รายงาน โปรแกรม Havij

เสนอ

รศ.ดร.จักรชัย โสอินทร์

สมาชิก

นางสาวณัฐชา	รักษาสัตย์	563020207-8
นางสาวธิดารัตน์	หงษา	573021144-2
นางสาวนัยนา	เศษฤทธิ์	573021147-6
นางสาวพัชริญา	วันสาสืบ	573021153-1
นางสาวมาริษา	วาระสิทธิ์	573021160-4

สาขาเทคโนโลยีสารสนเทศและการสื่อสาร

รายงานนี้เป็นส่วนหนึ่งของการศึกษาวิชา 322376 Information and Communication Technology Security

ภาคเรียนที่ 1 ปีการศึกษา 2559

คณะวิทยาศาสตร์ มหาวิทยาลัยขอนแก่น

คำนำ

รายงานเล่มนี้เป็นส่วนหนึ่งของวิชา 322376 Information and Communication Technology Security ซึ่งคณะผู้จัดทำได้จัดทำเกี่ยวกับโปรแกรม Havij เนื่องจากเว็บไซต์ในปัจจุบัน มีความไม่ปลอดภัยจากผู้คุกคามที่ไม่หวังดี คอยสร้างปัญหาจากหลากหลายช่องทางผ่านอินเตอร์เน็ต อาทิเช่น การดักจับข้อมูลผ่านเว็บไซต์ที่มีช่องโหว่ ก็ยังสร้างความเสียหายมหาศาล ทำให้เราต้องหาทางลดความเสี่ยงด้วยความรู้จากการศึกษาหารูปแบบและช่องทางที่เหล่าผู้ไม่หวังดีนำเอามาใช้ และด้วยเหตุจึงเป็นการศึกษา เพื่อป้องกันอันตรายจากผู้ไม่หวังดีที่หวังจะมาเสาะแสวงหาข้อมูลของเราไปเป็นทางเลือกอีกทาง ที่จะทำให้เว็บไซต์ของเราได้รับการแก้ไขและมีประสิทธิภาพจากข้อบกพร่องที่เราค้นพบไปพัฒนาระบบได้อย่างสมบูรณ์

ขอขอบคุณ รศ.ดร. จักรชัย โสอินทร์ และทุกคนที่มีส่วนร่วมในการทำรายงานเล่มนี้ ที่คอยช่วยเหลือและให้คำแนะนำสำหรับปัญหาต่างๆตั้งแต่ต้นจนงานมีความสำเร็จสมบูรณ์ และหวังว่ารายงานนี้จะช่วยให้ผู้ที่พัฒนาเว็บไซต์และผู้ที่ต้องการศึกษาข้อมูลในด้านนี้นำไปใช้ประโยชน์ได้สูงสุดในทางที่ดี หรือศึกษาต่อยอดงานอื่นๆได้ หากมีข้อผิดพลาดผู้จัดทำก็ขออภัยมา ณ ที่นี้ด้วย

หลักการและเหตุผล

ในปัจจุบันนี้ เทคโนโลยีมีบทบาทในชีวิตประจำมากขึ้น รวมถึงการโจมตีข้อมูลที่มีหลากหลายรูปแบบ เช่น การ Hack ข้อมูลของระบบราชการหรือองค์กรต่างๆ ทำให้ข้อมูลที่สำคัญสูญหายได้ ผู้จัดทำจึงเล็งเห็นว่าเป็นปัญหาสำคัญต่อการใช้เทคโนโลยี ทำอย่างเช่นไร จึงจะมีการรักษาความปลอดภัยในการเข้าถึงระบบและการทำงานนั้นๆ ดังนั้นผู้จัดทำจึงได้ศึกษาโปรแกรมที่สามารถโจมตีฐานข้อมูล เพื่อนำมาปรับปรุงเว็บไซต์ให้มีประสิทธิภาพมากยิ่งขึ้น จากการโจมตีโดยหาช่องโหว่ของฐานข้อมูล

วัตถุประสงค์ของโปรแกรม

1. เพื่อศึกษาการโจมตีฐานข้อมูลโดยใช้โปรแกรม havij
2. เพื่อศึกษาวิธีการป้องกันการโจมตีเว็บไซต์ที่มีช่องโหว่ในการเข้าใช้งาน

ขอบเขตการศึกษา

- Hack ได้เฉพาะเว็บที่เป็นแอปพลิเคชั่น มี ID ที่ใช้ดึงข้อมูลคอลัมน์ใน database

กลุ่มผู้ใช้งาน

- ผู้ที่ต้องการตรวจสอบช่องโหว่ของเว็บไซต์ เพื่อป้องกันการโดนโจมตี

ประโยชน์ที่คาดว่าจะได้รับ

1. สามารถนำไปใช้ปรับปรุงกับเว็บไซต์ที่เราพัฒนาได้
2. เพื่อทราบถึงช่องโหว่หรือความไม่สมบูรณ์ของเว็บไซต์นั้นๆได้

เอกสารที่เกี่ยวข้อง

1 SQL Injection

เป็นเทคนิคที่ใช้ประโยชน์จากส่งคำสั่ง SQL ผ่านทางเว็บแอพพลิเคชันเพื่อไปโจมตีระบบฐานข้อมูลหลังบ้าน โดยอาศัยช่องโหว่ของการใส่ข้อมูล input ของผู้ใช้ที่สามารถตรวจสอบรูปแบบการโจมตีได้อย่างจำกัด แฮ็คเกอร์รู้ดีว่านักเขียนโปรแกรมจะนำข้อมูลที่ผู้ใช้ input ลงไป ไปใช้เป็นส่วนหนึ่งของคำสั่ง SQL เพื่อส่งไปยังระบบฐานข้อมูล จึงได้แอบฝังคำสั่ง SQL บางอย่างลงไปใน input เหล่านั้นด้วย ส่งผลให้แฮ็คเกอร์สามารถดึงข้อมูลหรือเปลี่ยนแปลงแก้ไขข้อมูลในระบบฐานข้อมูลตามคำสั่ง SQL ที่แอบฝังลงไปได้ทันที ยกตัวอย่างง่ายๆที่พบเห็นบ่อยๆ คือ “OR 1=1” ที่นิยมใช้เพื่อบายพาสการพิสูจน์ตัวตน ปกติแล้วหน้าพิสูจน์ตัวตนจะมีช่องให้ใส่ชื่อผู้ใช้และรหัสผ่าน ซึ่งนักเขียนโปรแกรมก็จะนำข้อมูลที่ผู้ใช้กรอกลงไป ไปตรวจสอบกับระบบฐานข้อมูลโดยใช้คำสั่ง

SELECT * FROM authen_db WHERE username=‘suthee’ and password=‘12345678’;

เพื่อเช็คดูว่าในฐานข้อมูลการพิสูจน์ตัวตน (authen_db) มีชื่อผู้ใช้และรหัสผ่านตรงตามที่ผู้ใช้กรอกลงไปหรือไม่ ซึ่งเมื่อแฮ็คเกอร์รู้ว่าต้องมีการนำข้อมูลที่ผู้ใช้กรอกลงไป (ในที่นี้คือ suthee และ 12345678) ส่งไปยังระบบฐานข้อมูลโดยตรง จึงได้แอบฝังคำสั่ง SQL ลงไปเพื่อหลีกเลี่ยงการตรวจสอบ คือ การใส่ชื่อผู้ใช้เป็น “admin” และรหัสผ่านเป็น “‘ OR ‘1’=‘1” ส่งผลให้คำสั่ง SQL ที่ใช้ตรวจสอบเพื่อพิสูจน์ตัวตนก็จะกลายเป็น

SELECT * FROM authen_db WHERE username=‘admin’ and password=‘’ OR ‘1’=‘1’;

ผลลัพธ์ที่ได้ คือ แฮ็คเกอร์สามารถลงชื่อเข้าใช้เป็น “admin” ได้ทันที เนื่องจากด้านหลังมีนิพจน์ OR 1=1 ทำให้คำสั่ง SQL ดังกล่าวเป็นจริงเสมอ นอกจากการบายพาสการพิสูจน์ตัวตนแล้ว SQL Injection ยังสามารถดึงข้อมูล, เปลี่ยนแปลงแก้ไข, ลบข้อมูล หรือทำลายฐานข้อมูลทั้งหมดขึ้นอยู่กับคำสั่ง SQL ที่แอบฝังลงไปได้เช่นกัน

2. Havij

Havij ถือเป็น Sql injection tool (โปรแกรมแฮ็ก) ตัวหนึ่งที่ใช้ สำหรับเจาะฐานข้อมูล web application โดยใช้หลักการ sql injection อาศัยช่องโหว่ของฐานข้อมูล เพื่อเข้าไปใส่คำสั่ง sql จาก URL เป็นช่องโหว่ของ php นิยมใช้งานกันอย่างแพร่หลายในไทย และก็ทราบเป็นที่แน่นอนว่า "มันไม่ได้เกิดประโยชน์อะไรให้แก่ตัวเราเลยซักนิดเดียว"

3. ความต้องการพื้นฐานและความมั่นคงปลอดภัยของการป้องกันเว็บไซต์

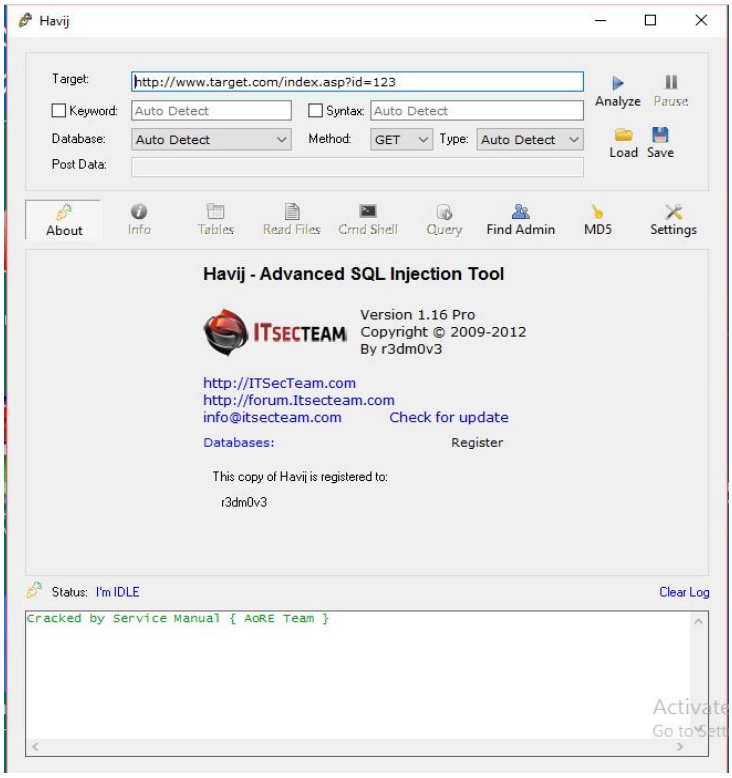
เว็บไซต์โดยทั่วไปที่ติดต่อซื้อขายสินค้าหรือให้บริการแก่ผู้ใช้บริการนำเว็บไซต์มาฝาก เว็บไซต์ที่เว็บโฮสติง (Web Hosting) เป็นบริการให้เช่าพื้นโดยมีการดูแลระบบความปลอดภัย ในการปรับปรุงเว็บเซิร์ฟเวอร์ (Web Server) ทำให้มีความปลอดภัยส่วนหนึ่ง ส่วนเว็บไซต์ เจ้าของต้องบริหารจัดการเว็บไซต์ให้มีความปลอดภัยได้แก่การปรับแต่งเว็บไซต์ส่วนที่เสี่ยง ก่อให้เกิดอันตรายได้หรือให้ผู้อื่นจัดการดูแลเพื่อให้เกิดความน่าเชื่อของลูกค้าที่มาใช้บริการกับ เว็บไซต์สามารถการันตีได้ข้อมูลเกี่ยวกับลูกค้าทั้งหมดจะไม่รั่วไหลแต่เว็บไซต์บนอินเตอร์เน็ต ยังอยู่ในอันตรายเนื่องจาก hacker ได้หาช่องทางในการโจมตีเว็บไซต์โดยใส่ SQL Query String เข้าไปบางส่วนของเว็บเพ็จบนเว็บไซต์ที่มีการป้องกัน ไม่ให้มีการเข้าถึงข้อมูล ข้อมูลสามารถถูกแสดง ตามที่ hacker ต้องการ อย่างไรก็ตามการรักษาความปลอดภัยของเว็บไซต์เป็นเป้าหมายหลักที่ต้องพิจารณา เป็นเรื่องแรกเพื่อให้ลูกค้าที่เข้ามาใช้บริการมีความมั่นใจและมีความน่าเชื่อถือเว็บไซตำโดย การตรวจสอบ SQL Query String กับฐานข้อมูลการโจมตีเว็บไซต์ที่ต้องปรับปรุงข้อมูลเป็น ปัจจุบันสามารถป้องกันได้ส่วนหนึ่งและตรวจสอบโมดูลที่ติดตั้งเพิ่มเพราะบางบรรทัด Source Code ของโมดูลมีช่องโหว่ทำให้ hacker โจมตีเว็บไซต์จากส่วนนี้ได้

งานวิจัยที่เกี่ยวข้อง

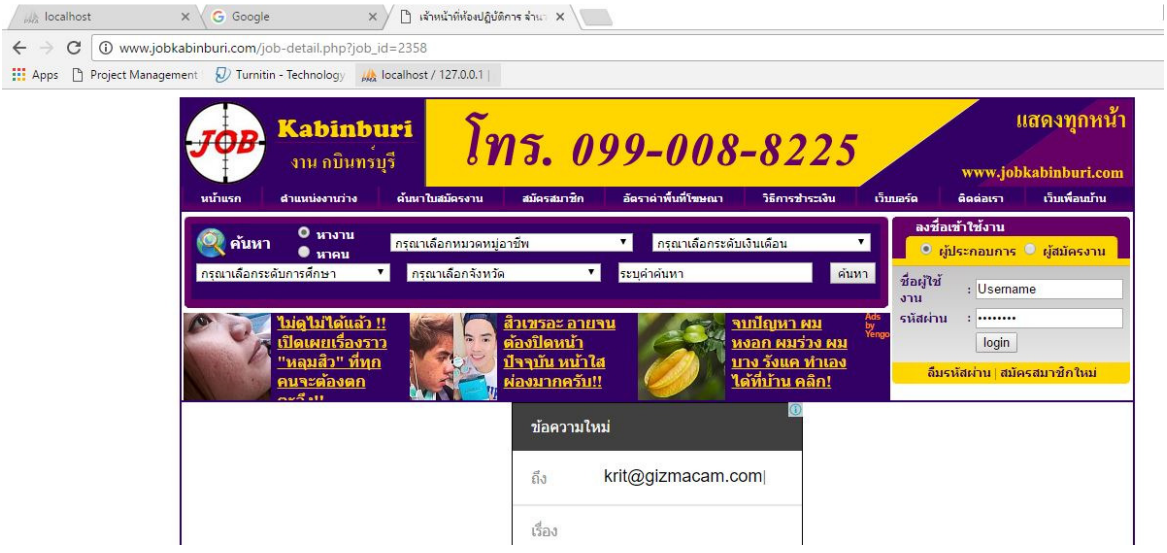
วิทยานิพนธ์ การพัฒนาเทคนิคการป้องกัน และตรวจจับ

วิทยานิพนธ์เล่มนี้ได้นำเสนอระบบการตรวจจับการโจมตีแบบ SQL injection โดยการเปรียบชุดคำสั่ง SQL query ที่พัฒนาโดยโปรแกรมเมอร์ กับชุดคำสั่ง SQL query จากผู้ใช้งาน ซึ่งระบบจะทำการเปรียบเทียบจำนวน Single quote จากโปรแกรมเมอร์กับจำนวน Single quote จากผู้ใช้งาน เพื่อตรวจสอบว่าคำสั่ง SQL query จากผู้ใช้งานเป็น SQL Injection หรือไม่ เพื่อเป็นการป้องกันSQL query ที่เป็น SQL Injection ไม่ให้เข้าถึงฐานข้อมูล ซึ่งผลการทดลอง ได้แสดงให้เห็นว่าระบบสามารถแก้ปัญหาเรื่อง SQL Injection รูปแบบใหม่ๆที่เกิดขึ้นจาก Hacker

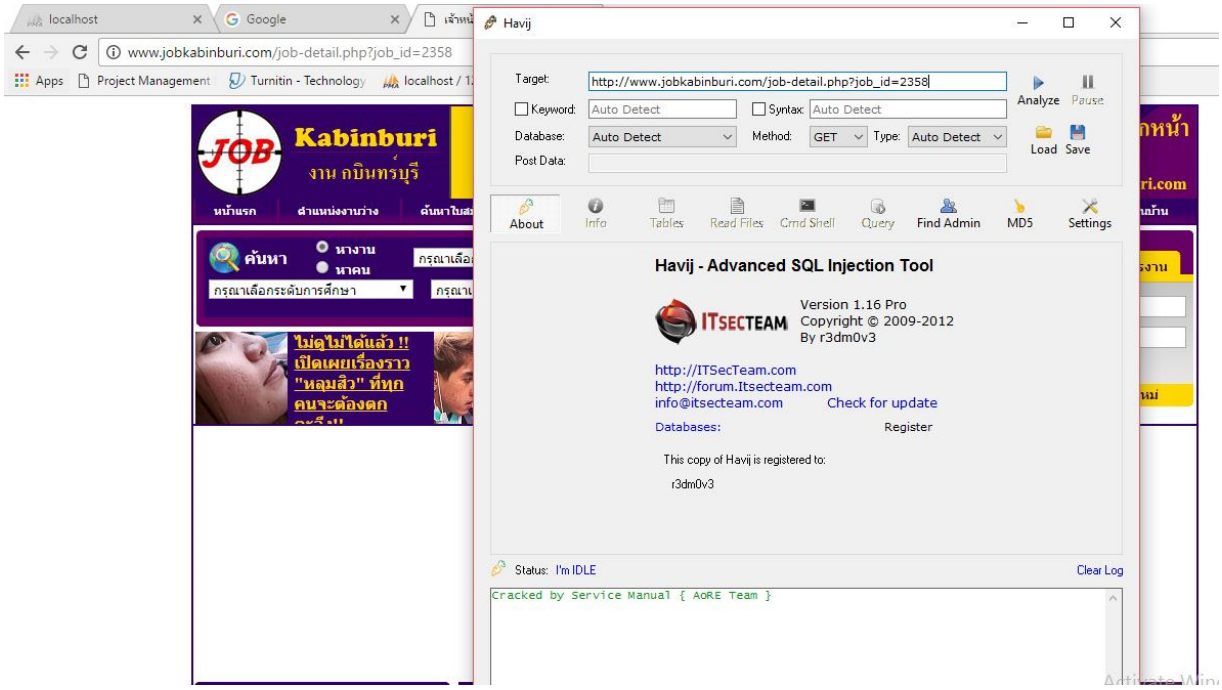
หน้าจอการทำงานทั้งหมดของโปรแกรม



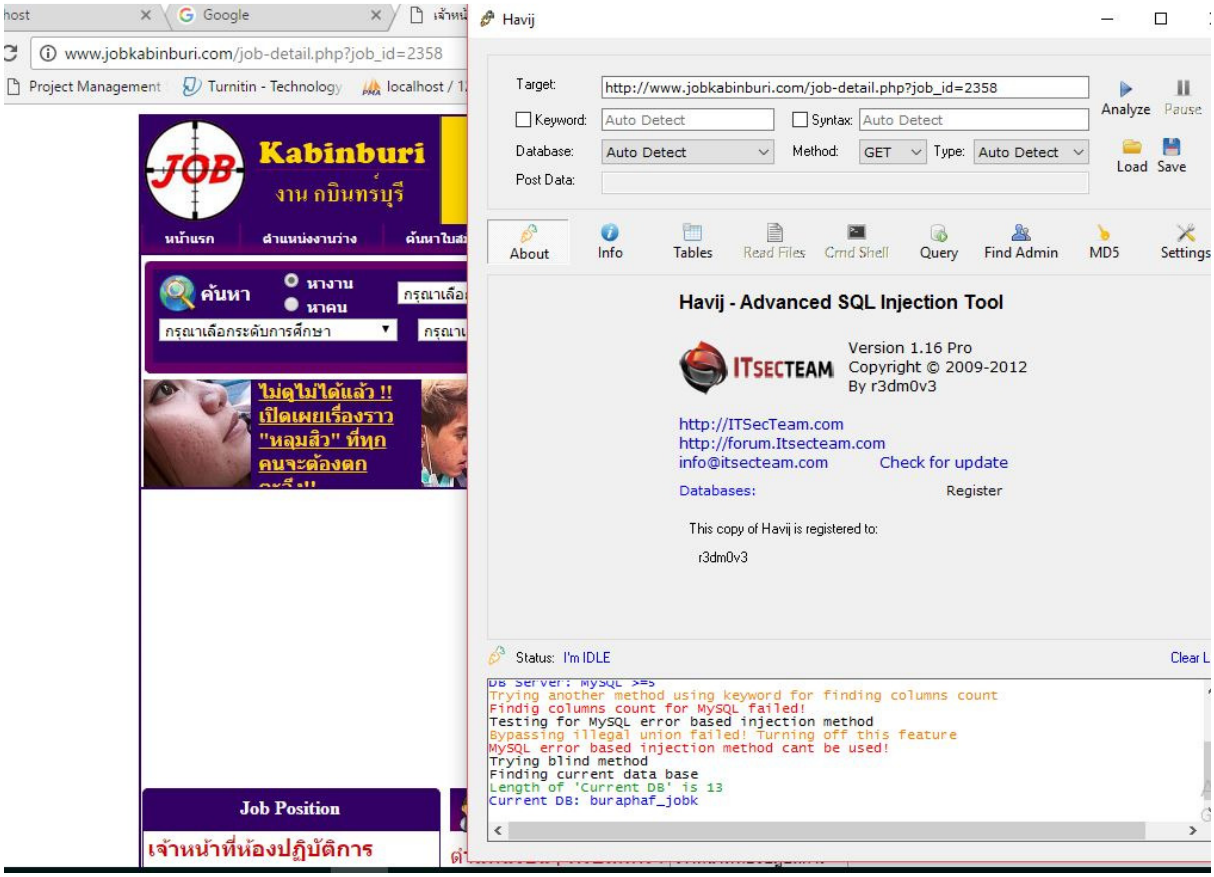
รูปภาพที่ 1 หน้าตาโปรแกรม



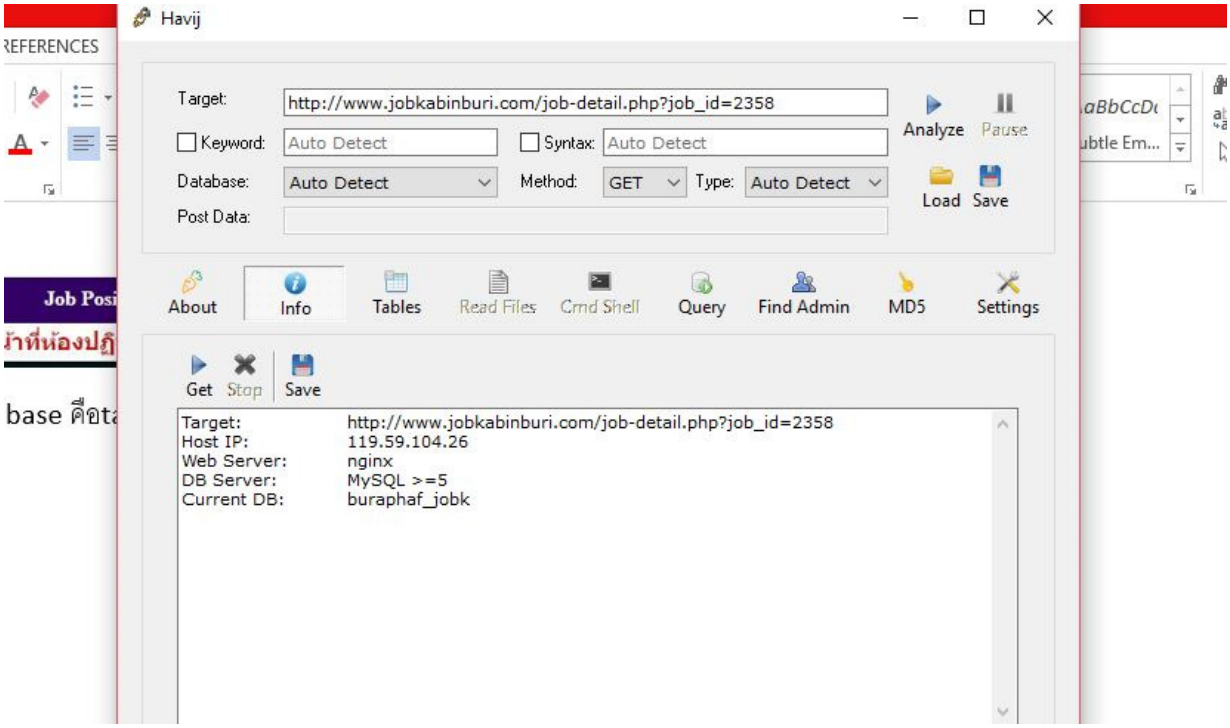
รูปภาพที่ 2 เว็บที่เป็นแอปพลิเคชั่น มี id ที่ใช้ดึงข้อมูลคอลัมน์ใน database



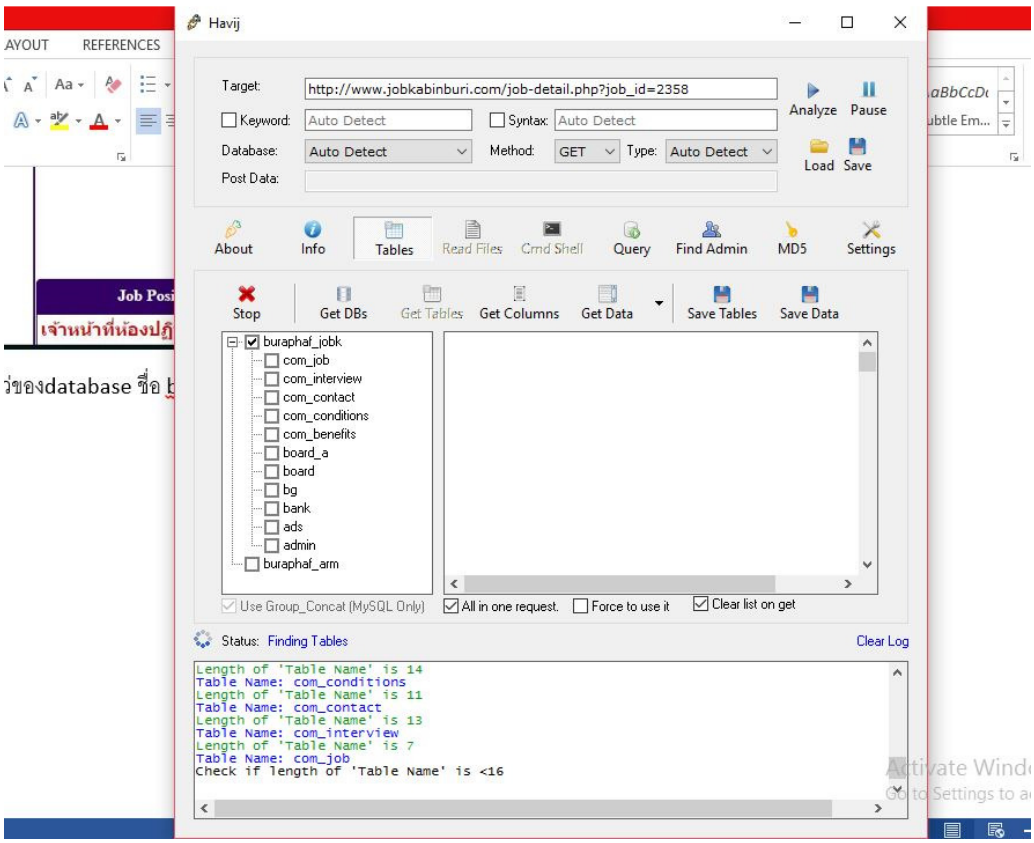
รูปภาพที่ 3 นำ url ของเว็บที่จะทำการแฮ็ก ใส่ช่อง Taget ของโปรแกรม เพื่อ analyze



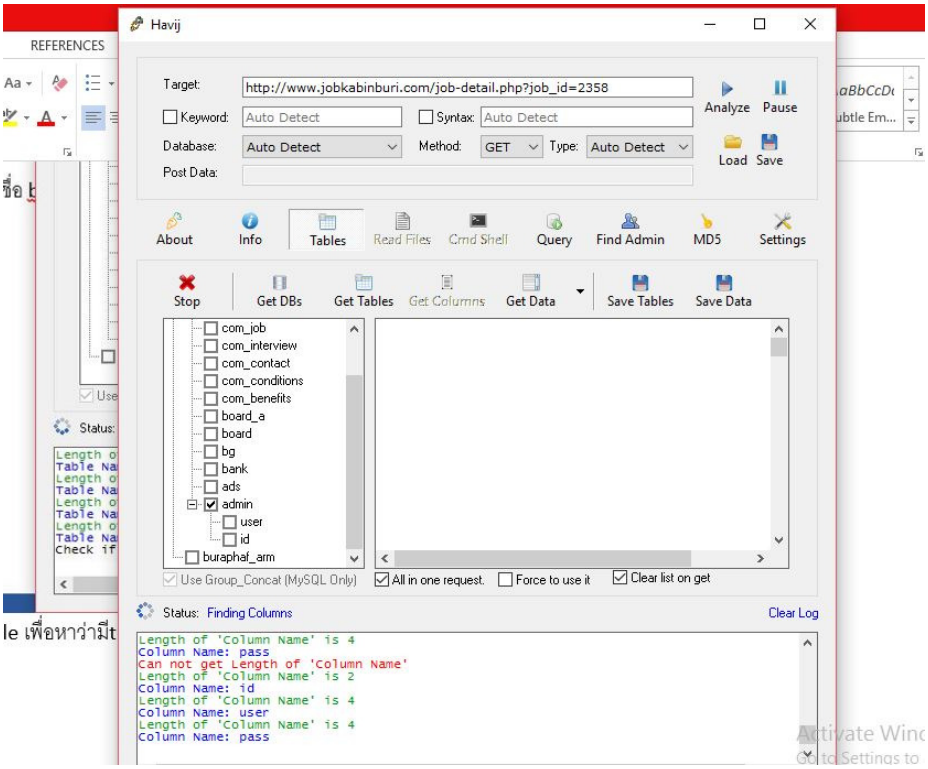
รูปภาพที่ 4 จะพบช่องโหว่ของ database ที่มีชื่อ buraphaf_jobk



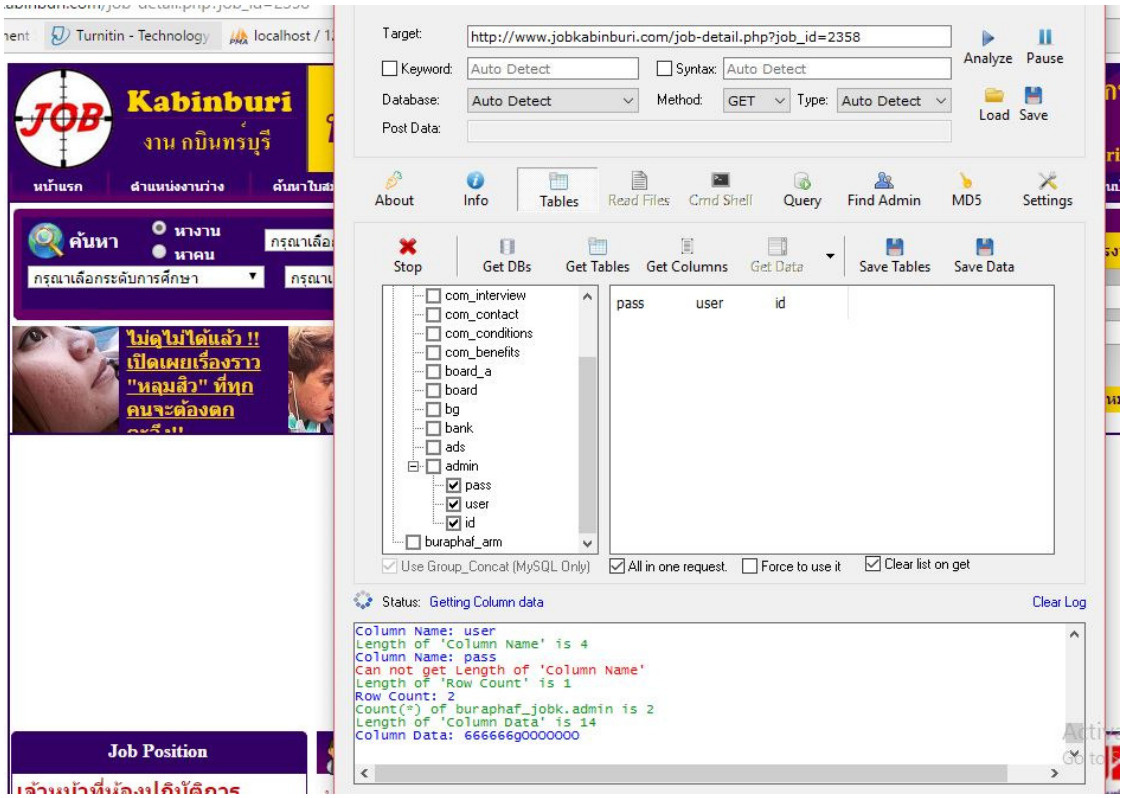
รูปภาพที่ 5 จะมี Info เป็นคำสั่งที่ดูข้อมูลของเว็บ และปรากฏ Target ,Host IP ,Web Server DB Sever Current DB ดังรูป



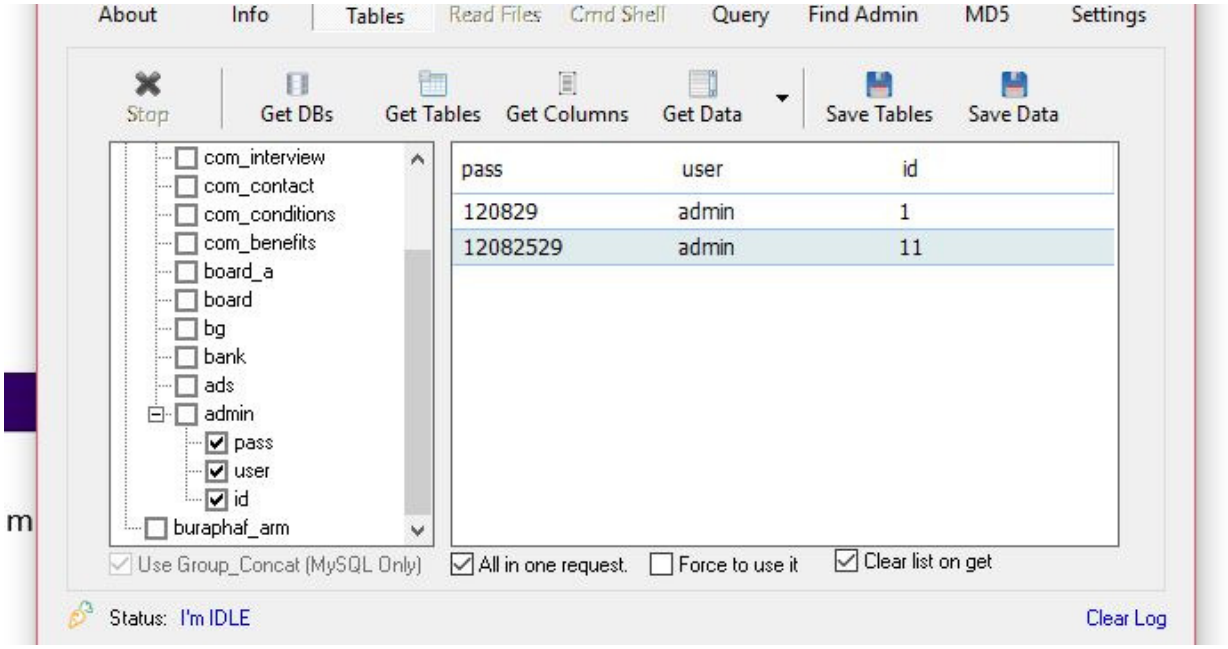
รูปภาพที่ 5 ให้รูปภาพที่ 6 ใช้คำสั่ง get table เพื่อค้นหาว่ามี table อะไรบ้าง



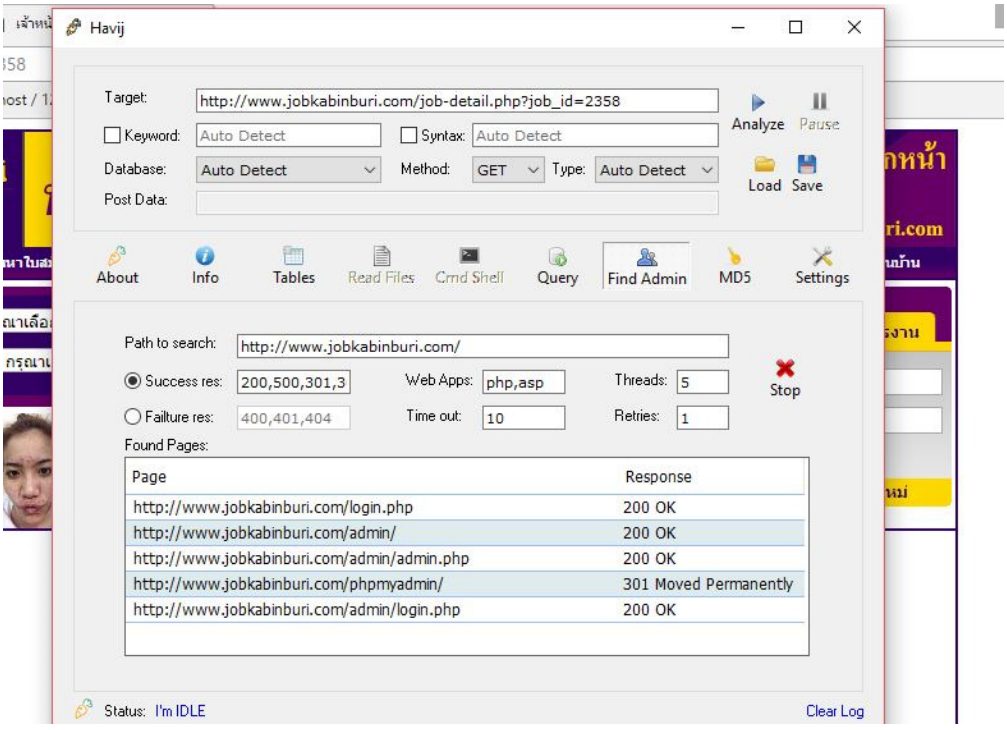
รูปภาพที่ 7 จะปรากฏ ตัวอย่าง เช่น ถ้าเลือก table ที่ชื่อว่า admin แล้วใช้คำสั่ง get columns เพื่อค้นหา columns ใน table admin



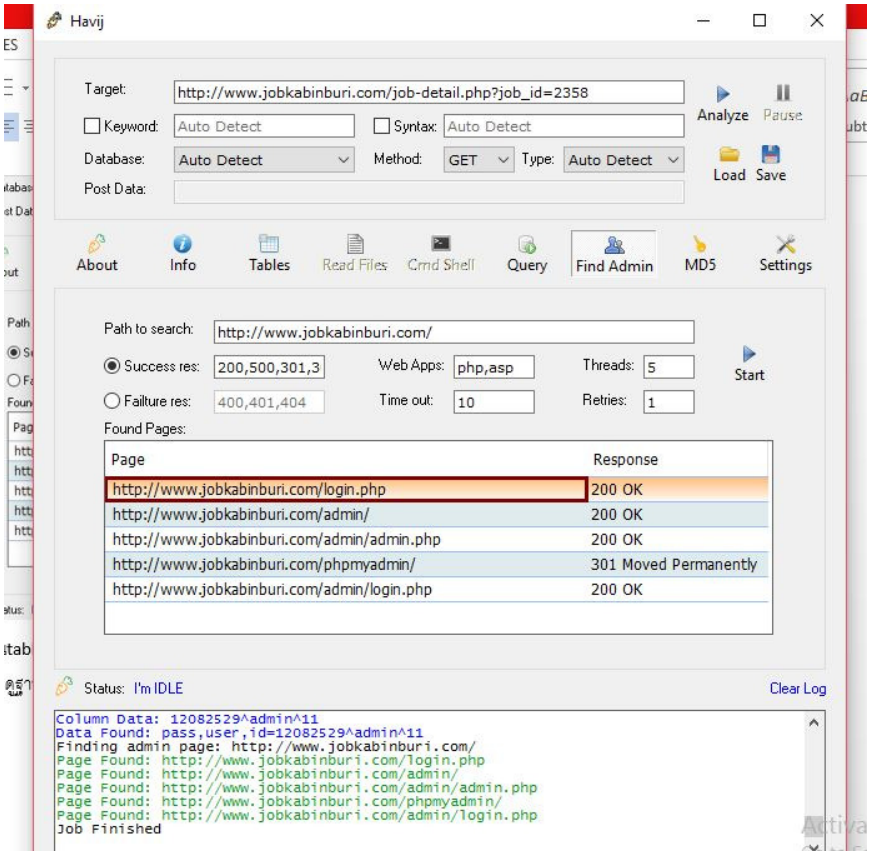
รูปภาพที่ 8 เมื่อค้นหาเจอ columns แล้ว ในที่นี้คือ pass user id



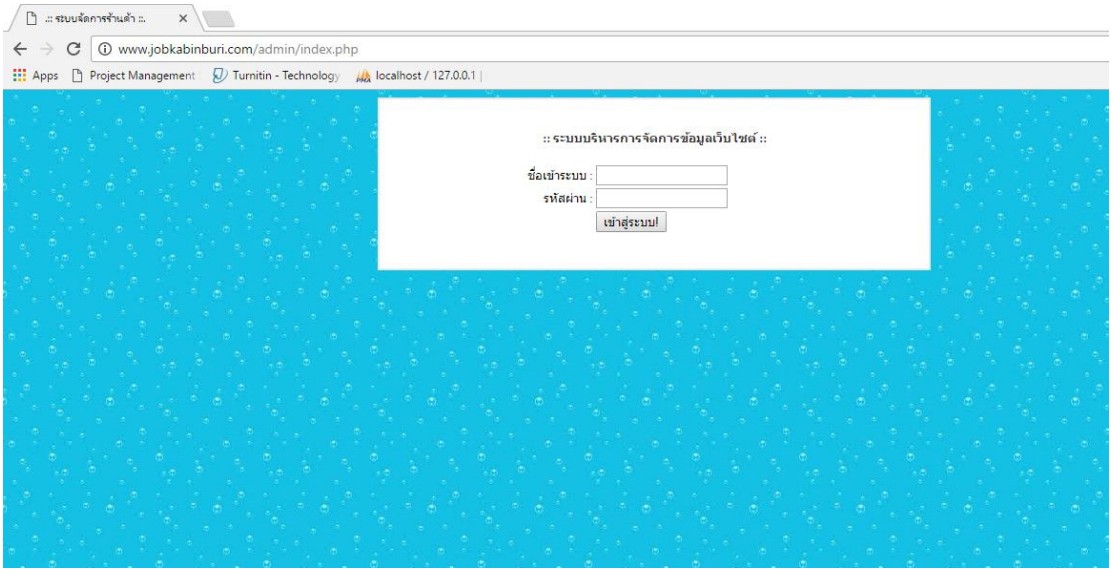
รูปภาพที่ 9 จากนั้นให้ใช้คำสั่ง เลือกทั้ง 3 columns ดังรูป และใช้คำสั่ง Get Data ค้นหาข้อมูลใน columns ของ table admin เพื่อจะเข้าสู่ระบบ



รูปภาพที่ 10 ใช้คำสั่ง Fine Admin ค้นหา URL ของเว็บดังกล่าวที่เป็นหน้ากรอก username , password

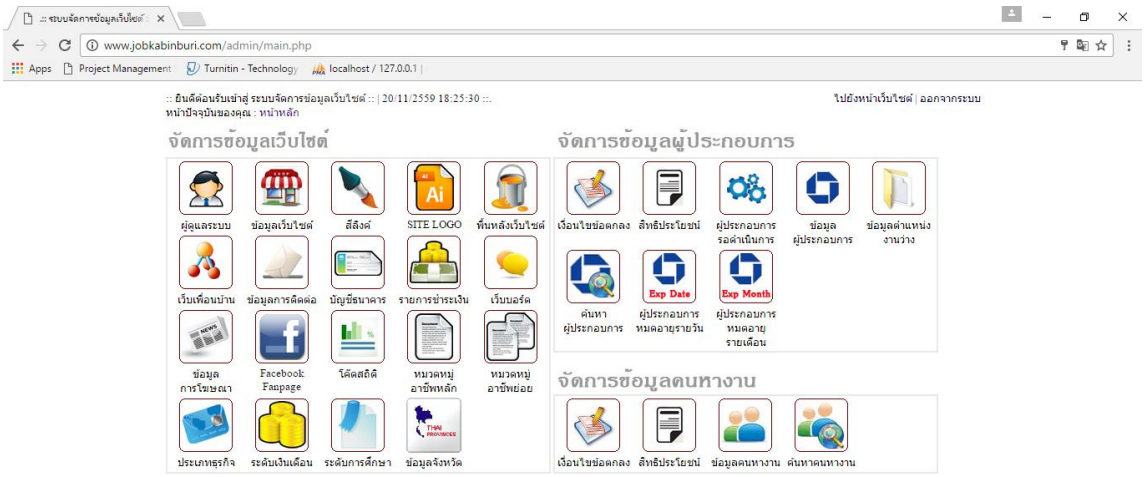


รูปภาพที่ 11 เลือกมาหนึ่ง URL คลิกขวาแล้วกด Open URL

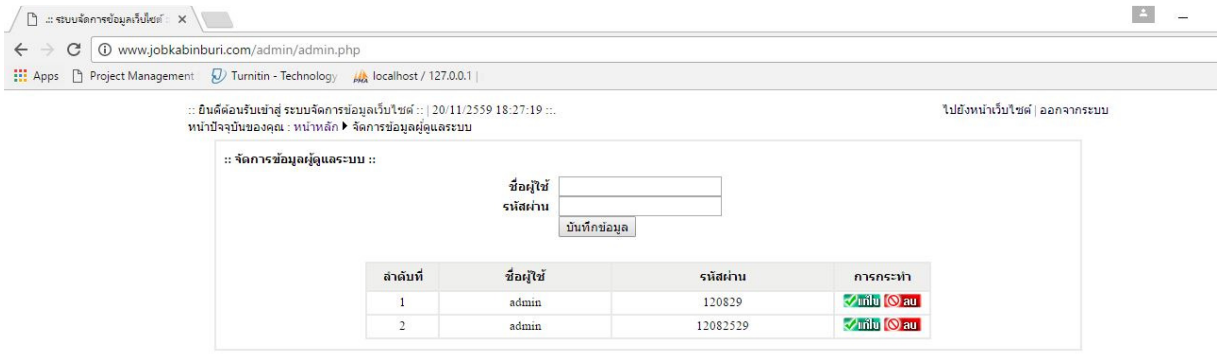


รูปภาพที่ 12 เมื่อนำ URL ที่ได้รันบนBrowser จะปรากฏหน้าต่างเพื่อทำการ login เข้าสู่ระบบให้เรา

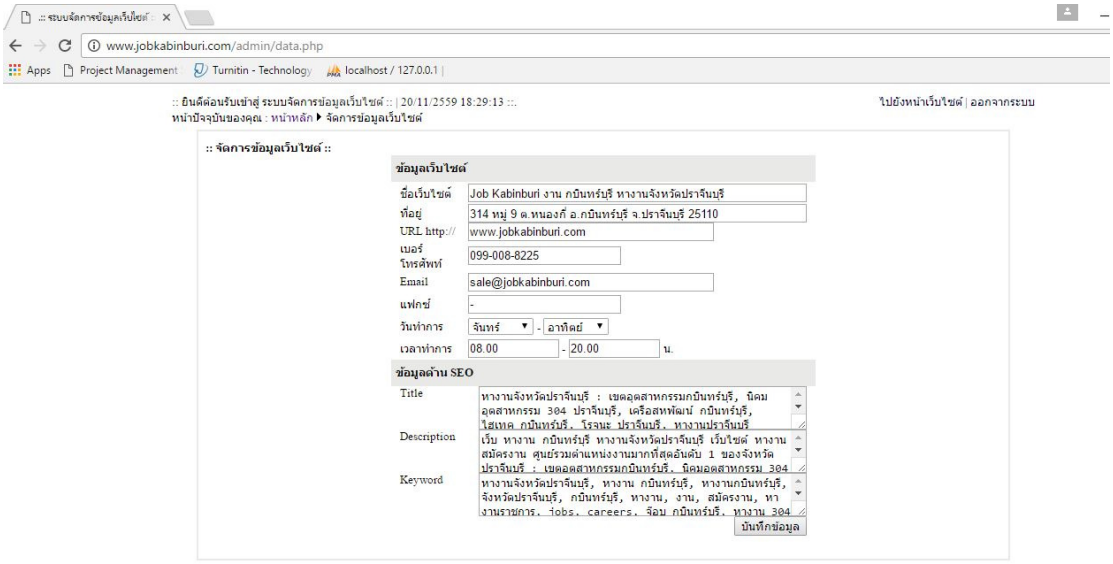
Username : admin Password : 120829



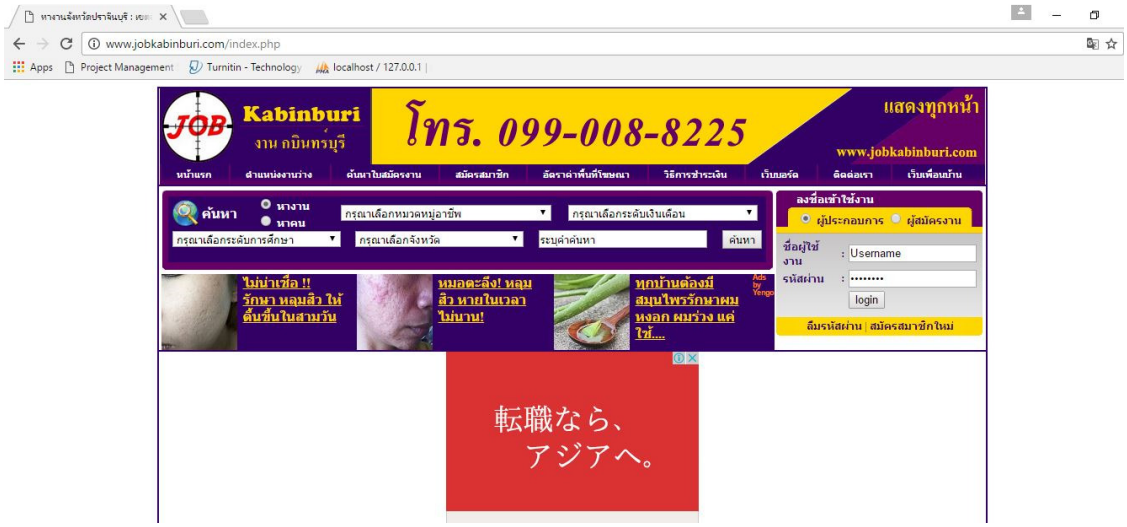
รูปภาพที่ 13 หลังจากกรอก Username และ Password จะปรากฏหน้าต่างข้อมูลภายในเว็บดังภาพ



รูปภาพที่ 14 ตัวอย่าง 1 : การแก้ไขข้อมูลของผู้ดูแลระบบ



รูปภาพที่ 15 ตัวอย่าง 2 : การจัดการข้อมูลเว็บไซต์



รูปภาพที่ 16 หลังจากนั้นออกจากระบบให้เรียบร้อย จะกลับมาหน้าเว็บไซต์ สำหรับผู้เข้าชมทั่วไป

เอกสารอ้างอิง

https://www.techtalkthai.com/fallacy-of-sql-injection/

http://newscentral.exsees.com/item/a7c466a18607b6348e7e7c21ee43deff-5d2fb6d42c432c12bd7a63ae5eb72197

Damn. (2016). Havij. ค้นเมื่อ 20 พฤศจิกายน 2559, จาก http://newscentral.exsees.com/item/a7c466a18607b6348e7e7c21ee43deff-5d2fb6d42c432c12bd7a63ae5eb72197

ทวีชัย เสริมสร้าง. "การพัฒนาเทคนิคการป้องกันและตรวจจับ SQLIA". วิทยาศาสตรมหาบัณฑิต สาขาความมั่นคงทางระบบสารสนเทศ คณะวิทยาการและเทคโนโลยีสารสนเทศ มหาวิทยาลัยเทคโนโลยีมหานคร, 2557.