบ

- ตั้งชื่อ function ให้เหมาะสม

3.1.1 การ Bypass Certificate pinning ของ HTTPS ในกรณีของ Pokémon Go

1. ค้นหา leaf certificate's public key จากแหล่งที่น่าเชื่อถือได้ เช่น Website
2. ใช้ข้อมูล Public key's byte sequence เพื่อสืบหา function การทำงานสำหรับตรวจสอบ certificate
3. เมื่อพบ function การตรวจสอบ certificate แล้วจึงใช้เทคนิคต่าง ๆ แก้ไขการทำงาน
   a. ข้ามการตรวจสอบ certificate
   b. ส่งข้อมูล certificate ปลอมเพื่อตรวจสอบ
4. แปลงไฟล์ Binary เป็นภาษา low level language เช่น ARM Assembly Language เพื่อแก้ไข Instruction
5. เมื่อแก้ไขแล้วถึงแปลงไฟล์เป็น Binary และสร้างเป็น APK สำหรับติดตั้ง

Pokémon Go ยังมีการใช้งาน hash function ในลักษณะของ Signature เพื่อตรวจสอบ Client ว่าเป็น 3rd party application หรือไม่อีกด้วย

การทำซ้ำ Hash function จำเป็นจะต้องทราบถึงข้อมูลตั้งต้น, วิธีการ และ ลำดับในการคำนวณ เพื่อให้ได้ Signature ที่ถูกต้อง การ Reverse engineering hash function ทำได้โดยการ ใช้เครื่องมือ Debugger เช่น ARM CPU emulator เพื่อตรวจสอบค่าใน register แต่ละตัว และตั้ง Break points เพื่อดูการทำงานของ Instructions

3.1.2 การตรวจสอบการสื่อสาร ระหว่าง Client และ Server

การสื่อสารระหว่าง Client – Server อยู่ในรูปแบบของ Remote Procedure Call (RPC) ผ่าน HTTPS Protocol ซึ่งการช่วงแรกที่เปิดให้บริการ Niantic Labs ไม่มีการใช้ Certificate ในการตรวจสอบ Client ที่ทำการเชื่อมต่อ ส่งผลให้ สามารถใช้เครื่องมือต่าง ๆ ในการ Monitor Network Traffic ได้ นำไปสู่การศึกษา Communication Protocol ของเกม Pokémon GO

เทคนิค Man in the Middle ถูกใช้ในการศึกษาการติดต่อสื่อสาร ทำให้ทราบได้ว่า Pokémon GO ได้ใช้ Protocol buffer 3 เพื่อสร้าง Communication Protocol และ base64 เข้ารหัสข้อมูลที่ส่งผ่าน Network

3.1.2.1 Protocol Buffer 3

ข้อมูลที่ใช้ Protocol Buffer นั้นจะอยู่ในรูปแบบของ key และ value คู่กัน โดย key จะเป็นตัวเลข และชนิดของ Value จะบอกได้จาก Tag 3 bits โดยจะมีการกำหนดไฟล์ .proto ขึ้นก่อนการสร้างข้อความ หรือ Message

```
// See README.txt for information and build instructions.
//
// Note: START and END tags are used in comments to define sections used in
// tutorials.  They are not part of the syntax for Protocol Buffers.
//
// To get an in-depth walkthrough of this file and the related examples, see:
// https://developers.google.com/protocol-buffers/docs/tutorials

// [START declaration]
syntax = "proto3";
package tutorial;
// [END declaration]

// [START java_declaration]
option java_package = "com.example.tutorial";
option java_outer_classname = "AddressBookProtos";
// [END java_declaration]

// [START csharp_declaration]
option csharp_namespace = "Google.Protobuf.Examples.AddressBook";
// [END csharp_declaration]

// [START messages]
message Person {
  string name = 1;
  int32 id = 2;  // Unique ID number for this person.
  string email = 3;

  enum PhoneType {
    MOBILE = 0;
    HOME = 1;
    WORK = 2;
  }

  message PhoneNumber {
    string number = 1;
    PhoneType type = 2;
  }

  repeated PhoneNumber phones = 4;
}

// Our address book file is just one of these.
message AddressBook {
  repeated Person people = 1;
}
// [END messages]
```

รูปที่ 3.1 ตัวอย่างไฟล์ addressbook.proto จาก Protocol Buffer JAVA Tutorial

การจะอ่านข้อมูลที่ถูกส่งด้วย Protocol Buffer นั้น ต้องมี Schema เนื่องจาก ข้อมูลที่รับและส่งสามารถเป็นชนิดใดก็ได้ เช่น ตัวแปร 32 bits สามารถเป็นได้ทั้ง Floating point number หรือ Unsigned Integer หรือ Signed Integer ก็ได้ ดังนั้นเครื่องมือในการแปลงข้อมูลจึงค่อนข้างจำเป็น ในการวิเคราะห์และทำการ Reverse Engineering

จากการศึกษาพบว่า ข้อมูลจะอยู่ในรูปแบบเข้ารหัสด้วย Protocol Buffer แล้วจึงเข้ารหัสด้วย base64 เพื่อส่งผ่าน Internet

```
00000000: 436b 454b 4157 4551 4152 6f4c 5955 426c  CkEKAWEQARoLYUBl
00000010: 6257 4670 6243 356a 6232 3069 4467 6f4d  bWFpbC5jb20iDgoM
00000020: 4d44 6778 4c54 4578 4d53 3078 4d54 4578  MDgxLTExMS0xMTEx
00000030: 4967 384b 437a 4179 4c54 4578 4d53 3078  Ig8KCzAyLTExMS0x
00000040: 4d54 4578 4541 4569 4441 6f4b 4d44 4578  MTExEAEiDAoKMDEx
00000050: 4d7a 4930 4f54 5534 4e67 6f6b 4367 4669  MzI0OTU4NgokCgFi
00000060: 4541 4961 4332 4a41 5a57 3168 6157 7775  EAIaC2JAZW1haWwu
00000070: 5932 3974 4968 414b 4444 4134 4d69 3079  Y29tIhAKDDA4Mi0y
00000080: 4d6a 4974 4d6a 4979 4d68 4143 436a 554b  MjItMjIyMhACCjUK
00000090: 4346 4a6c 5957 5167 5447 3933 454d 4441  CFJlYWQgTG93EMDA
000000a0: 6d42 7361 446e 4a73 6233 6441 5a57 3168  mBsaDnJsb3dAZW1h
000000b0: 6157 7775 5932 3974 4968 514b 4543 7332  aWwuY29tIhQKECs2
000000c0: 4d43 3034 4e7a 6774 4f54 6b35 4c54 5532  MC04Nzgt0Tk5LTU2
000000d0: 4e7a 5151 4167 3d3d 0a                   NzQQAg==.
```

รูปที่ 3.2 ข้อมูลดิบ Encoded ด้วย base64

รูปที่ 3.1 แสดงให้เห็นลักษณะของข้อมูลที่สามารถดักจับได้ระหว่างการสื่อสาร ในการจะได้ข้อมูล จะต้องนำไปถอดรหัสด้วย algorithm base64 ผลลัพธ์จะได้ดังรูปที่ 3.2 โดยทั้งสองรูปใช้คำสั่ง xxd เพื่อแสดงในรูปแบบ Binary (x86_64)

```
00000000: 0a41 0a01 6110 011a 0b61 4065 6d61 696c  .A..a....a@email
00000010: 2e63 6f6d 220e 0a0c 3038 312d 3131 312d  .com"...081-111-
00000020: 3131 3131 220f 0a0b 3032 2d31 3131 2d31  1111"...02-111-1
00000030: 3131 3110 0122 0c0a 0a30 3131 3332 3439  111.."...0113249
00000040: 3538 360a 240a 0162 1002 1a0b 6240 656d  586.$..b....b@em
00000050: 6169 6c2e 636f 6d22 100a 0c30 3832 2d32  ail.com"...082-2
00000060: 3232 2d32 3232 3210 020a 350a 0852 6561  22-2222...5..Rea
00000070: 6420 4c6f 7710 c0c0 981b 1a0e 726c 6f77  d Low.......rlow
00000080: 4065 6d61 696c 2e63 6f6d 2214 0a10 2b36  @email.com"...+6
00000090: 302d 3837 382d 3939 392d 3536 3734 1002  0-878-999-5674..
```

รูปที่ 3.3 ข้อมูลหลัง Decoded ด้วย base64 ซึ่ง Encoded ด้วย Protocol Buffer

จากรูปที่ 3.2 จะเห็นว่าข้อมูลถูกเข้ารหัสด้วย Protocol Buffer โดยเราสามารถถอดข้อความออกมาได้ด้วยคำสั่ง protoc –decode_raw จะได้ดังรูปที่ 3.3 แต่ข้อจำกัดจากการถอดข้อความเมื่อไม่มีไฟล์ Schema หรือ template .proto ที่ใช้ในการสร้าง จะไม่สามารถทราบว่า Value ต่าง ๆ คืออะไร และเป็นชนิดใด (String, Unsigned Integer, Signed Integer, etc.)

```
1 {
  1: "a"
  2: 1
  3: "a@email.com"
  4 {
    1: "081-111-1111"
  }
  4 {
    1: "02-111-1111"
    2: 1
  }
  4 {
    1: "0113249586"
  }
}
1 {
  1: "b"
  2: 2
  3: "b@email.com"
  4 {
    1: "082-222-2222"
    2: 2
  }
}
1 {
  1: "Read Low"
  2: 57024576
  3: "rlow@email.com"
  4 {
    1: "+60-878-999-5674"
    2: 2
  }
}
```

รูปที่ 3.4 ข้อมูล Decoded ด้วย Protocol Buffer (protoc –decode_raw)

โปรแกรม protofudger ถูกพัฒนาขึ้น โดยนักพัฒนา Conrad Pankoff เพื่อตรวจสอบ tag ของ Value แต่ละค่า และทำการคาดเดา type หรือชนิดของ Value ให้ง่ายต่อการศึกษาต่อและนำไปใช้วิเคราะห์ ดังรูปที่ 3.4 จะเห็นว่าผลลัพธ์จากโปรแกรมสามารถบอกอย่างคร่าว ๆ ได้ว่าข้อมูลแต่ละบรรทัดคือ type ใด

```
decoded 3 fields

1: {
  1: (string) "a"
  2: (varint) 1
  3: (string) "a@email.com"
  4: {
    1: (string) "081-111-1111"
  }
  4: {
    1: {
      6: (varint) 50
      5: (int64be) 3544668451885887793
    }
    2: (varint) 1
  }
  4: {
    1: {
      6: (varint) 49
      6: (int64le) 152614499865196O3
    }
  }
}
1: {
  1: (string) "b"
  2: (varint) 2
  3: (string) "b@email.com"
  4: {
    1: (string) "082-222-2222"
    2: (varint) 2
  }
}
1: {
  1: (string) "Read Low"
  2: (varint) 57024576
  3: (string) "rlow@email.com"
  4: {
    1: (string) "+60-878-999-5674"
    2: (varint) 2
  }
}
```

รูปที่ 3.5 ผลลัพธ์ที่ได้จากโปรแกรม protofudger คาดเดา type ของตัวแปรต่างๆ

และเมื่อนำไฟล์ .proto หรือ Schema สำหรับ decode ด้วยการใช้คำสั่ง protoc –decode tutorial.AddressBook addressbook.proto < data (--decode tutorial.AddressBook เพื่อถอดข้อความจาก Message AddressBook) จะได้ผลลัพธ์ดังรูปที่ 3.5 ซึ่งจะเห็นว่าข้อมูลถูกแปลงอยู่ในรูปแบบที่ถูกต้องและเข้าใจได้แล้ว

```
people {
  name: "a"
  id: 1
  email: "a@email.com"
  phones {
    number: "081-111-1111"
  }
  phones {
    number: "02-111-1111"
    type: HOME
  }
  phones {
    number: "0113249586"
  }
}
people {
  name: "b"
  id: 2
  email: "b@email.com"
  phones {
    number: "082-222-2222"
    type: WORK
  }
}
people {
  name: "Read Low"
  id: 57024576
  email: "rlow@email.com"
  phones {
    number: "+60-878-999-5674"
    type: WORK
  }
}
```

รูปที่ 3.6 แสดงข้อมูลที่ใช้ Protocol Buffer decode ด้วย Schema

บทที่ 4

การทดลองและผลการทดลอง

4.1 การทดลองโกงเกม Pokémon GO

การทดลองระบบตรวจจับในเกม Pokémon Go ทำโดยการใช้โปรแกรมอัตโนมัติในการเล่น หรือ Bot เพื่อทดลอง โดยทำการใช้งาน 3rd party software ที่ชื่อว่า PokemonGo-Bot

การทดลองประกอบด้วยบัญชี Pokémon Trainer Club ทั้งหมด 15 บัญชี ตั้งแต่วันที่ 16 พฤศจิกายน พ.ศ. 2559 โดยมีการตั้งค่าทั้งหมด 6 ลักษณะ ดังต่อไปนี้

1. (config_nosleep.json) Bot ทำการเลียนแบบผู้เล่น โดยเดินอย่างต่อเนื่อง เข้าหาเสาโปเกมอน ด้วยความเร็วระหว่าง 2.16 ถึง 4.16 เมตรต่อวินาที จับโปเกมอนทุกตัวที่พบ การปาโปเกบอลมีความเป็นไปได้เลียนแบบผู้เล่นจริง เมื่อกระเป๋าเก็บไอเทมใกล้เต็ม จะทำการทิ้งไอเทมบางส่วน แลกโปเกมอนกับลูกอมตามที่ตั้งค่าไว้ นำไข่ใส่เครื่องฟักไข่เสมอ พัฒนาร่างโปเกมอนเมื่อลูกอมเพียงพอเสมอ หมุนเสาเพื่อเก็บไอเทมเสมอ จำกัดโปเกมอนที่จับได้สูงสุด 800 ตัว และหมุนเสาได้สูงสุด 1900 ต้น ต่อวัน

2. (config_p2000_f3000.json) Bot ทำการเลียนแบบผู้เล่น โดยเดินอย่างต่อเนื่อง เข้าหาเสาโปเกมอน ด้วยความเร็วระหว่าง 2.16 ถึง 4.16 เมตรต่อวินาที จับโปเกมอนทุกตัวที่พบ การปาโปเกบอลมีความเป็นไปได้เลียนแบบผู้เล่นจริง เมื่อกระเป๋าเก็บไอเทมใกล้เต็ม จะทำการทิ้งไอเทมบางส่วน แลกโปเกมอนกับลูกอมตามที่ตั้งค่าไว้ นำไข่ใส่เครื่องฟักไข่เสมอ พัฒนาร่างโปเกมอนเมื่อลูกอมเพียงพอเสมอ หมุนเสาเพื่อเก็บไอเทมเสมอ จำกัดโปเกมอนที่จับได้สูงสุด 2000 ตัว และหมุนเสาได้สูงสุด 3000 ต้น ต่อวัน

3. (config_walk14_p1100_f2100.json) Bot ทำการเลียนแบบผู้เล่น โดยเดินอย่างต่อเนื่อง เข้าหาเสาโปเกมอน ด้วยความเร็วระหว่าง 2.16 ถึง 14 เมตรต่อวินาที จับโปเกมอนทุกตัวที่พบ การปาโปเกบอลมีความเป็นไปได้เลียนแบบผู้เล่นจริง เมื่อกระเป๋าเก็บไอเทมใกล้เต็ม จะทำการทิ้งไอเทมบางส่วน แลกโปเกมอนกับลูกอมตามที่ตั้งค่าไว้ นำไข่ใส่เครื่องฟักไข่เสมอ พัฒนาร่างโปเกมอนเมื่อลูกอมเพียงพอเสมอ หมุนเสาเพื่อเก็บไอเทมเสมอ จำกัดโปเกมอนที่จับได้สูงสุด 1100 ตัว และหมุนเสาได้สูงสุด 2100 ต้น ต่อวัน

4. (config_walk14_p1100_f2100_excellent_rate_1.json) Bot ทำการเลียนแบบผู้เล่น โดยเดินอย่างต่อเนื่อง เข้าหาเสาโปเกมอน ด้วยความเร็วระหว่าง 2.16 ถึง 14 เมตรต่อวินาที จับโปเกมอนทุกตัวที่พบ การปาโปเกบอลได้ Excellent ทุกครั้ง เมื่อกระเป๋าเก็บไอเทมใกล้เต็ม จะทำการทิ้งไอ-เทมบางส่วน แลกโปเกมอนกับลูกอมตามที่ตั้งค่าไว้ นำไข่ใส่เครื่องฟักไข่เสมอ พัฒนาร่างโปเก-มอนเมื่อลูกอมเพียงพอเสมอ หมุนเสาเพื่อเก็บไอเทมเสมอ จำกัดโปเกมอนที่จับได้สูงสุด 1100 ตัว และหมุนเสาได้สูงสุด 2100 ต้น ต่อวัน

5. (config_walkconstant14_p9999_f9999.json) Bot ทำการเลียนแบบผู้เล่น โดยเดินอย่างต่อเนื่องเข้าหาเสาโปเกมอน ด้วยความเร็ว 14 เมตรต่อวินาที จับโปเกมอนทุกตัวที่พบ การปาโปเกบอลมีความเป็นไปได้เลียนแบบผู้เล่นจริง เมื่อกระเป๋าเก็บไอเทมใกล้เต็ม จะทำการทิ้งไอเทมบางส่วน แลกโปเกมอนกับลูกอมตามที่ตั้งค่าไว้ นำไข่ใส่เครื่องฟักไข่เสมอ พัฒนาร่างโปเกมอนเมื่อลูกอมเพียงพอเสมอ หมุนเสาเพื่อเก็บไอเทมเสมอ จำกัดโปเกมอนที่จับได้สูงสุด 9999 ตัว และหมุนเสาได้สูงสุด 9999 ต้น ต่อวัน

6. (config_walkconstant14_p9999_f9999.json) Bot ทำการเลียนแบบผู้เล่น โดยเดินอย่างต่อเนื่องเข้าหาเสาโปเกมอน ด้วยความเร็ว 14 เมตรต่อวินาที จับโปเกมอนทุกตัวที่พบ การปาโปเกบอลมีความเป็นไปได้เลียนแบบผู้เล่นจริง เมื่อกระเป๋าเก็บไอเทมใกล้เต็ม จะทำการทิ้งไอเทมบางส่วน แลกโปเกมอนกับลูกอมตามที่ตั้งค่าไว้ นำไข่ใส่เครื่องฟักไข่เสมอ พัฒนาร่างโปเกมอนเมื่อลูกอมเพียงพอเสมอ หมุนเสาเพื่อเก็บไอเทมเสมอ จำกัดโปเกมอนที่จับได้สูงสุด 9999 ตัว และหมุนเสาได้สูงสุด 9999 ต้น ต่อวัน

7. (config_walkconstant14_p9999_f9999_excellent_rate_1_gps_no_noise. json) Bot ทำการเลียนแบบผู้เล่น โดยเดินอย่างต่อเนื่อง เข้าหาเสาโปเกมอน ด้วยความเร็ว 14 เมตรต่อวินาที จับโปเกมอนทุกตัวที่พบ การปาโปเกบอลได้ Excellent ทุกครั้ง เมื่อกระเป๋าเก็บไอเทมใกล้เต็ม จะทำการทิ้งไอเทมบางส่วน แลกโปเกมอนกับลูกอมตามที่ตั้งค่าไว้ นำไข่ใส่เครื่องฟักไข่เสมอ พัฒนาร่างโปเกมอนเมื่อลูกอมเพียงพอเสมอ หมุนเสาเพื่อเก็บไอเทมเสมอ จำกัดโปเกมอนที่จับได้สูงสุด 9999 ตัว และหมุนเสาได้สูงสุด 9999 ต้น ต่อวัน ไม่มีการสร้างค่าหลอก GPS แกน xyz

4.2 ผลการทดลองการโกงโดยใช้ PokemonGo-Bot

จากข้อมูลที่ได้ทำการทดลอง และคัดกรองแล้ว ตัวอย่างต่าง ๆมีการถูก Soft ban หรือระงับการใช้งานชั่วคราว โดยข้อมูลที่นำมาพิจารณานั้นจะอยู่ในช่วงก่อนที่จะถูก Soft ban 24 ชั่วโมง ได้ผลที่น่าสนใจดังนี้ (การทดลองระหว่างวันที่ 17 กันยายน พ.ศ. 2559 ถึงวันที่ 6 ตุลาคม พ.ศ. 2559)

ตาราง 4.1 ตารางแสดงจำนวนครั้งที่ถูก Soft ban

Account	Soft ban
projectpokemon1	5
projectpokemon2	6
projectpokemon3	2
projectpokemon4	3
projectpokemon5	5
projectpokemon6	3

projectpokemon7	2
projectpokemon8	3
projectpokemon9	1
projectpokemon10	0
projectpokemon11	2

ตาราง 4.2 ตารางแสดงจำนวนสถิติก่อน Soft ban ของ projectpokemon1

	Caught Pokémon	Spun Pokestop	Error	Login
softban_log_20160923193546_20160924193546.txt	653	1712	17	53
softban_log_20160924005324_20160925005324.txt	522	1814	18	54
softban_log_20160925231820_20160926231820.txt	605	1666	24	51
softban_log_20160926014818_20160927014818.txt	577	1692	24	51
softban_log_20161001191354_20161002191354.txt	747	1065	2	48

จากข้อมูลผลการทดลองที่เก็บได้ สรุปการทดลองได้ว่า การ Login เข้าสู่ระบบ มากกว่า 50 ครั้ง จับ Pokémon มากกว่า 500 ตัว ขึ้นไปภายใน 1 วัน จะทำให้ระบบทำการ Soft ban ผู้เล่นอัตโนมัติ

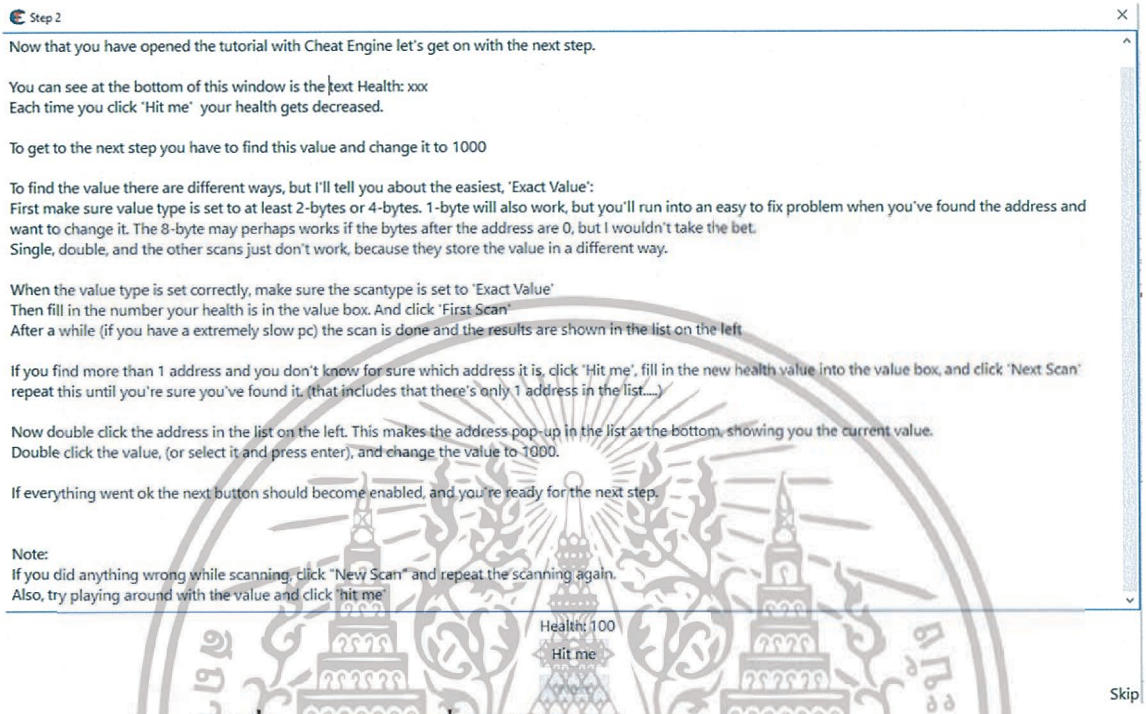
4.3 การทดลอง Cheat Engine

การทดลอง Cheat Engine โดยใช้ Cheat Engine 6.6 เพื่อทดลองโกมเกมออฟไลน์ต่าง ๆ

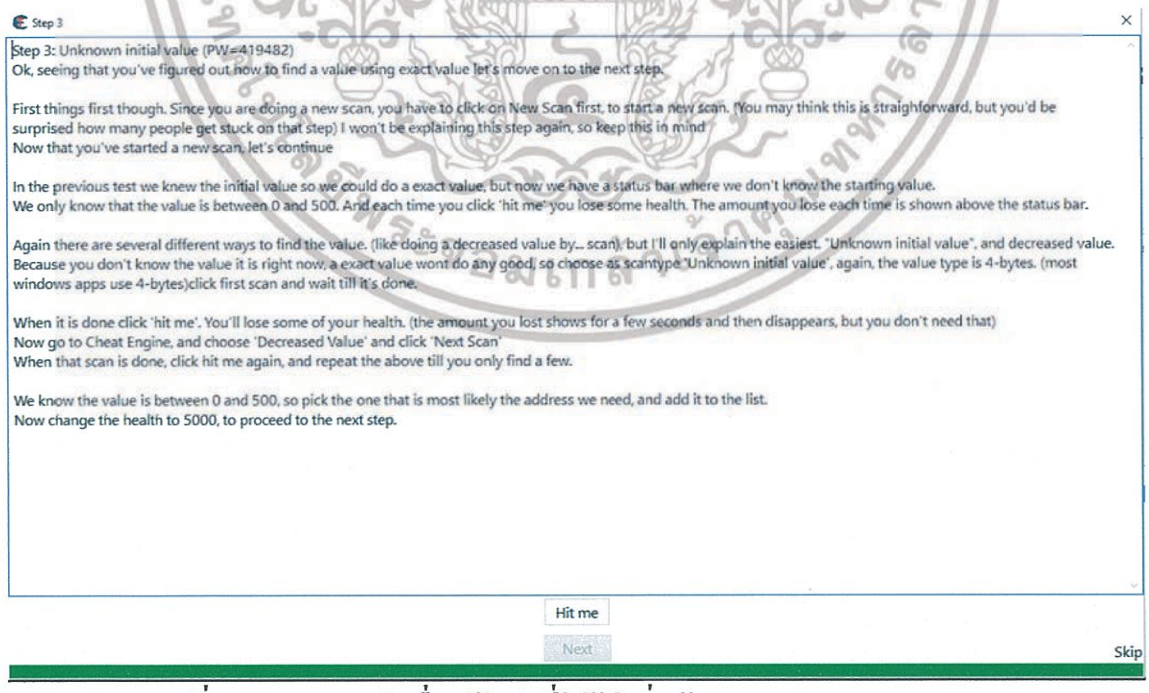
4.3.1 Tutorial

Cheat Engine tutorial จะทำการสอนการใช้งานการโกงโดยใช้การสแกนหน่วยความจำเบื้องต้น เช่น การสแกนเพื่อแก้ไขค่าโดยตรง (Exact Value scanning) การสแกนค่าเพื่อแก้ไขค่าที่ไม่รู้ค่าเริ่มต้น (Unknown initial value) การสแกนเพื่อแก้ไขค่าFloat หรือ Double การสแกนเพื่อหาและแก้ไข

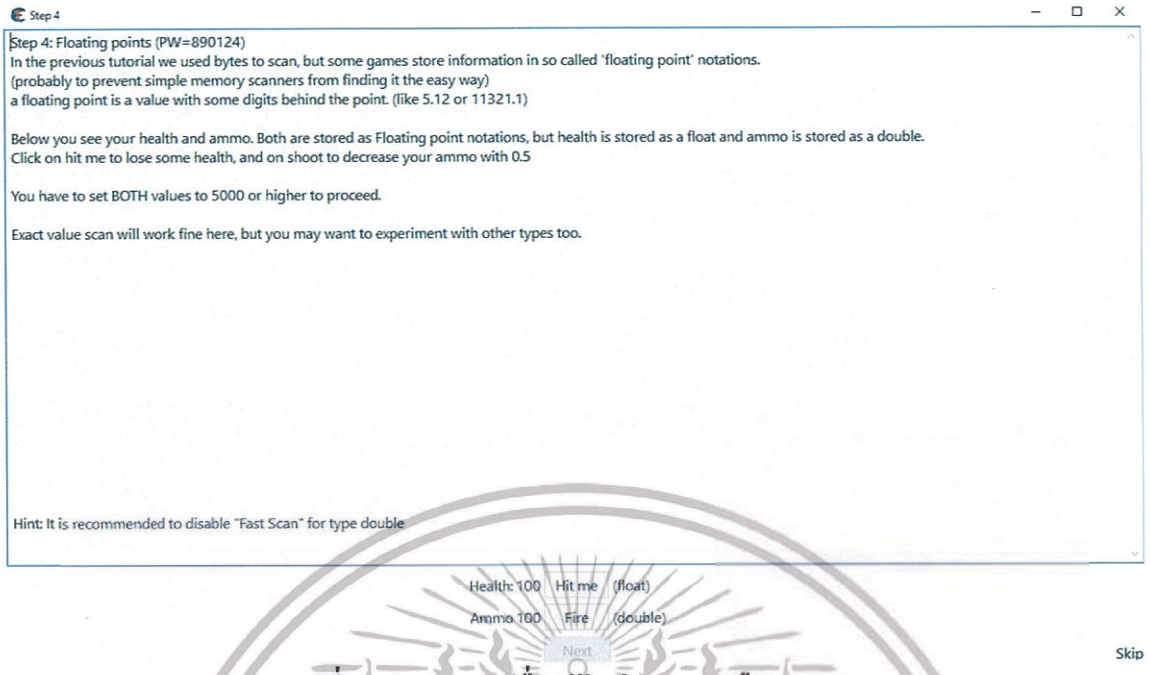
ส่วนของโค้ด (Code finder) การสแกนหา Pointer เพื่อแก้ไขค่า (Pointer and Multilevel pointers) และการแก้ไขโค้ดที่ใช้ร่วมกัน (Shared code)



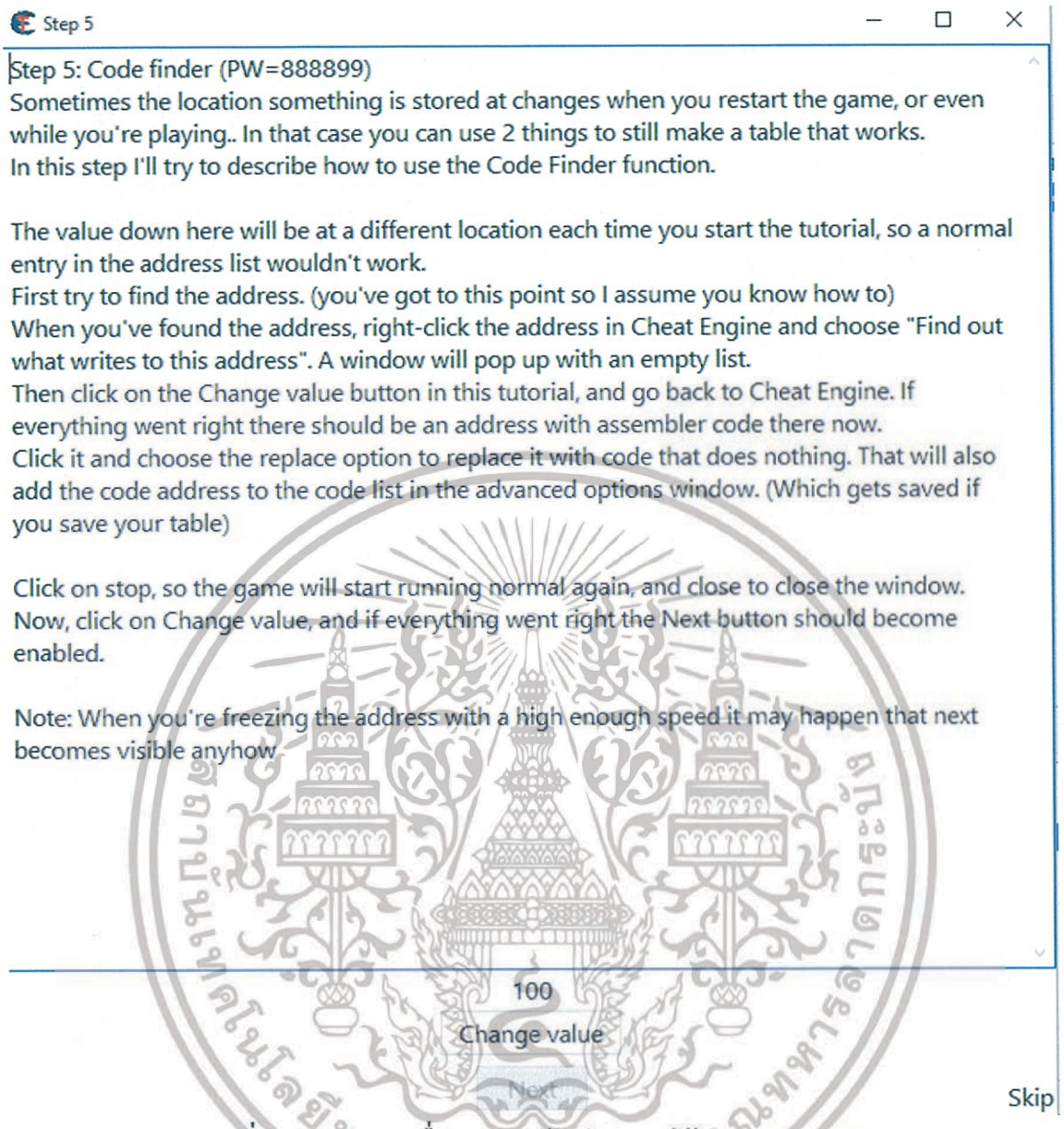
รูปที่ 4.1 การสแกนเพื่อแก้ไขค่าโดยตรง (Exact Value scanning)



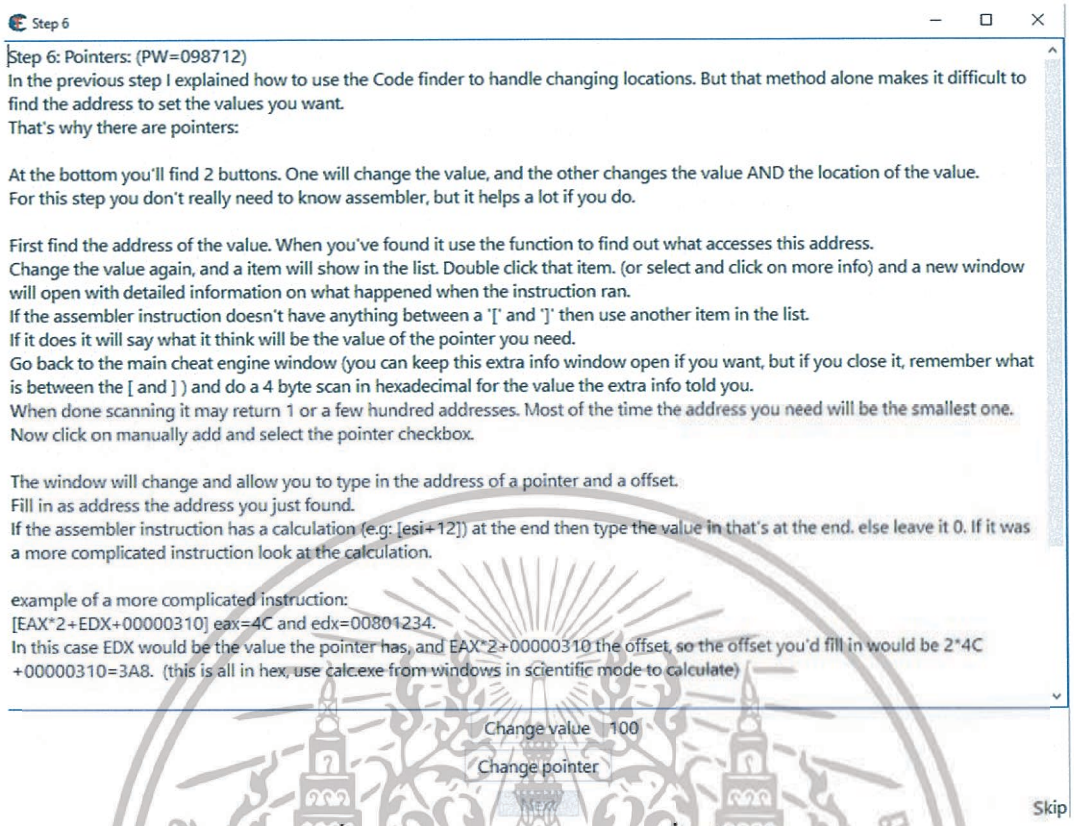
รูปที่ 4.2 การสแกนค่าเพื่อแก้ไขค่าที่ไม่รู้ค่าเริ่มต้น(Unknown initial value)



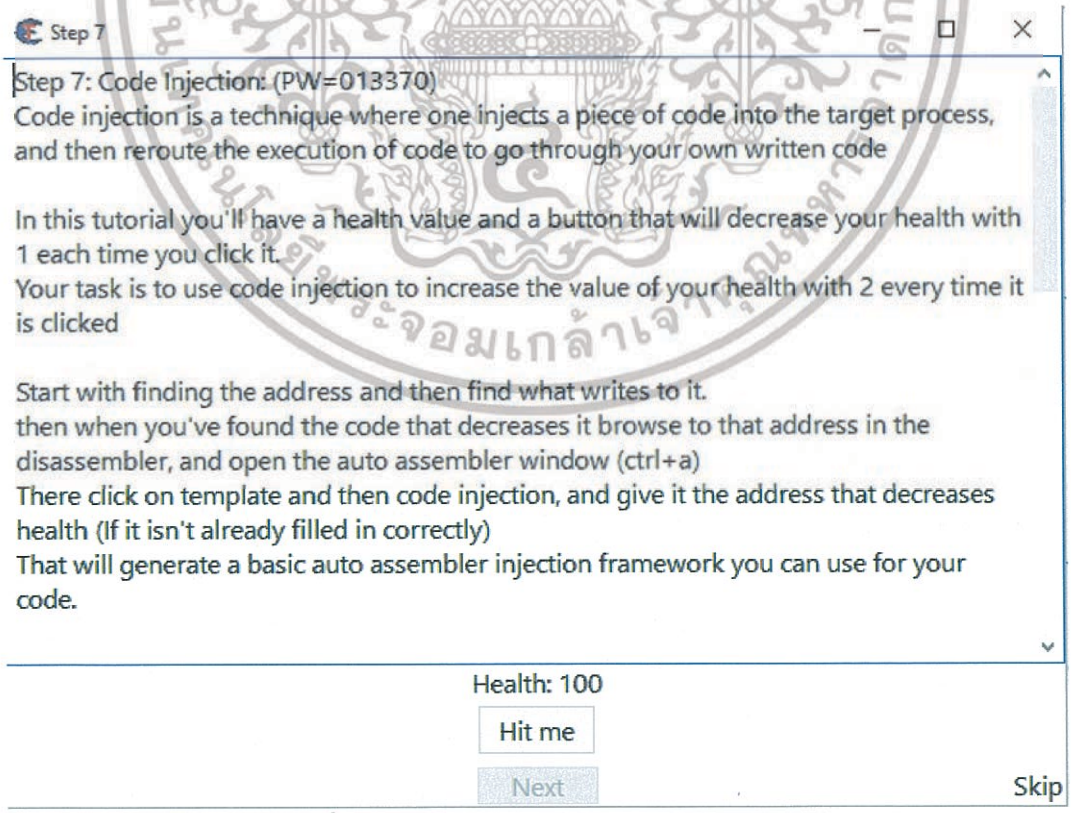
รูปที่ 4.3 การสแกนเพื่อแก้ไขค่า Float หรือ Double



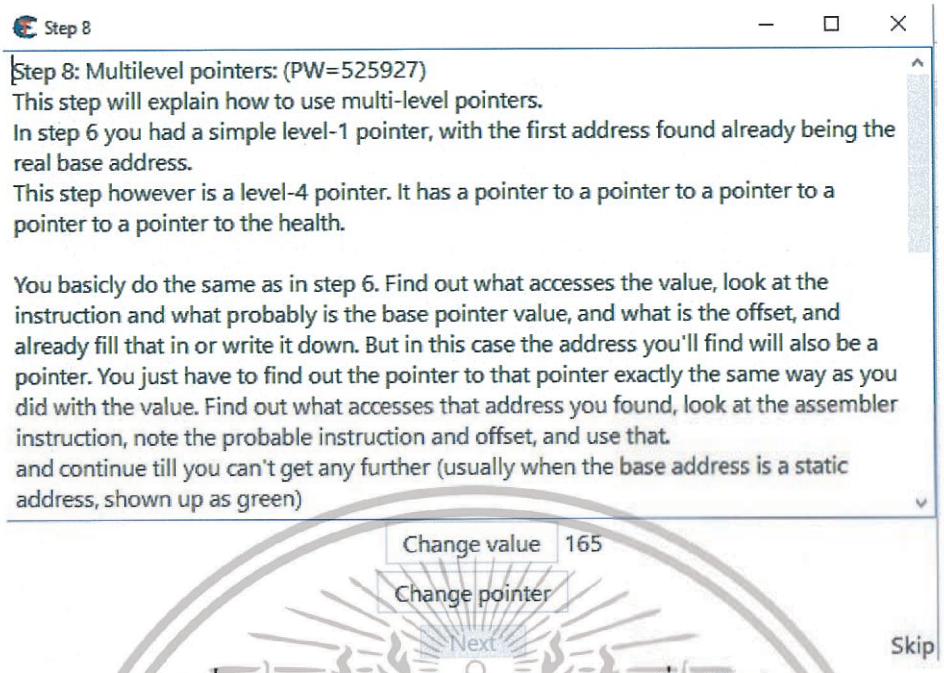
รูปที่ 4.4 การสแกนเพื่อหาและแก้ไขส่วนของโค้ด(Code finder)



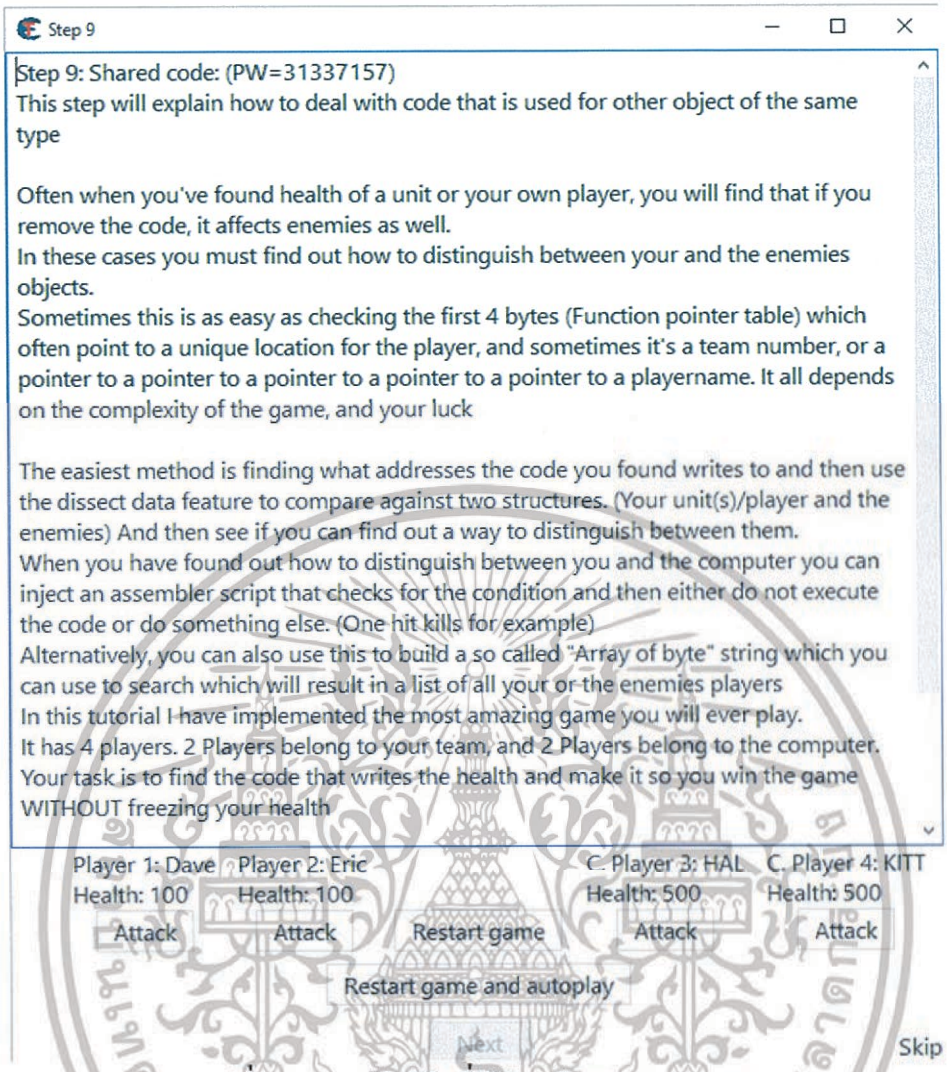
รูปที่ 4.5 การสแกนหา Pointer เพื่อแก้ไขค่า



รูปที่ 4.6 การแก้ไขโค้ดโดยใช้ Code Injection



รูปที่ 4.7 การสแกนหา Multilevel pointer เพื่อแก้ไขค่า



รูปที่ 4.8 การแก้ไขโค้ดที่ใช้ร่วมกัน(Shared code)

4.3.2 Plant vs Zombie

4.3.2.1 การแก้ไขค่าโดยตรงเพื่อเพิ่มจำนวน Sun

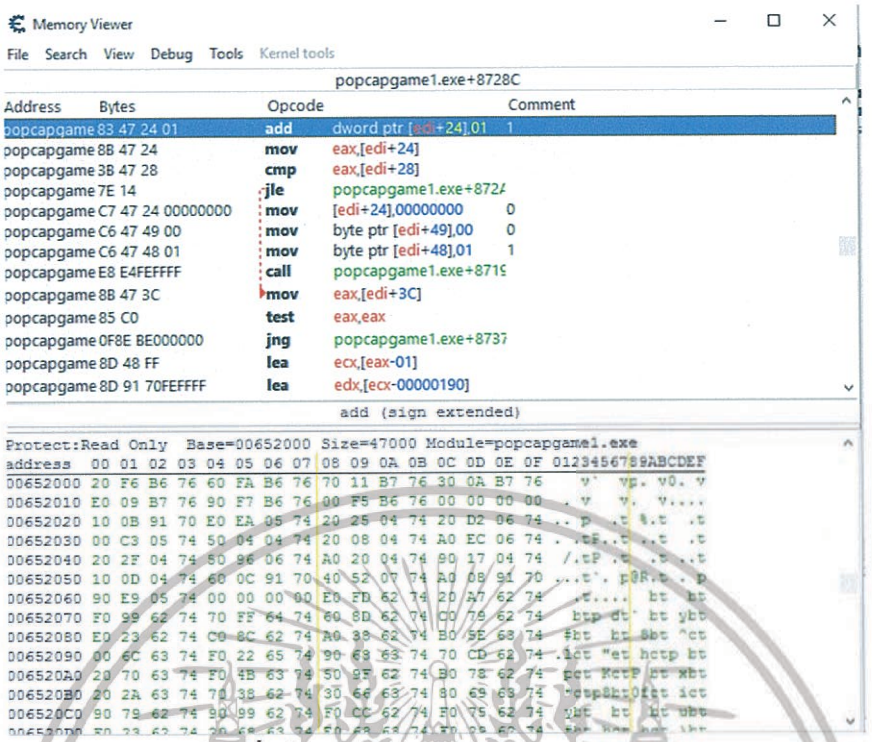
การแก้ไขจำนวน Sun ทำได้โดยการหา Address ของค่า Sun และทำการเพิ่มจำนวนตามที่ต้องการ แต่หลังจากเก็บ Sun เพิ่ม จะทำการลดค่าเหลือ 9990 คาดว่าน่าจะเป็นค่า Sun สูงสุดของเกม

รูปที่ 4.9 ก่อนทำการแก้ไขค่า Sun

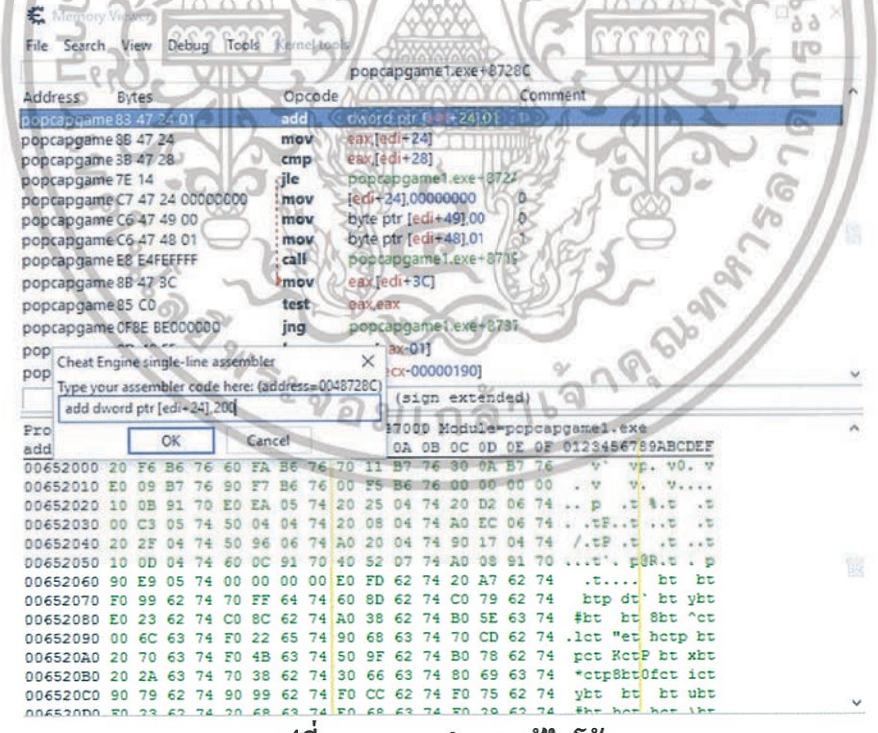
รูปที่ 4.10 หลังจากทำการแก้ไขค่า Sun เป็น 50000

4.3.2.2 การแก้ไขโค้ดเพื่อทำให้ต้นไม้ไม่ต้อง reload

การแก้ไขโค้ดเพื่อทำให้ต้นไม้ไม่มีการ Reload ทำให้สามารถลงต้นไม้ได้โดยไม่ต้องรอเวลา ทำได้โดยการค้นหา Address ส่วนที่ทำการ Reload และแก้ไขโค้ดในส่วนนั้น



รูปที่ 4.11 ส่วนของโค้ดก่อนทำการแก้ไข

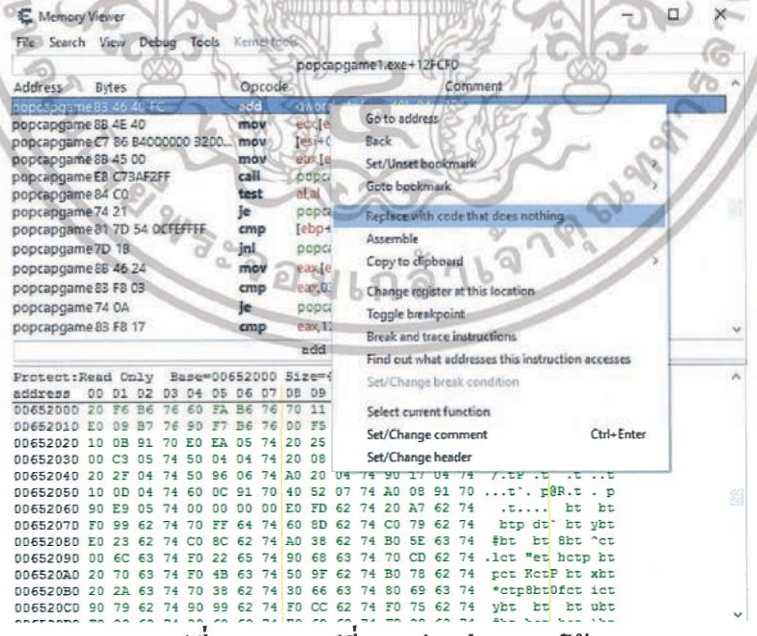


รูปที่ 4.12 ขณะทำการแก้ไขโค้ด

รูปที่ 4.13 เกมหลังจากทำการแก้ไข No reload Plant

4.3.2.3 การแก้ไขโค้ดเพื่อไม่ให้ลดเลือดของต้นไม้

วิธีการจะคล้ายกับการแก้ไขโค้ดเพื่อทำให้ต้นไม้ไม่ต้อง Reload คือทำการหา Address ส่วนที่ลดเลือดของต้นไม้ และเปลี่ยนเป็นโค้ดที่ไม่ทำอะไรเลย (Does Nothing) เพื่อไม่ให้ตัวโค้ดทำการลดเลือดของต้นไม้



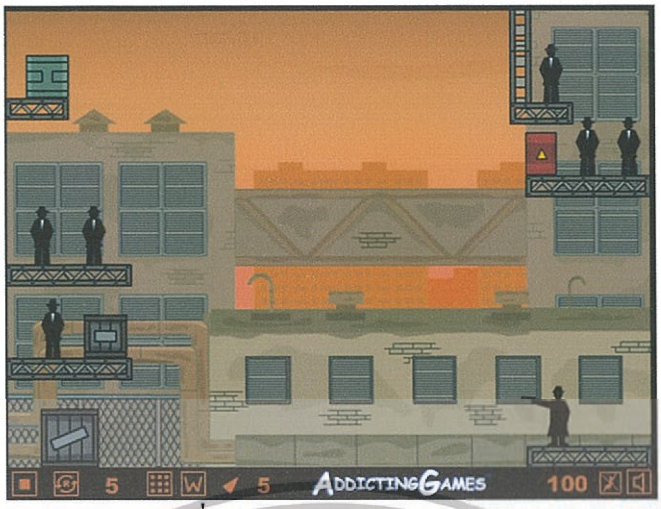
รูปที่ 4.14 การเปลี่ยนแปลงส่วนของโค้ด

รูปที่ 4.15 เกมหลังจากเปลี่ยนแปลงโค้ด

4.3.3 Flash-based game (เกม Flash)

Ricochet kills ทำการแก้ไขจำนวนกระสุนและแต้มคะแนน

รูปที่ 4.16 เกม Ricochet kills 3



รูปที่ 4.17 เกมก่อนทำการแก้ไข

รูปที่ 4.18 หลังจากทำการแก้ไขแต้มคะแนน

รูปที่ 4.19 หลังจากทำการแก้ไขจำนวนกระสุน

บทที่ 5

Copycat Go

รูปที่ 5.1 หน้าหลักของ Copycat Go

จากองค์ความรู้ที่ได้มาจากการทดลอง ได้นำมาสร้างเกมตัวอย่าง ชื่อว่า Copycat Go เพื่อแสดงถึงการประยุกต์ใช้องค์ความรู้ที่ได้ กับเกมออนไลน์ โดย Copycat Go นั้นมีรายละเอียดดังนี้

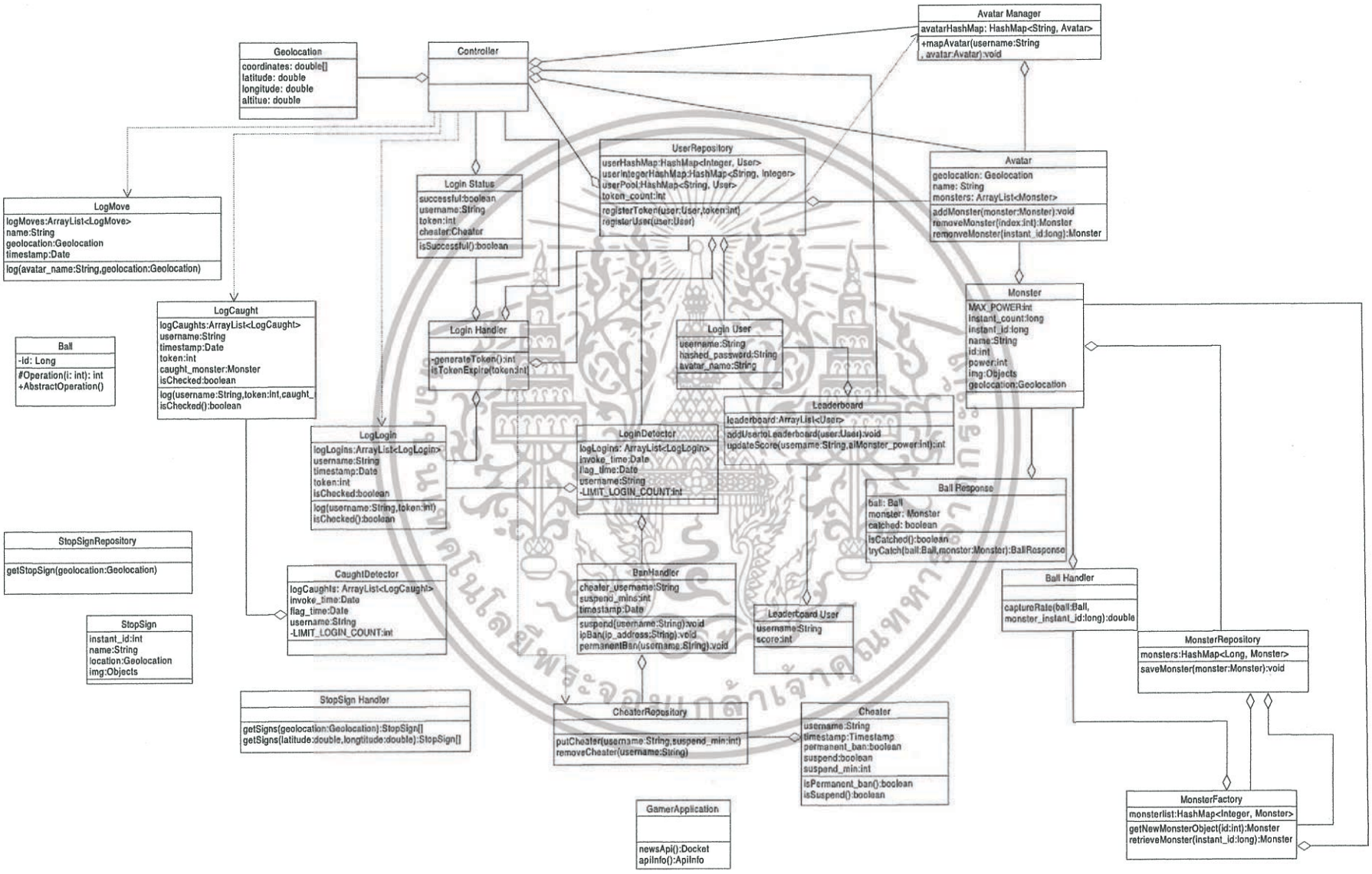
Server เกมที่ได้พัฒนาขึ้นเป็นรูปแบบ RESTful service โดยจะมี Front-end และ Back-end แยกกันทำงาน และติดต่อกันผ่านทาง HTTP Protocol โดยการรับและส่งข้อมูลจะอยู่ในรูปแบบของ JSON Object เป็นหลัก

5.1 การออกแบบระบบ

5.1.1 จุดประสงค์ของระบบ

ระบบ Copycat Go มีจุดประสงค์หลัก เพื่อท้าทายผู้เล่น ให้ทำการสะสมมอนสเตอร์ และนำออกไปต่อสู้ เพื่อก้าวไปมีอันดับใน Leaderboard

5.1.2 Class diagram

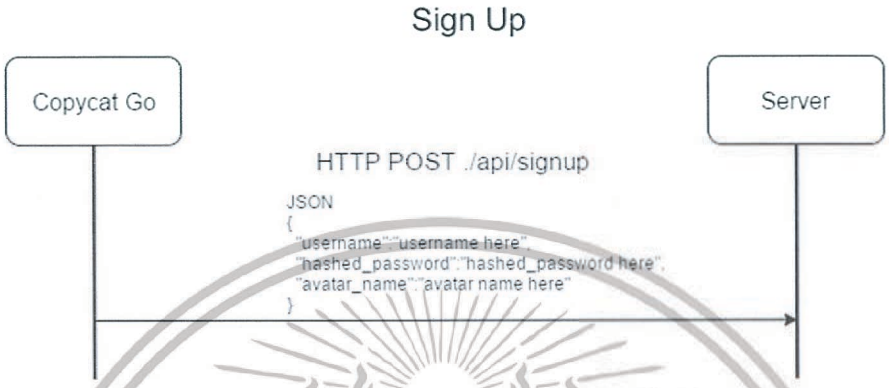


รูปที่ 5.2 Class diagram ของ Copycat Go

5.2 Copycat Go's features

5.2.1 Sign up

ขั้นตอนการ Sign up เมื่อเข้าไปยังหน้า ./api/signup แล้วจะพบกับหน้าต่างดังรูปที่ 5.2 จากนั้นกรอก Username, Password, และชื่อสำหรับตัวละคร

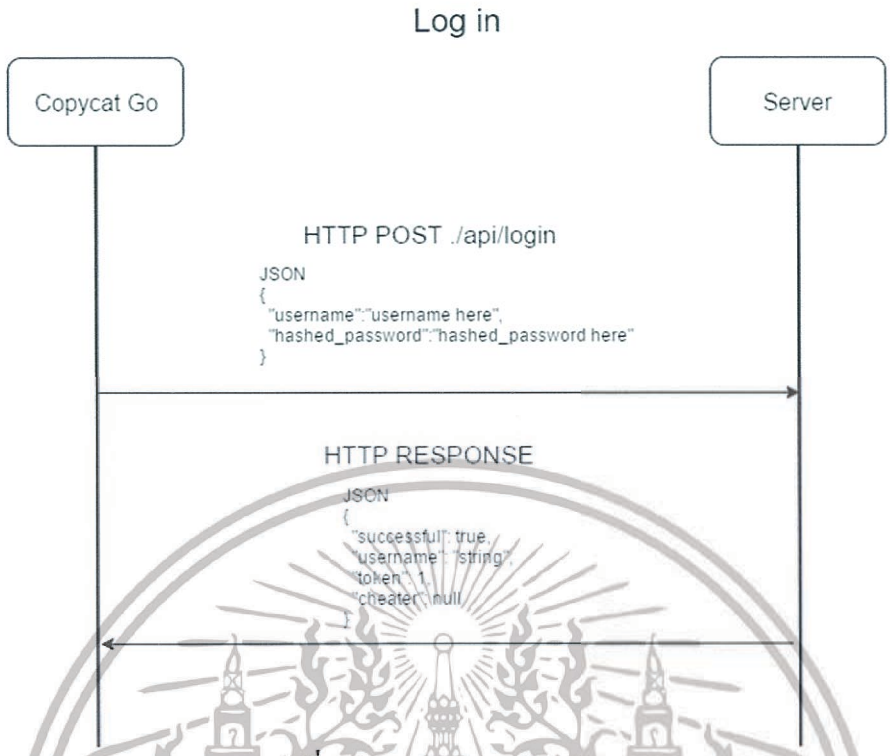


รูปที่ 5.3 การ Sign up กับ Server

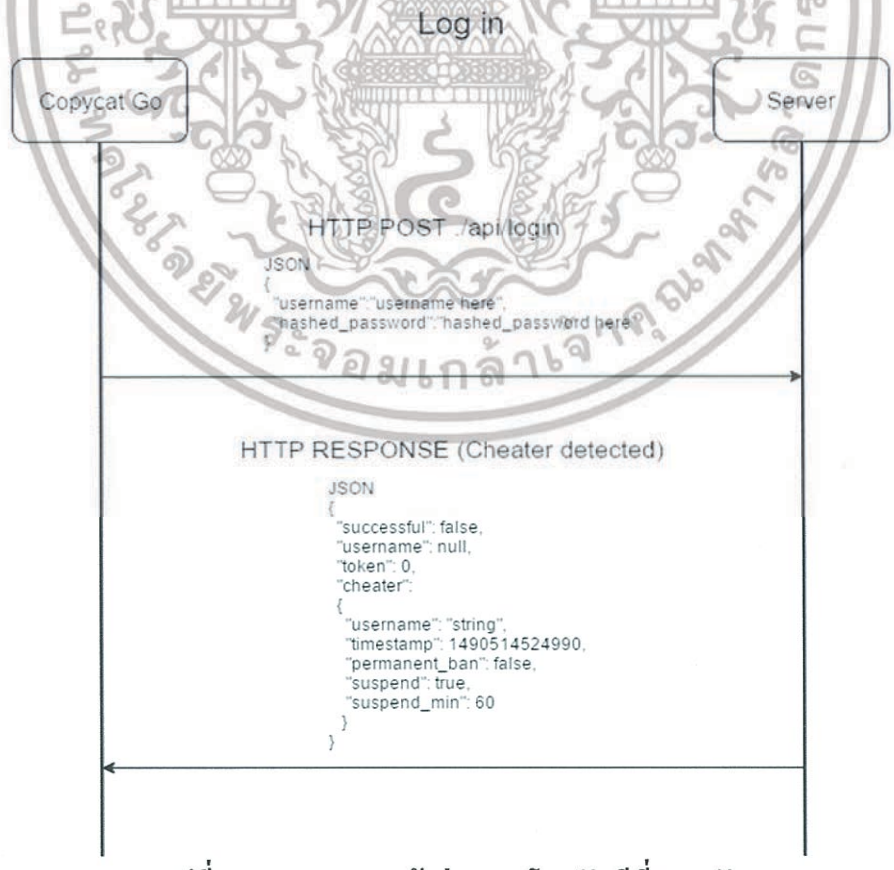
รูปที่ 5.4 หน้าต่าง Sign up

5.2.2 Login

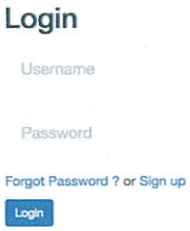
การ Log in สามารถทำได้หลังจากได้ Sign up แล้ว เมื่อเข้าที่หน้า ./api/login จะพบกับหน้าต่างให้กรอก Username และ Password ดังรูปที่ 5.5 เมื่อกรอกแล้ว กด Login ระบบจะย้ายหน้าไปสู่หน้าหลักของเกม และในกรณีที่ผู้เล่นที่ถูกระงับบัญชีผู้ใช้พยายามเข้าสู่ระบบ ระบบจะไม่อนุญาต และไม่สร้าง Token ใหม่ให้



รูปที่ 5.5 การ Log in เข้าสู่ระบบ



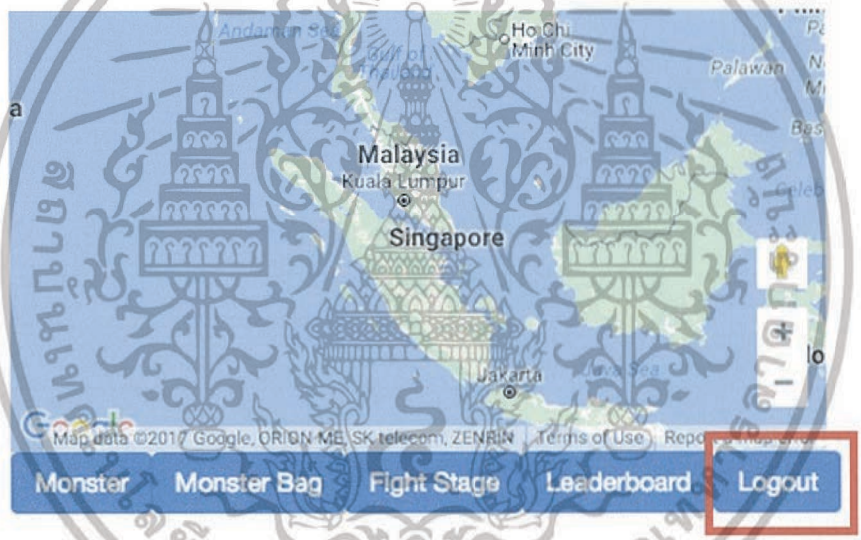
รูปที่ 5.6 การ Log in เข้าสู่ระบบ โดยบัญชีที่ถูกระงับ



รูปที่ 5.7 หน้าต่าง Log in

5.2.3 Logout

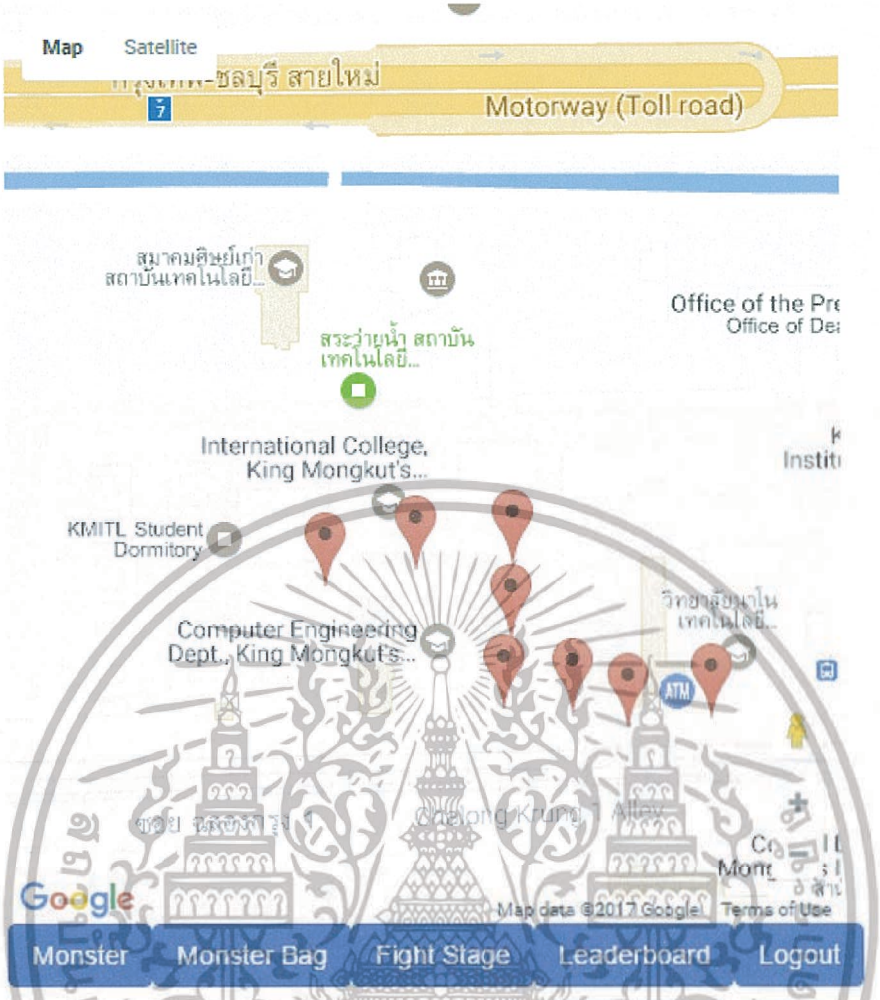
เมื่อผู้เล่นต้องการออกจากระบบ จะต้องกด Logout ที่หน้าหลัก เมื่อ Logout Browser จะลบ Cookie ที่ทำการบันทึกไว้ในเครื่อง



รูปที่ 5.8 ปุ่ม Log out

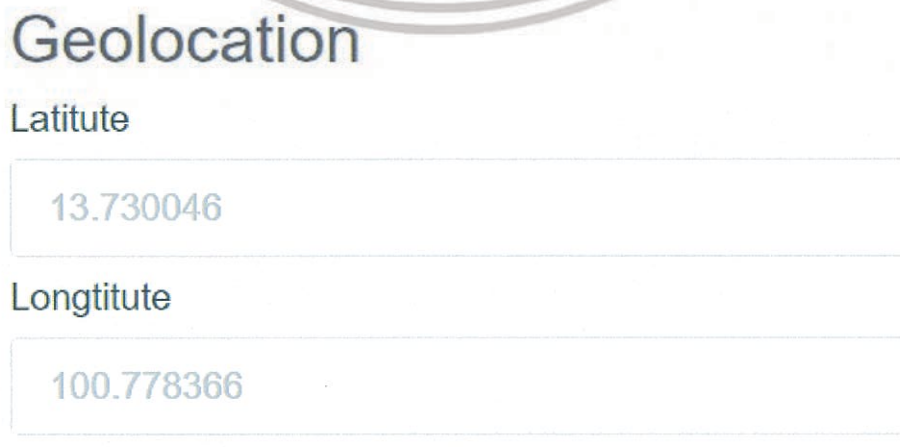
5.2.4 Move

เมื่อผู้ใช้ต้องการเคลื่อนที่ สามารถทำได้โดยการ Mark จุดบนแผนที่ จะเป็นการเคลื่อนที่ไปยังตำแหน่งนั้น ๆในทันที



รูปที่ 5.9 ตัวอย่างการเดิน (Move)

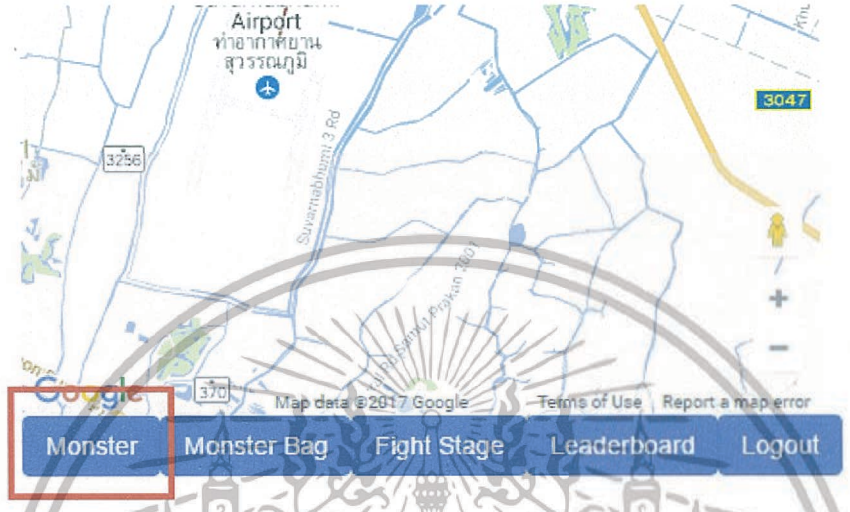
อีกช่องทางในการเคลื่อนที่ (Move) คือการป้อนค่าละติจูด และลองจิจูด ลงไปในช่องทั้งสอง และกด Submit แผนที่จะทำการย้ายไปบริเวณนั้น และจะมี Mark ลงไปที่ตำแหน่งนั้น



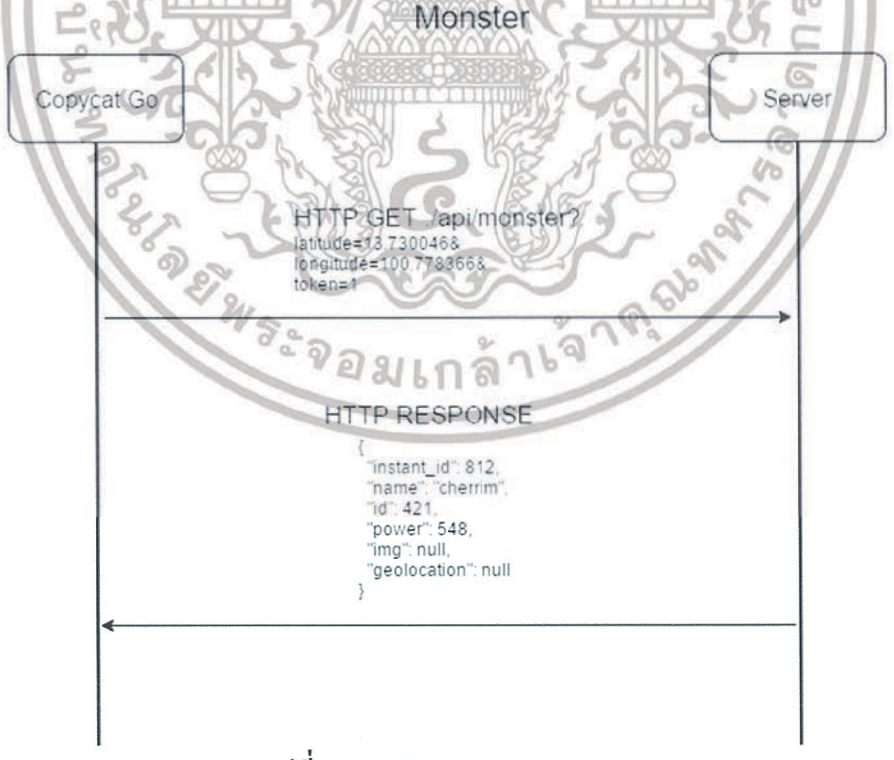
รูปที่ 5.10 หน้าต่างรับอินพุตเพื่อเคลื่อนที่ (Move)

5.2.5 Encounter a Wild Monster

เมื่อผู้เล่นต้องการจับ Monster จะต้องกดปุ่ม Monster ในหน้าหลัก จากนั้นจะย้ายหน้าไปยังหน้าที่แสดง ID และชื่อของ Monster ที่พบ ดังรูปที่ 5.9



รูปที่ 5.11 ปุ่ม Monster ในหน้าหลัก



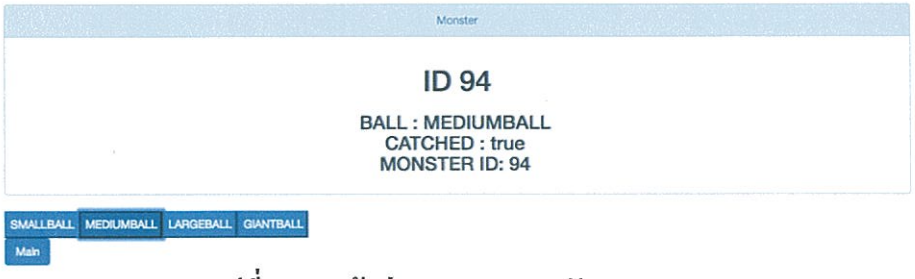
รูปที่ 5.12 การเจอ Wild Monster

รูปที่ 5.13 หน้าต่างแสดง ID ของ Monster ที่เจอ

5.2.6 Catch the Wild Monster

เมื่อผู้เล่นต้องการจับ Monster สามารถเลือกได้ระหว่าง SMALLBALL(1), MEDIUMBALL(1.25), LARGEBALL(1.5), GIANTBALL(2) โดยตัวเลือกจะมีความสามารถในการจับ Monster หรือ “โยนบอล” เรียงจากน้อยไปมาก มีมาตรฐานคือ SMALLBALL 1 เท่า, MEDIUMBALL 1.25 เท่า, LARGEBALL 1.5 เท่า, GIANTBALL 2 เท่า

รูปที่ 5.14 การจับ Monster

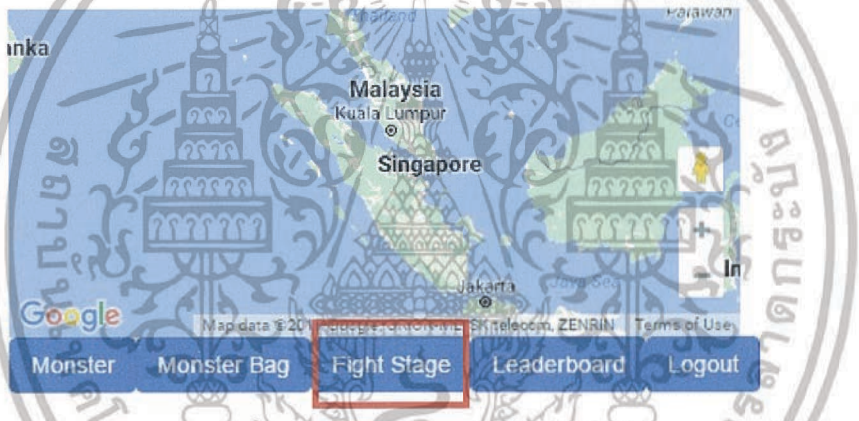


รูปที่ 5.15 หน้าต่างแสดงผลการจับ Monster

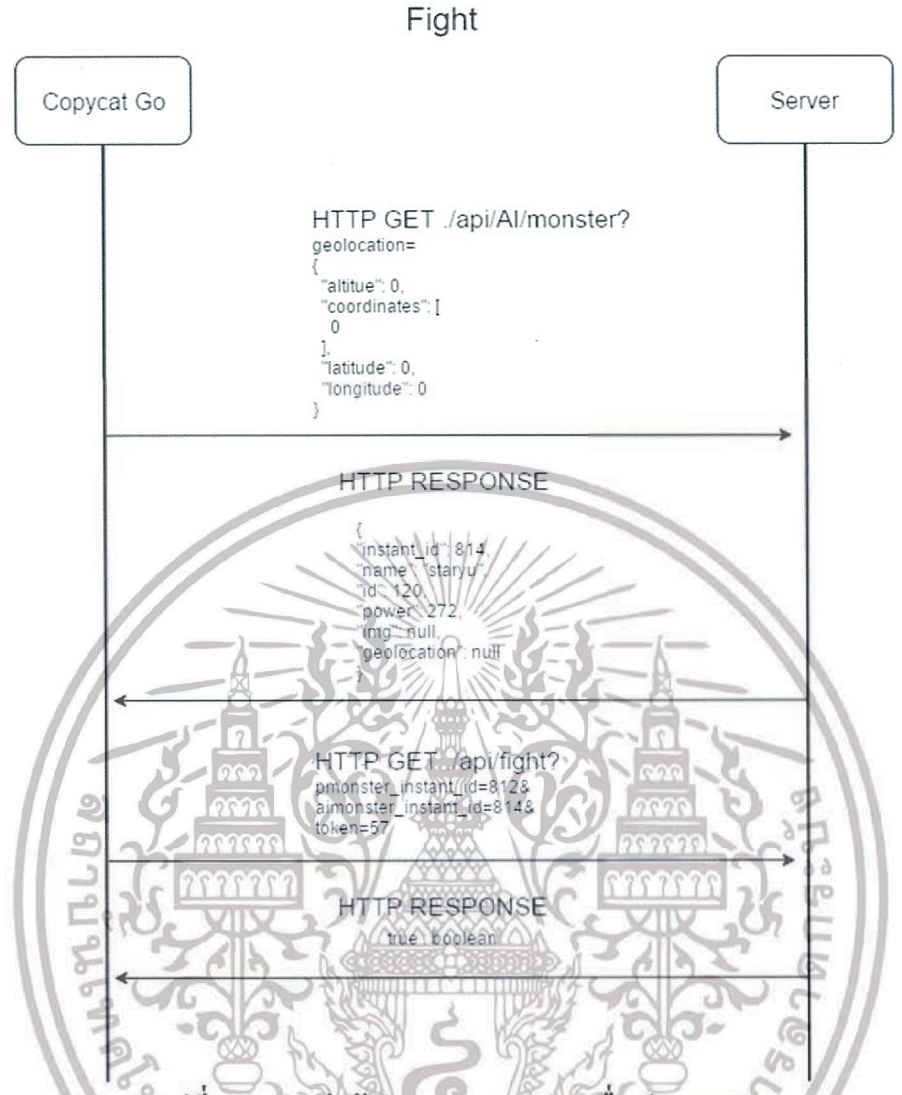
เมื่อทำการโยนบอลแล้วจะมีสถานะ Catched แสดงผล เมื่อจับได้จะแสดง CATCHED : true ในทางกลับกัน ถ้าจับไม่ได้จะแสดง CATCHED : FALSE

5.2.7 Fight the Wild Monster and Get Score

การต่อสู้กับ Wild Monster เพื่อสะสมคะแนน ทำได้โดยกดปุ่ม Fight Stage ดังรูปที่ 5.12



รูปที่ 5.16 ปุ่ม Fight Stage ในหน้าหลัก

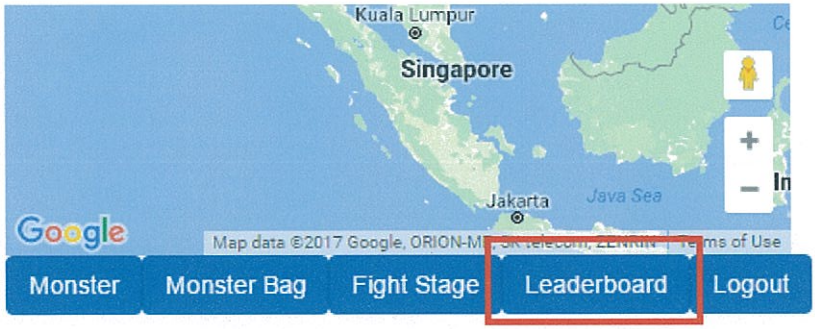


รูปที่ 5.17 การต่อสู้กับ Wild Monster เพื่อเก็บคะแนน

รูปที่ 5.18 หน้าต่างสำหรับการต่อสู้กับ Wild Monster

5.2.8 View and Maintain Leaderboard

เมื่อผู้เล่นต้องการดูคะแนนของตนเองเทียบกับผู้เล่นคนอื่น ๆ สามารถใช้ LeaderBoard เพื่อแสดงอันดับและคะแนนของผู้เล่นทุกคนได้



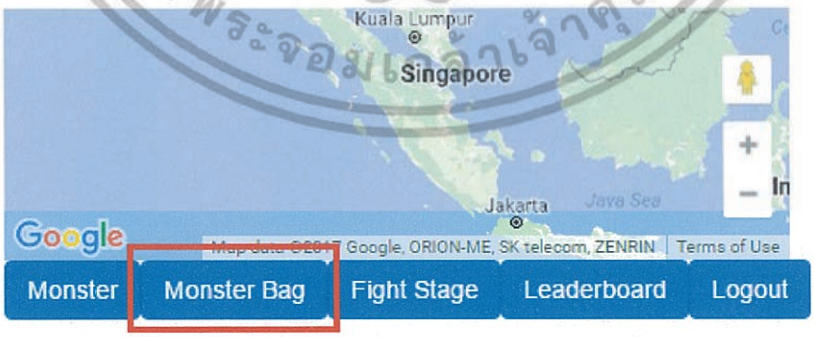
รูปที่ 5.19 ปุ่ม LeaderBoard ในหน้าหลัก

รูปที่ 5.20 หน้าต่าง Leaderboard แสดงคะแนนของผู้เล่น

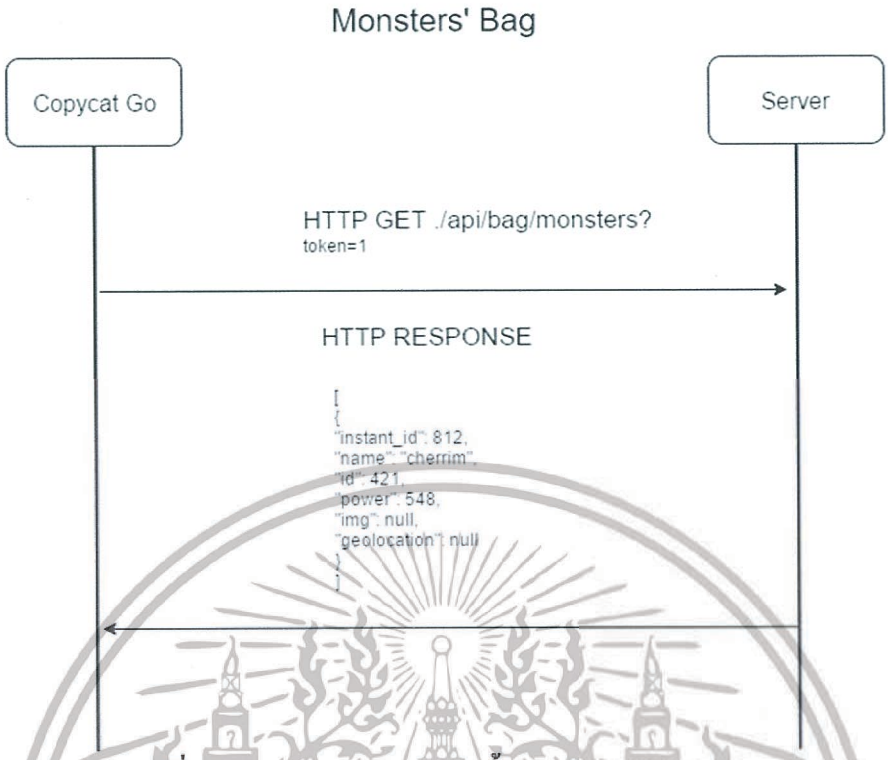
โดยอันดับจะเรียงจากผู้เล่นที่คะแนนมากอยู่ด้านบนสุด ลงมาด้านล่าง

5.2.9 View Monsters' Bag

เมื่อผู้เล่นต้องการดู Monster ทั้งหมดของตนเอง ให้ใช้ Monster Bag จะได้รายการ Monster ในกระเป๋าออกมา



รูปที่ 5.21 ปุ่ม Monster bag ในหน้าหลัก



รูปที่ 5.22 การเรียกดู Monster ทั้งหมดใน Monster Bag

รูปที่ 5.23 หน้าต่างแสดง Monster ใน Monster Bag

5.3 การป้องกันการโกงเกม Copycat Go

เนื่องจากการออกแบบเกมให้ Server เป็นฝั่งที่ทำการประมวลผลต่าง ๆทั้งหมด ดังนั้นการเปลี่ยนแปลงค่าในเกม (Client side) จึงทำได้ยากขึ้นจนถึงทำไม่ได้ ดังตัวอย่างต่อไปนี้

5.3.1 การบังคับเลือก Monster ที่จะพบจาก feature 5.1.5

ไม่สามารถทำได้ เนื่องจากการเรียก Monster ที่จะเจอเป็นส่วนของ Server และ Client มีหน้าที่เพียงรับข้อมูลมาแสดงเท่านั้น

5.3.2 การแก้ไข Monster ที่อยู่ใน Monster Bag

เนื่องจากข้อมูล Monster อยู่ที่ Server ดังนั้นการแก้ไข Monster จึงเป็นไปไม่ได้

5.3.3 การเปลี่ยนค่าพลังของ Monster ระหว่างการต่อสู้

การต่อสู้นั้นส่งข้อมูลเพียงแค่ Monster instant id จึงไม่มีการนำค่าจาก Client มาวิเคราะห์ ทำให้ไม่สามารถโกงได้

5.3.4 การเปลี่ยนแปลงคะแนนบน Leaderboard

เนื่องจากคะแนนเก็บอยู่ที่ Server ทำให้ไม่สามารถเปลี่ยนแปลงจาก Client ได้

และมีการพัฒนา Cheat Detector สำหรับการป้องกันการโกงโดยไม่เปลี่ยนแปลงค่าของตัวเกมอีกด้วย

5.4 Copycat Go's Cheat Detector

การตรวจจับการโกงของเกม Copycat Go ได้ทำตามแนวทางที่ได้จากผลการทดลองของบทที่ 4 แบ่งออกได้ดังนี้

5.4.1 ตรวจจับด้วยจำนวนครั้งการ Log in ใน 24 ชั่วโมง

จากผลการทดลองในบทที่ 4 เมื่อผู้ใช้ทำการ Log in อย่างน้อย 50 ครั้งภายใน 24 ชั่วโมงจะทำให้บัญชีนั้นถูกระงับการใช้งานเป็นระยะเวลาหนึ่ง

5.4.2 ตรวจจับด้วยจำนวนครั้งการจับ Monster ใน 24 ชั่วโมง

จากผลการทดลองในบทที่ 4 ลักษณะเดียวกับการตรวจจับด้วยจำนวนครั้งการ Log in เมื่อจับ Monster มากกว่า 500 ตัวภายใน 24 ชั่วโมง จะทำให้บัญชีผู้ใช้ถูกระงับการใช้งานเป็นระยะเวลาหนึ่ง

บทที่ 6
บทสรุปและข้อเสนอแนะ

6.1 การป้องกันการโกงเกม Pokémon Go

จากการทดลองพบว่า มาตรการการตรวจจับของ Pokémon Go หรือทาง Niantic ผู้ให้บริการนั้น ได้มีความหย่อนยานลง จากเมื่อช่วงที่เปิดเกมเดือนแรก คาดการว่า เนื่องจากจำนวนผู้เล่นน้อยลง และทางผู้พัฒนาต้องการเร่งมือ และโฟกัสไปที่การสร้างระบบใหม่ ๆ และทำระบบเก่าให้เสถียร จึงไม่มีความจำเป็นในการปราบปรามการโกงอย่างมีประสิทธิภาพ จะเห็นว่ามีการป้องกันเพียงแค่ตรวจสอบความถูกต้องของ Client เท่านั้น และระบบตรวจจับอัตโนมัติยังทำงานได้ไม่มีประสิทธิภาพดีพอ รวมทั้งการป้องกันยังสามารถได้รับการแก้ไขได้ ภายในระยะเวลาอันสั้น ถึงไม่มีการเกรงกลัวในการใช้งานโปรแกรมช่วยเล่น ต่างจากบางระบบ ที่มีการตรวจสอบ และระงับการใช้งานปัจจุบัน และในอนาคต

ระบบตรวจจับอัตโนมัติฝั่งผู้ให้บริการ ไม่สามารถป้องกันการเลียนแบบของโปรแกรมเล่นอัตโนมัติได้อย่างมีประสิทธิภาพ แต่การอัพเดทเกมอย่างสม่ำเสมอ ทำให้ความยากในการโกงนั้นได้ผลประโยชน์ไม่คุ้มกับเวลาที่เสียไป ทำให้การใช้งานโปรแกรมช่วยเล่นนั้น ส่งผลกระทบได้ไม่มากต่อผู้ให้บริการและผู้เล่นคนอื่น ๆ อีกทั้งลักษณะของเกม ณ เวลาทำการทดลอง เป็นการเล่นเกมเชิงสะสม ไม่ใช่เชิงแข่งขัน จึงส่งผลกระทบน้อยถึงน้อยมาก

เกม Copycat Go จึงถูกจัดทำขึ้นเพื่อเป็นตัวอย่างจากผลสรุปที่ได้จากการทดลองบอทโปเกมอนโก ได้แก่ ตรวจจับด้วยจำนวนครั้งการ Log in ใน 24 ชั่วโมง และ จำนวนครั้งการจับ Monster ใน 24 ชั่วโมง โดยตัวเกม Copycat Go มีการออกแบบให้ Server ประมวลผลต่าง ๆ จึงทำให้การเปลี่ยนแปลงค่าในเกมเป็นไปค่อนข้างยากไปจนถึงทำไม่ได้

6.2 กลไกการป้องกันการโกง เพิ่มเติม

เพิ่มความซับซ้อนที่กลไกของเกม และการรับ-ส่งข้อมูล, การตั้งชื่อตัวแปรที่ผู้ใช้สามารถเข้าถึงได้ หรือข้อมูล Response ที่ได้จาก Server ของเกม ให้เกิดความสับสน หรือมีความพยายามซ่อนกลไกของเกมจากผู้ใช้ ทำให้ยากต่อการสร้างโอกาสในการโกง

Client-Server Sync - การทำให้สถานะของเกมบน Client และ Server ตรงกันอยู่เสมอ โดยถ้ากลไกและ logic ต่าง ๆไม่ได้อยู่บนฝั่ง Client จะทำให้ควบคุมการโกงได้มีประสิทธิภาพมากขึ้น

6.3 แนวทางในอนาคต

จากข้อมูลผลการทดลองที่เก็บได้ สรุปการทดลองได้ว่า การ Login เข้าสู่ระบบ มากกว่า 50 ครั้ง จับ Pokémon มากกว่า 500 ตัว ขึ้นไปภายใน 1 วัน จะทำให้ระบบทำการ Soft ban ผู้เล่นอัตโนมัติ ซึ่งสามารถนำไปเป็น Input ของ โครงข่ายประสาท(Neutral networks) เพื่อทำระบบตรวจจับพฤติกรรมที่ผิดปกติของผู้เล่นแบบอัตโนมัติได้ และนอกจากนี้ ในกรณีที่มีพารามิเตอร์ที่เกี่ยวข้องจำนวนมาก และต้องการหาค่าพารามิเตอร์ที่มีผลกระทบกับพฤติกรรมที่ผิดปกติ สามารถประยุกต์ใช้ Symbolic regression เพื่อให้ได้ค่าสมการของข้อมูล ดังตัวอย่างงานวิจัยการระบุเมล็ดข้าวว่าเป็นเมล็ดข้าวแท้หรือเทียม จากคาต้าเซ็ตจำนวน 9 คาต้าเซ็ต แต่สมการที่ได้จาก Symbolic Regression เหลือคาต้าเซ็ตที่เกี่ยวข้องเพียง 3-4 คาต้าเซ็ต ทำให้สามารถระบุได้ว่า ปัจจัยอะไรบ้างที่มีผลต่อเมล็ดข้าว เช่นเดียวกันกับการตรวจสอบพฤติกรรมที่ผิดปกติของผู้เล่น หากมีค่าพารามิเตอร์จำนวนมาก เมื่อนำไปคำนวณด้วย Symbolic Regression ก็จะสามารถระบุได้ว่า ค่าพารามิเตอร์ใดมีความเกี่ยวข้องกับผู้เล่นที่มีพฤติกรรมที่ผิดปกติ และมีความเสี่ยงที่จะเป็นผู้ช่วยเล่น นอกจากนี้ เนื่องด้วยค่าความไม่คงที่ของการตรวจสอบ ทำให้สมการที่ได้อาจแปรผันไปตามช่วงเวลาอีกด้วย เมื่อมีการเปลี่ยนแปลง ก็สามารถคำนวณค่าสมการใหม่ เพื่อหาปัจจัยหรือ แนวทางการรับมือของผู้พัฒนาในช่วงเวลานั้น ๆได้

บรรณานุกรม

[1] Yuan Tian, Eric Chen, Xiaojun Ma, Shuo Chen, Xiao Wang, Patrick Tague. 2016. “Swords and shields: a study of mobile game hacks and existing defenses.” 386-397. **ACSAC '16 Proceedings of the 32nd Annual Conference on Computer Security Applications.** New York.

[2] S. Mitterhofer, C. Kruegel, E. Kirda and C. Platzer. May-June 2009. "Server-Side Bot Detection in Massively Multiplayer Online Games." in **IEEE Security & Privacy.** 7(3) : 29-36.

[3] Ying-Chieh Chen, Jing-Jang Hwang, Ronggong Song, G. Yee and L. Korba. 2005. "Online gaming cheating and security issue." **International Conference on Information Technology: Coding and Computing (ITCC'05) - Volume II.** 1 : 518-523

[4] M. Consalvo and I. S. Vazquez. 2014. "Cheating, Social Network Games and the Role of Platforms." 1687-1694. **2014 47th Hawaii International Conference on System Sciences.** Waikoloa. Hawaii.

[5] P. Laurens, R. F. Paige, P. J. Brooke and H. Chivers. 2007. "A Novel Approach to the Detection of Cheating in Multiplayer Online Games." 97-106. **12th IEEE International Conference on Engineering Complex Computer Systems (ICECCS 2007).** Auckland.

[6] Tom´a˘s Curda. 2014. “Analysis and detection of online game cheating software.” Bachelor Thesis of MASARYK UNIVERSITY.

[7] Pakorn Watanachaturaporn. 2016. “Identification of Rice Using Sysbolic Regression.” **8th International Conference on Information Technology and Electrical Engineering (ICTEE).** Yogyakarta. Indonesia

Conrad Pankoff. 2016. **Reverse Engineering: Pokemon GO.** [Online].Available : https://www.fknsrs.biz/blog/reverse-engineering-pokemon-go.html.

__isitin__. 2016. **Guide to Pokemon Go Server Responses.** [Online].Available : https://www.reddit.com/r/pokemongodev/comments/4svl1o/guide_to_pokemon_go_server_responses/

Shape Security. 2016. **Pokémon Go API – A Closer Look at Automated Attacks.** [Online].Available : https://blog.shapesecurity.com/2016/08/16/pokemon-go-api/

Eaton Z. 2016. **Reverse engineering and removing Pokémon GO's certificate pinning.** [Online].Available : https://eaton-works.com/2016/07/

Aaron Ardiri. 2016. **Pokémon GO - REVISITING THE "HACKING" SCENE (PART 1).** [Online].Available : http://ardiri.com/blog/pokemon_go_revisiting_the_hacking_scene_part_1