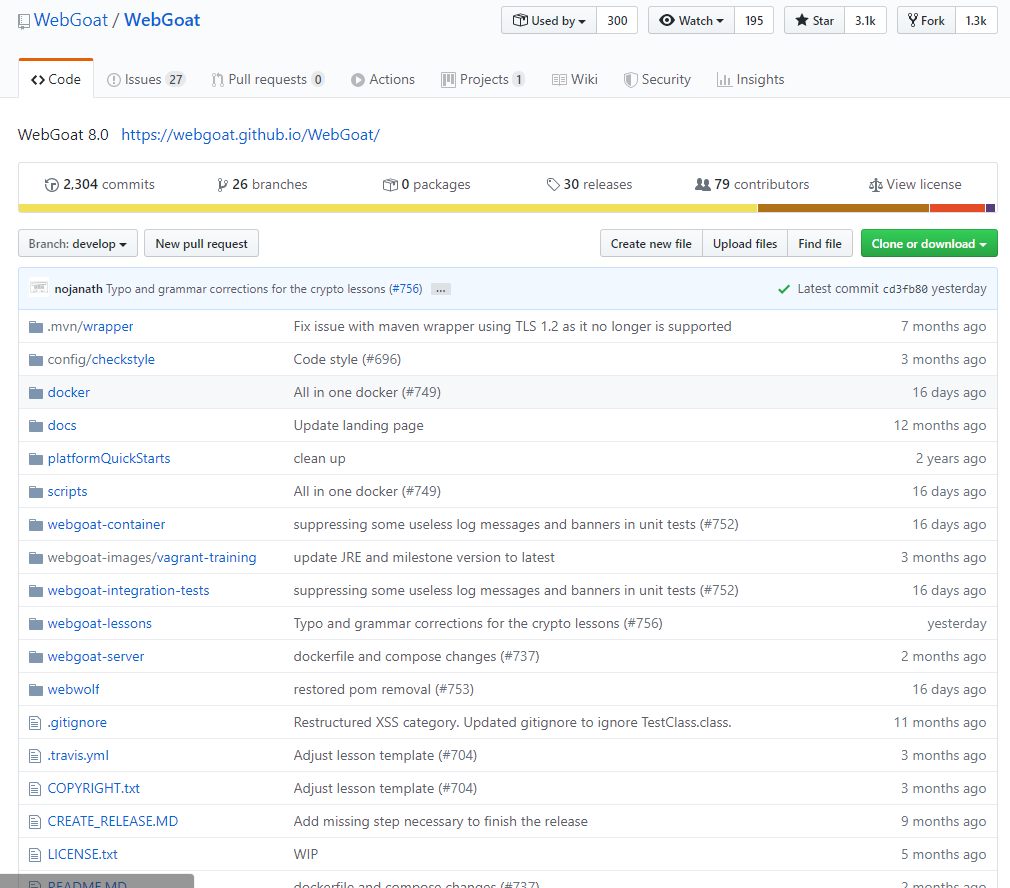
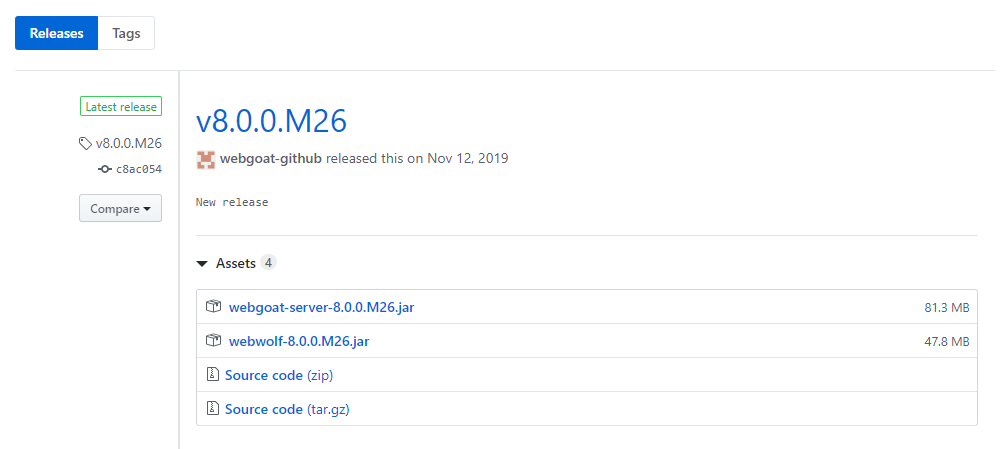
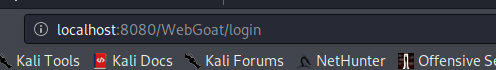
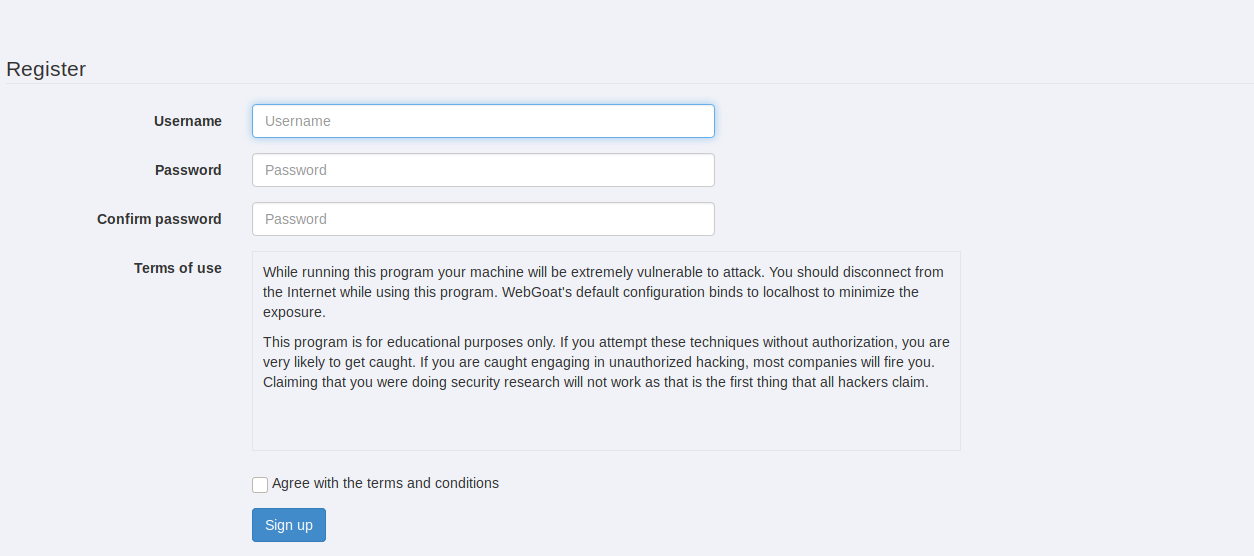
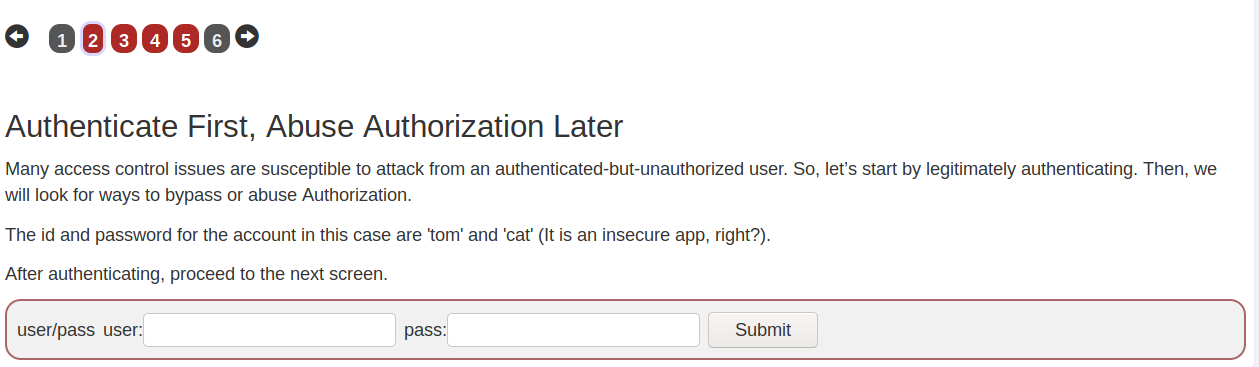
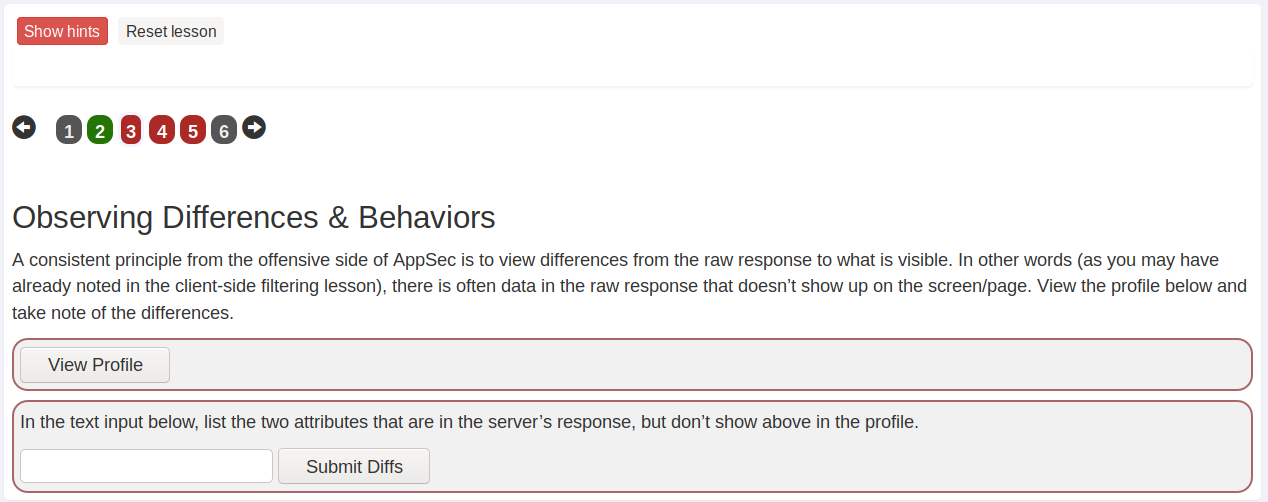
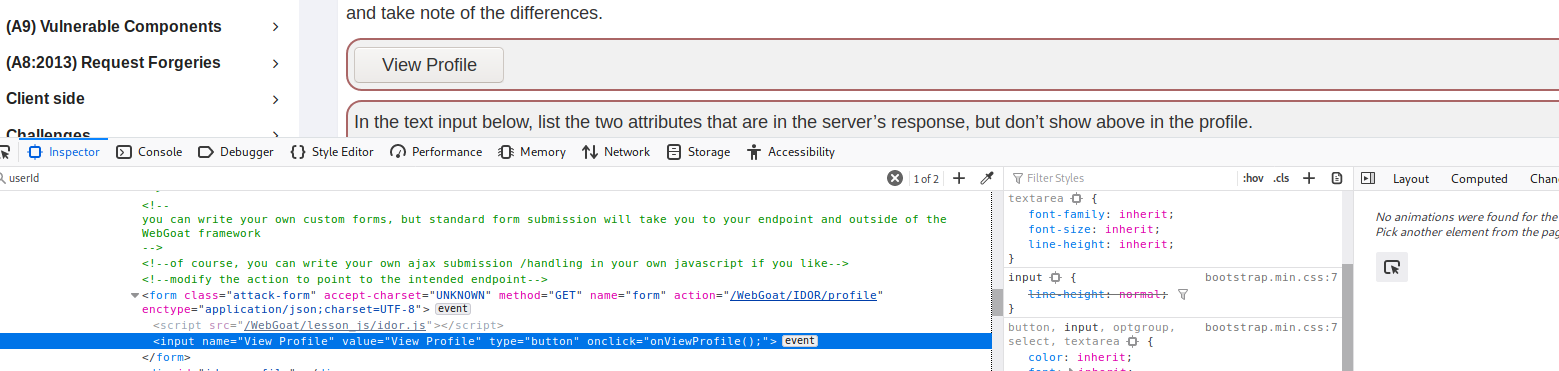
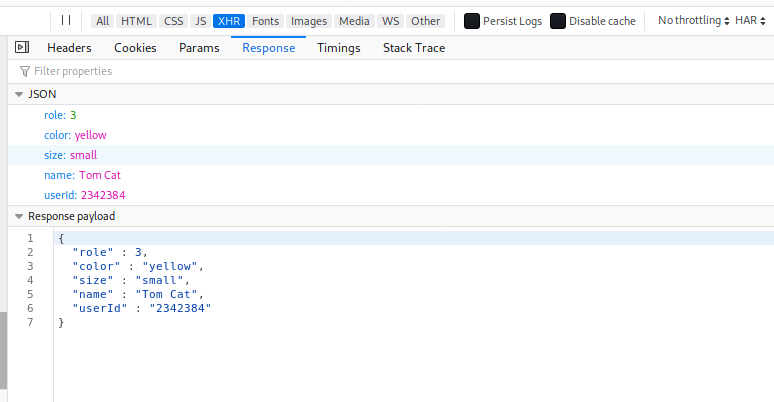
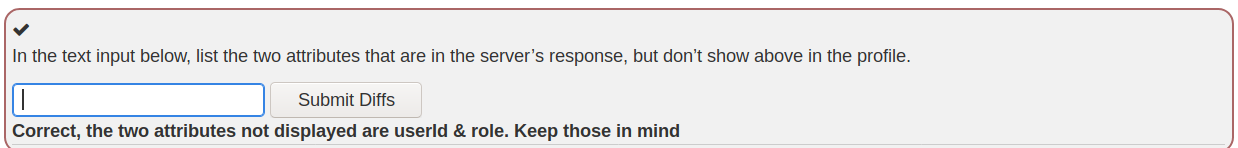
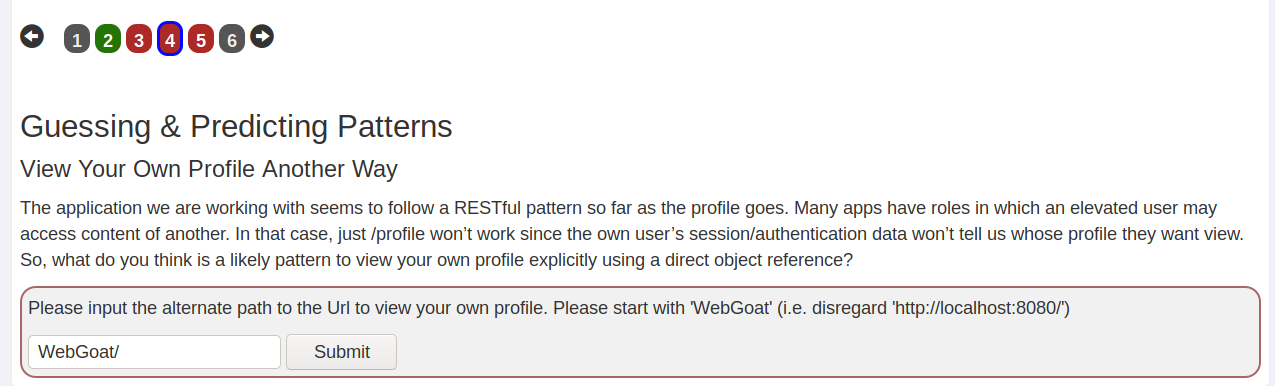
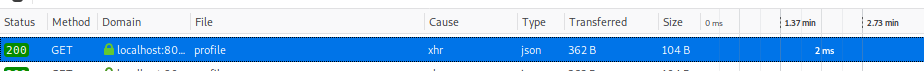
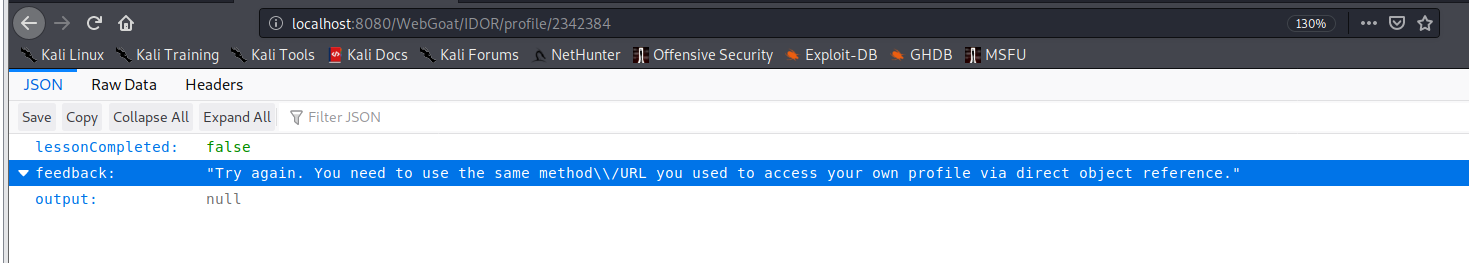
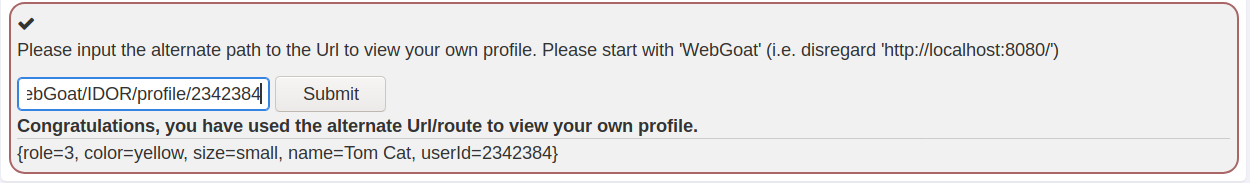
Webgoat  
สอนการติดตั้ง  
 เนื่องจาก webgoat นั้นถูกออกแบบมาให้มีช่องโหว่ เวลาใช้งานควรปิดอินเตอร์เน็ต แต่โดยปกติ webgoat จะรันบน localhost ที่ port 8080 ให้เราทำการติดตั้ง webgoat บน virtual machine แทนตามขั้นตอนต่อไปนี้  
1. เข้าไปที่ github ของ webgoat   
  
2. ไปที่ releases แล้วเลือกดาวโหลด webgoat-server-8.0.0.VERSION.jar   
3. เข้าไปยัง directory ที่ได้ทำการดาวโหลดไว้ รันคอมมานต่อไปนี้  
  
4. รอจน webgoat เตรียมพร้อม env ให้เราเสร็จแล้วเข้าไปที่ url ใน web brower ของเรา   
  
5. ทำการสมัครเพื่อใช้งานตามปกติ  
  
6. ทำการลอคอินเข้าใช้งาน

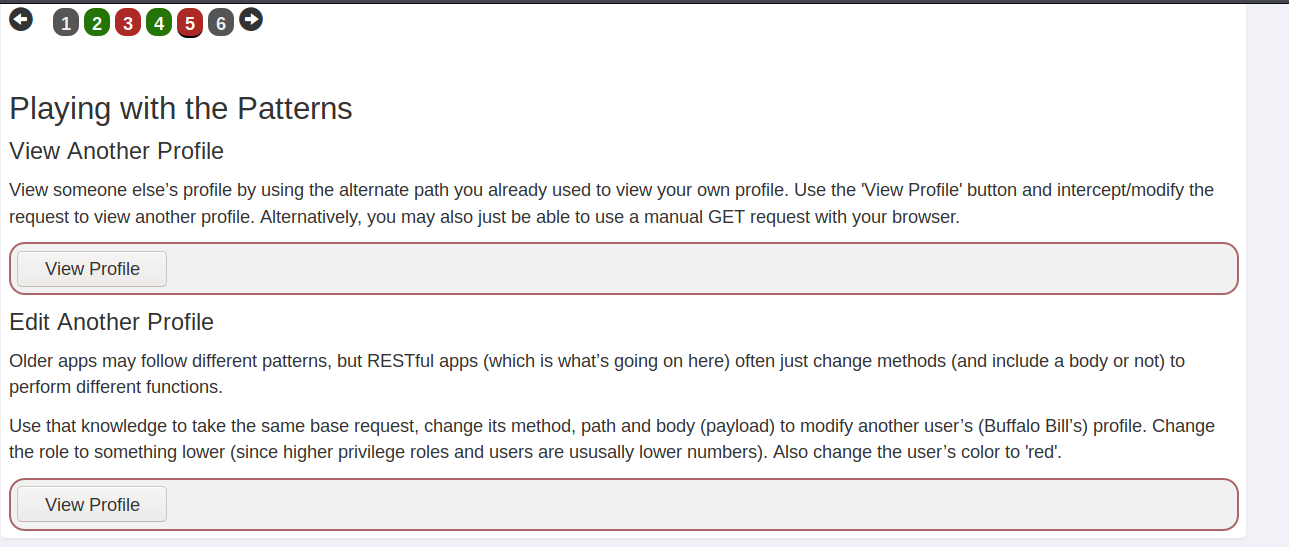
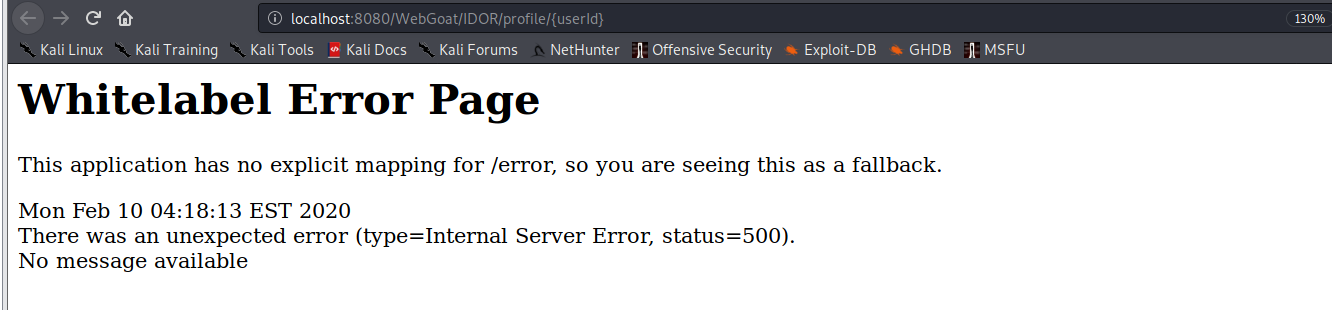
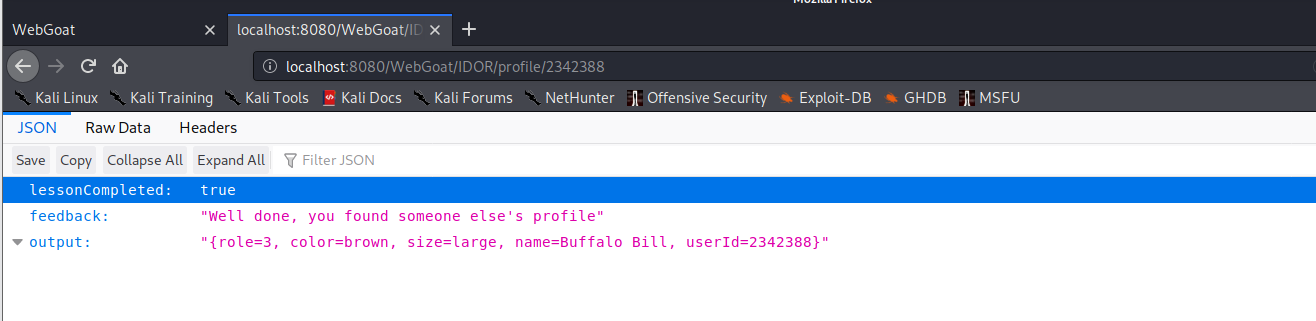
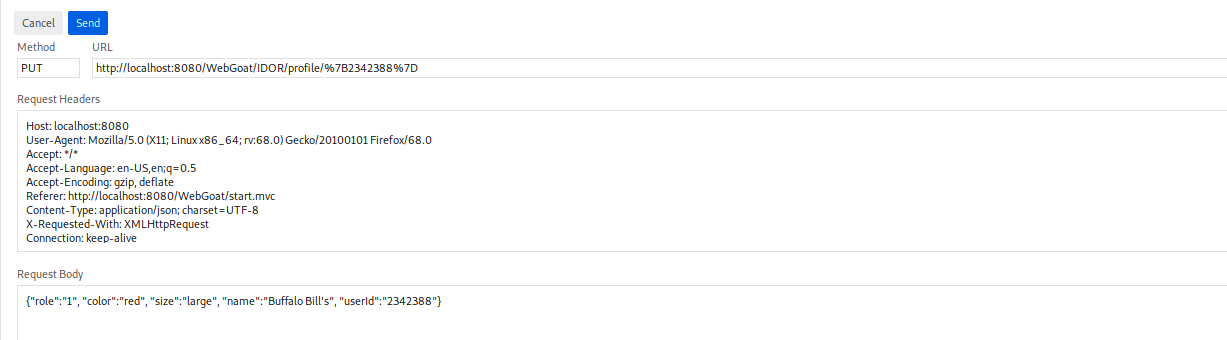
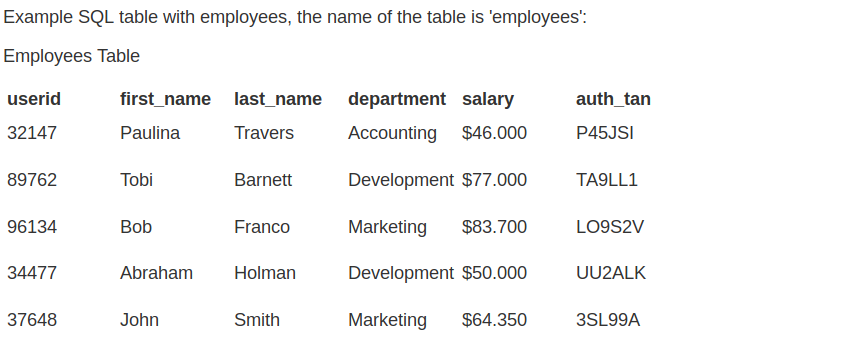
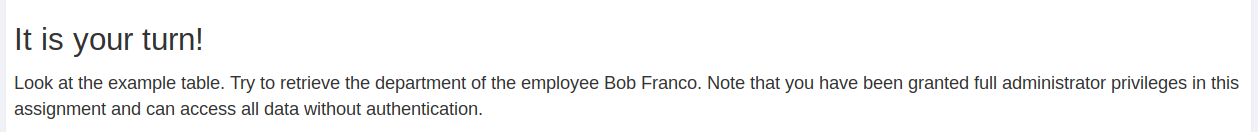
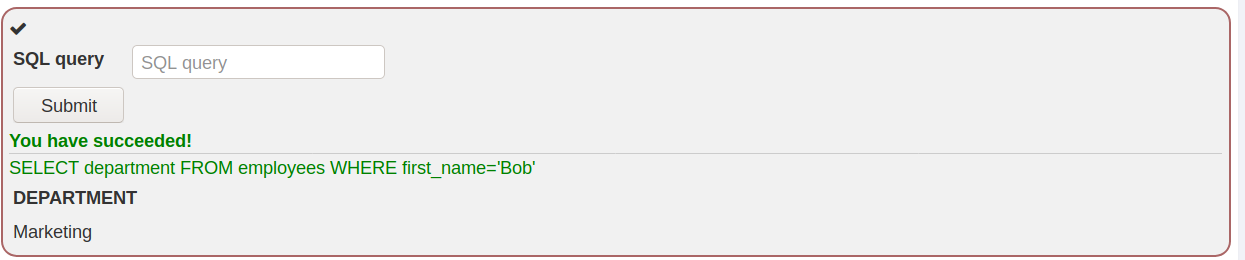
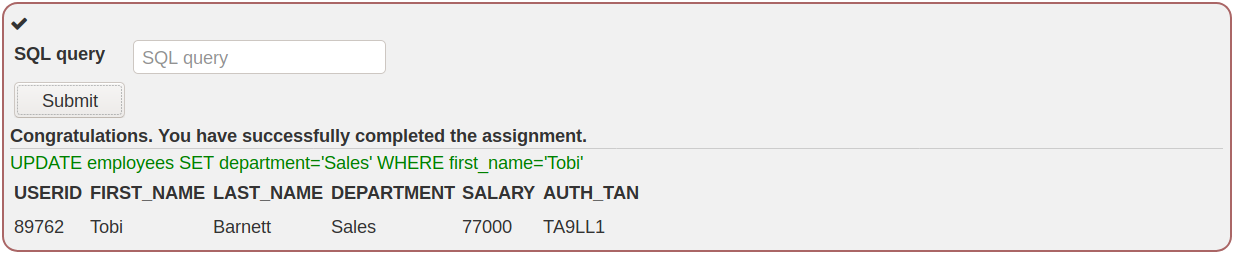
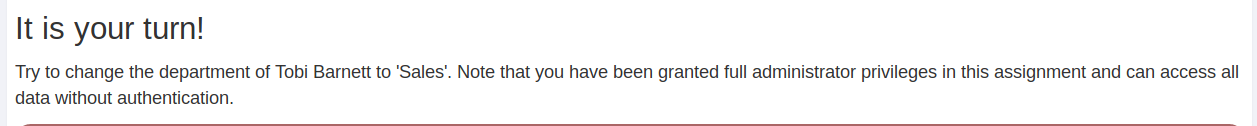
หมวก A5 Broken Access Control  
 คือ ช่องโหว่ที่เกิดจากการควบคุม user ว่าทำอะไรได้และไม่ได้อย่างไรได้ไม่ดีพอ ทำให้ Hacker สามารถเข้าถึงการทำงานและข้อมูลต่างๆที่ไม่ได้รับอนุญาต เช่น การเข้าถึงข้อมูลของ user ผู้อื่น, การเข้าถึงไฟล์สำคัญของระบบ, การแก้ไขข้อมูลของ user อื่นๆ, การกระทำคำสั่งของผู้ดูแลระบบโดยใช้สิทธิ์ของ user ธรรมดา เป็นต้น  
  
เริ่มกันที่หมวด IDOR  
 Insecure Direct Object Reference ถ้าตีความจากชื่อของช่องโหว่แล้วหละก็คงจะหมายถึง การอ้างถึงวัตถุทางตรงแบบไม่ปลอดภัย? ในความเป็นจริงแล้ว ช่องโหว่ประเภทนี้เป็นช่องโหว่ที่เกิดจากความผิดพลาดในการทำ authorization (การตรวจสอบสิทธิ์ในการเข้าถึง) นั่นก็คือ หากเว็บแอปพลิเคชันไหนปล่อยให้ผู้ใช้งานสามารถเข้าถึงข้อมูล หรือ ไฟล์ต่าง ๆ ที่ไม่มีสิทธิ์ในการเข้าถึงได้โดยตรง ก็เรียกได้ว่ามีช่องโหว่ประเภท IDOR อยู่ ทำได้โดยการเปลี่ยนค่า ที่อยู่ใน url โดยอาจจะเป็น id หรือ ชื่อ file ก็ได้ หากไม่ได้ทำการตรวจสอบสิทธิ์ในการเข้าถึงแล้วทำให้ hacker สามารถเข้าถึงข้อมูลที่ไม่ควรได้  
  
ข้อที่ 1  
  
 ให้เราทำการกรอก user = tom, pass = cat แล้วทำการ submit จะเห็นได้ ลอคอินสำเร็จ

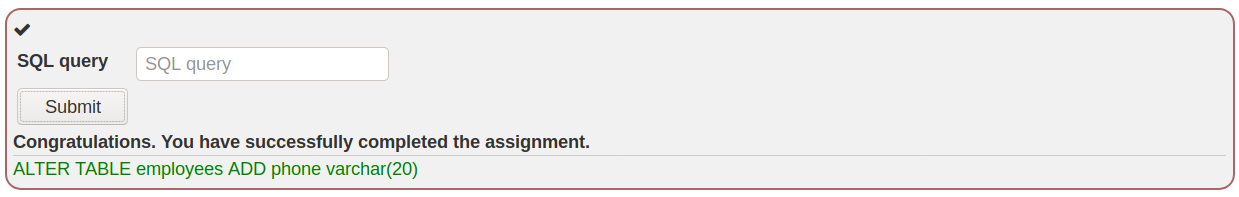
ข้อที่ 2  
ลองกด F12 เพื่อดูการทำงานคร่าวๆ  
  
จะเห็นได้ว่าใน tag action มีการส่งข้อมูลไปที่ path /WebGoat/IDOR/profile พอเราลองกดดูปุ้ปปป  
  
 เห็นได้ว่ามีการส่งข้อมูลกลับมา โดยบางครั้งหากคนออกแบบ api ไม่ดีจะมีการ return ค่าที่ไม่ต้องการกลับมาด้วยเช่น id หรืออื่นๆ ในรูปแบบของ json ก็ให้เราเข้าไปดูที่ network ด้วยการกด F12 ไปที่ XHR แล้วเลือก packet ที่เราต้องการดู response แล้วกด response เพื่อดูว่าใน json มีอะไรบ้าง

  
 จะเห็นได้ว่ามี 2 fieldที่มันไม่ได้ถูกนำมาแสดงโดยฝั่ง front end นั้นก็คือออ role, userId ก็ทำการใส่คำตอบนี้เข้าไป  
  
 เขาบอกว่า keep those in mind แปลว่า เก็บไว้ในใจนะจ้ะ

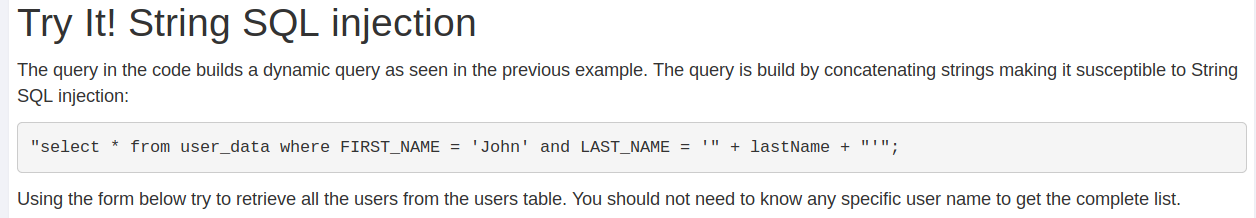
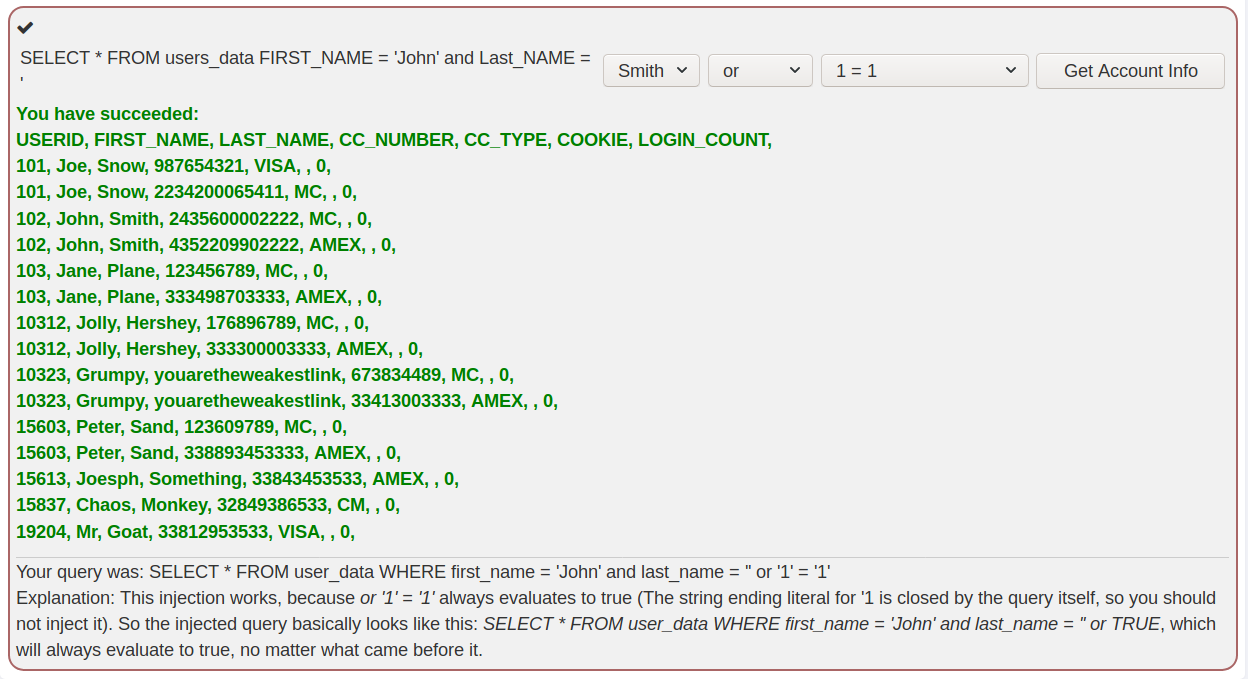
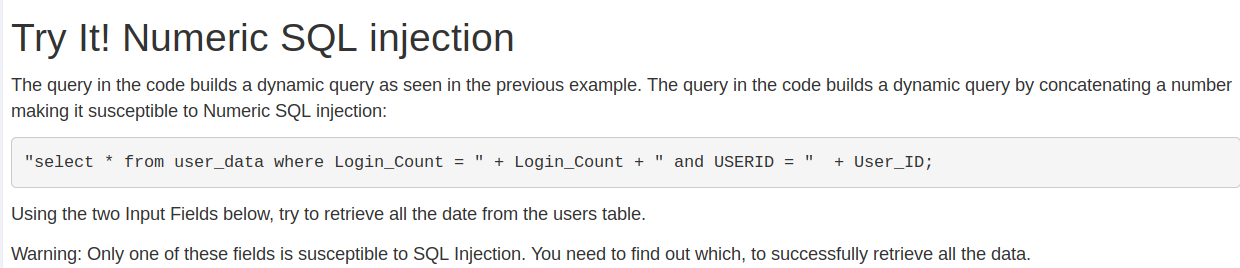
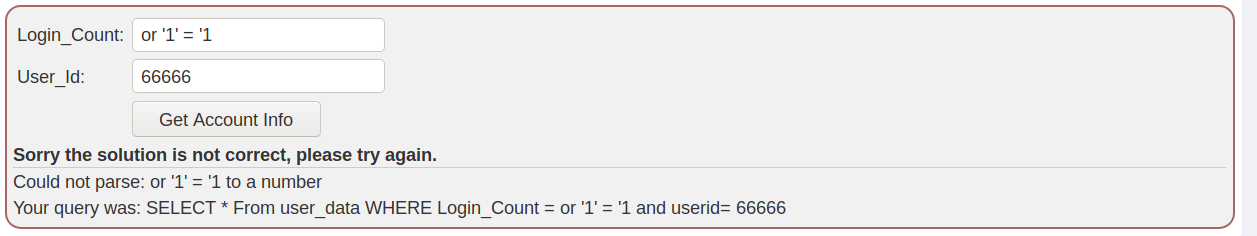
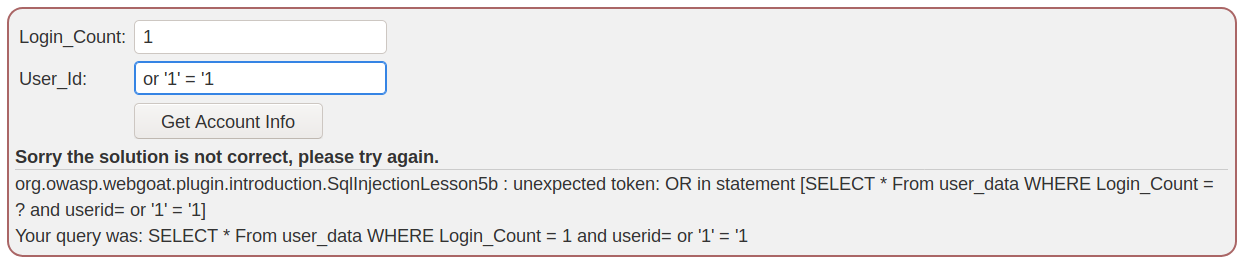
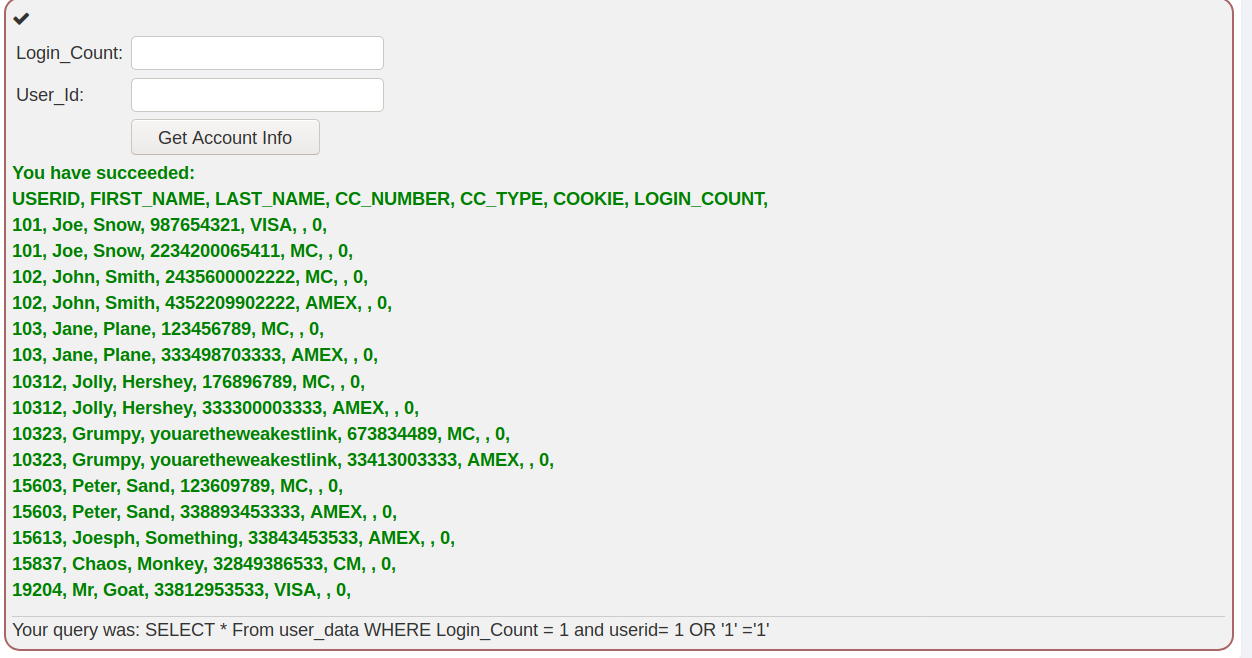
ข้อที่ 3

  
 ในรูปแบบ framework ของ restful api โดยการเข้าถึง record ต่างๆใน database โดยพื้นฐานจะถูกตั้งค่าเป็น path/xxxx/[user id]/[function name] โดย user id คือชื่อของ field ใน database ที่เป็น primary key โดยหากออกแบบรูปแบบ id หรือ permissionได้ไม่ดีจะทำให้เกิดช่องโหว่นี้ได้ง่าย ส่วน function name นั้นเราต้องรู้ชื่อ function ก่อนเพื่อให้ record ในนั้นสามารถรันฟังก์ชั่นที่เราต้องการได้  
 ย้อนกลับไปดูการส่ง packet อีกรอบจะเห็นได้ว่า column File มีชื่อว่า profile ดังนั้นแสดงว่า request เมื่อกี้เรียกข้อมูลจากไฟล์นี้  
  
 เราลองเรียกข้อมูล profile ของเราผ่าน path ที่ packet นี้ส่งไปดูโดยการ double click ที่ packet ข้างต้นต่อให้เราใส่ user id ของเราเข้าไปจากข้อ 2 ที่เราได้มา ตามหลัง profile/  
  
 แป้ววว ผิด method นะครับ แสดงว่า urlpattern แบบนี้แหละที่เป็นคำตอบ ขกไปนั่งเปลี่ยน method แล้วเนื่องจาก burpsuite ผมมีปัญหา ก็ทำการใส่ตามฟอร์มที่เขากำหนด  
  
กด submit

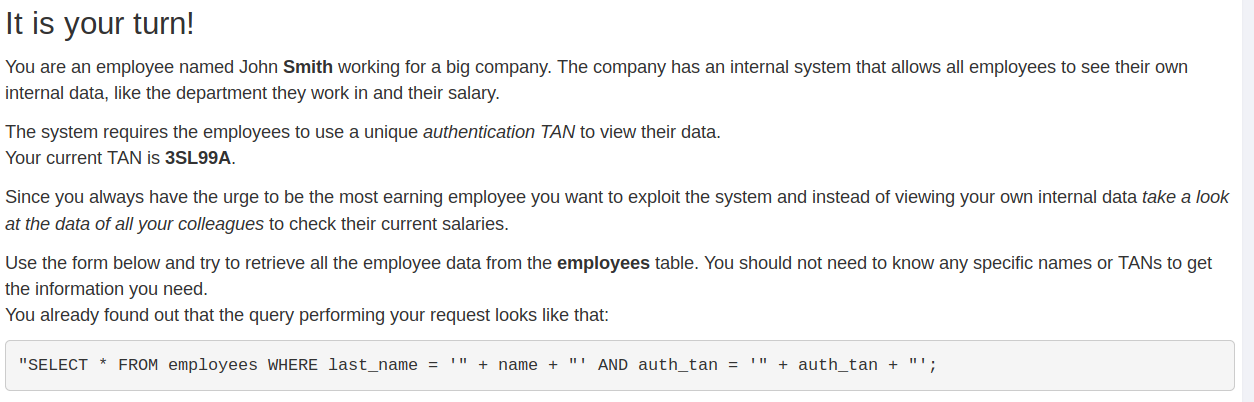
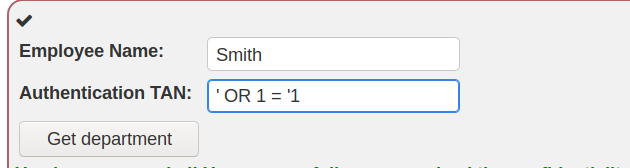
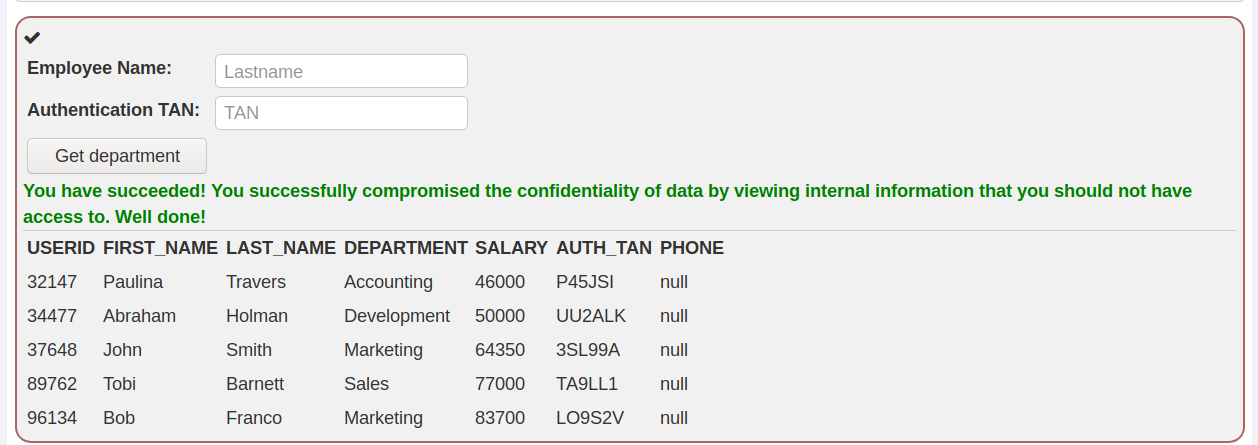
ข้อ 4  
  
 จากที่ที่เคยนกล่าวมาแล้วการออกแบบการตรวจสอบสิทธิ์ไม่ดีทำให้เราสามารถเข้าถึง profile คนอื่นได้ผ่าน path ที่คล้ายๆกัน  
 ลองกด view profile แล้วส่ง packet เล่นกันน  
  
 ลองเปลี่ยน {userId} เล่นเผื่อมีโอกาศตรงกับ userId ของคนอื่นทำให้เราสามารถเข้าถึงข้อมูลของคนอื่นได้  
  
 เจอ 2342388 เป็น path ที่เราเข้าถึงได้  
  
ทำการแก้ไขข้อมูลโดยใช้ method PUT กับค่าที่ส่งเช้าไป  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
SQL injection  
ข้อ 2  
  
 มันบอกว่าเรามี full admin แสดงว่าเรา query อะไรเข้าไปก็ได้  
  
ข้อ 3  
  
ข้อ 4

  
ใช้คำสั่ง DDL ประเภท alter แน่นอนเพราะมีการเปลี่ยนแปลง structure ของ table  


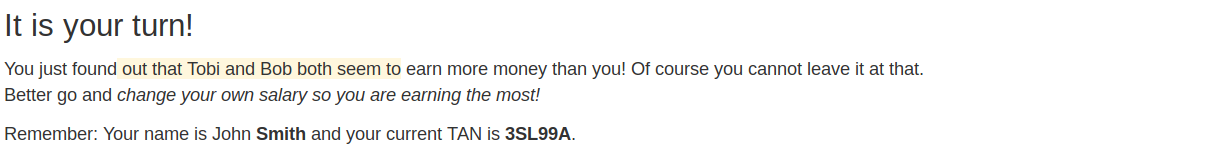
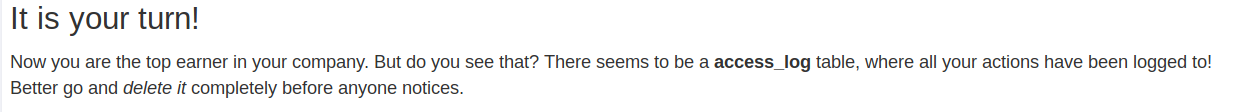
ข้อ 5  
  
เปรียบเสมือนการให้สิทฑิการเปลีย่แปลงกัย Unauth..  


ข้อ 6  
  
จากการวิเคราะห์ เราสารถทำให้เงื่อนไข where เป็นจริงได้โดยการที่ให lastname เป็นช่องว่างเปล่าๆ จบในตัวแล้วทำการ inject เงื่อนไขอื่นเพิ่มเข้าไป  
  
  
ข้อ 7  
  
คนเราต้องมั่ว  
  
จาก error จะเห็นได้ว่า Login\_count มันต้องทำการ cast t o int ก่อน string ไม่สามารถทำเป็น int ได้ทำให้เกิด exception ลองย้ายไปข้างล่างแทน  
  
มี error กลับมาแล้ววว ว้าวๆ  
 ทำให้เรารู้ค่าที่เราทำหายไป userid = Nani or ที่เรา inject เข้าไปก็ยังไม่ถูก syntax ก็ทำการแก้  


ข้อ 7

  
เริ่มแรกมาก็ลองเอาข้อมูลของตัวเองก่อนแม่งเลย เท่ดี  
  
ลอง inject เข้าไปในส่วนของ TAN   
ถ้าดูจาก syntax and auth\_tan = ‘เรากทำการปิดสตริงให้มัน ‘ แล้วก็ inject command เข้าไป  
  
ก็จะได้ output  


ข้อ8

  
ข้อนี้ใช้ chain query   
'; UPDATE employees SET salary=99999 WHERE last\_name ='Smith;--  
‘ แรกปิดตรง query ของ authen แล้วทำการใส่ ; เพื่อบอกว่ามันจบ 1 sql command แล้วก็ยัดเข้าไปอีกอันด้วยคำสั่ง UPDATE เสร็จแล้ว ก็ทำการปิด ;-- เผื่อเขามีด้านหลังต่อ  
ข้อ 9  
  
ข้อนี้ต้องทำลาย log  
จากการเดา “อะไรสักอย่าง = ‘ค่าที่ป้อนเข้ามา’” อาจจะต่อด้วยอะไรสักอย่าง”  
เราก็ทำการเปลี่ยนเป็นค่าที่ป้อนเข้ามาเป็น  
A'; DROP TABLE access\_log;--  
A’ เพื่อปิด การ query ครั้งแรก เสร็จแล้วก็ทำการ drop table นั้นทิ้ง ตามด้วย ;-- เพื่อทำการป้องกันเผื่อมีอะไรด้านหลังอยู่  
