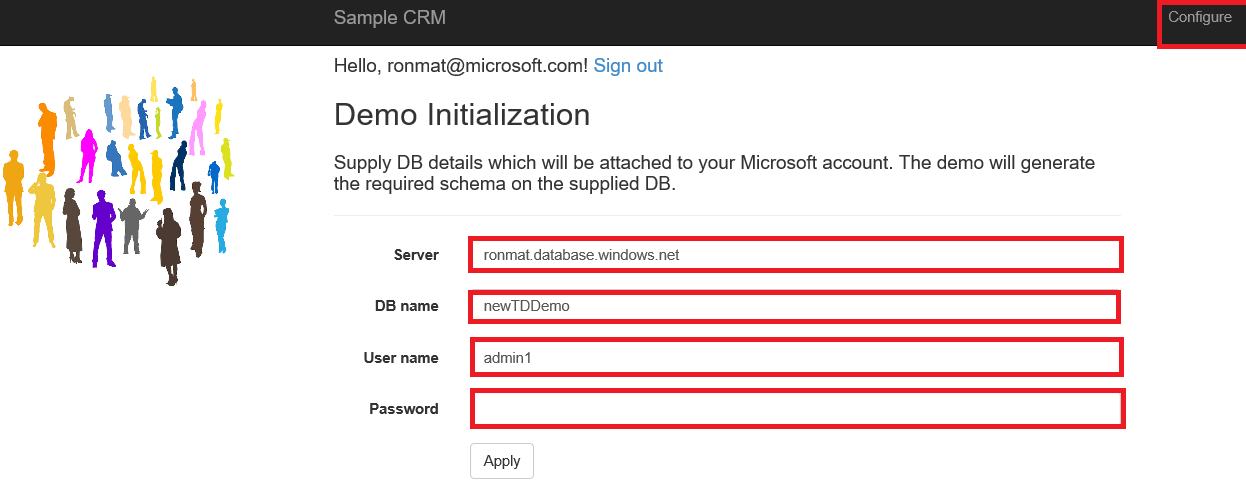
New SQL Threat Detection Demo

Demo Environment

1. **Create** SQL Database & [Configure](https://mybadwebsite.azurewebsites.net/Home/Init) the shared demo app to use the database as its backend
2. **Enable** Auditing and Threat Detection o from [Azure Portal](https://ms.portal.azure.com/)
3. **Run** below SQL Threat Detection Demo Script



Demo special notes:

* You should explain the audience that the way we detect SQLi is actually super complex based on database behavioral and not specifically detects the simple demo patterns.
* You cannot demo it again for 30 minutes as we do throttling for email alerts

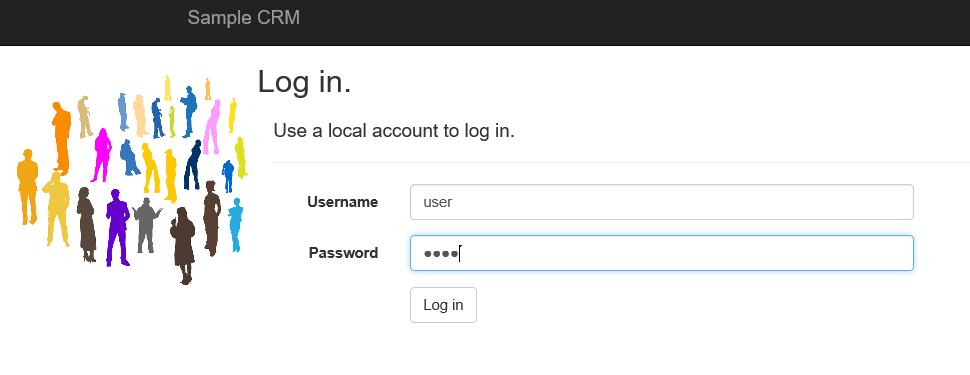
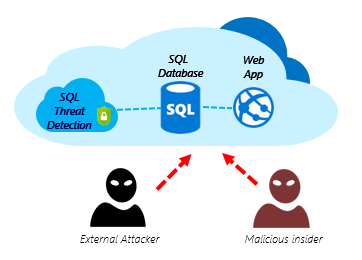
Demo Script

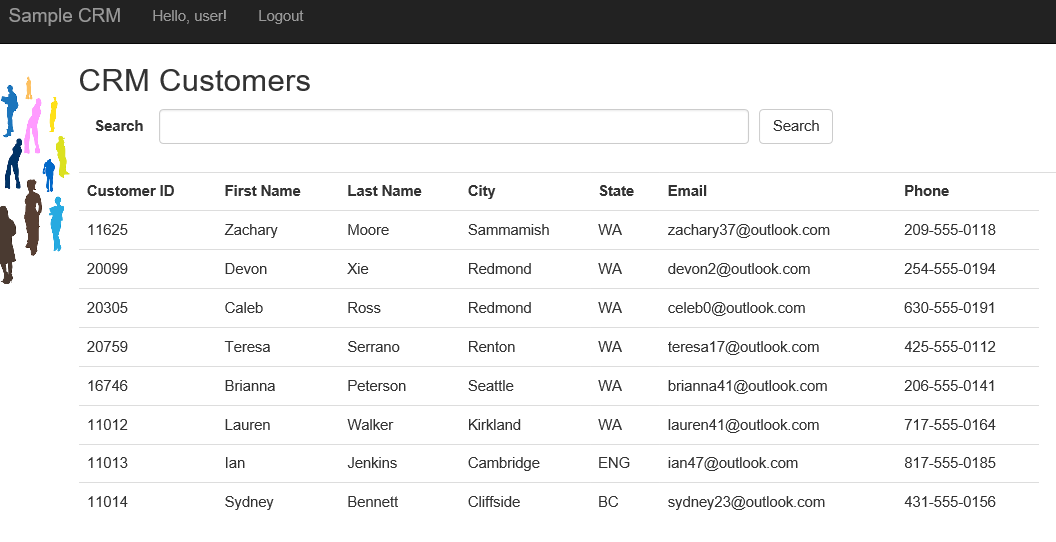
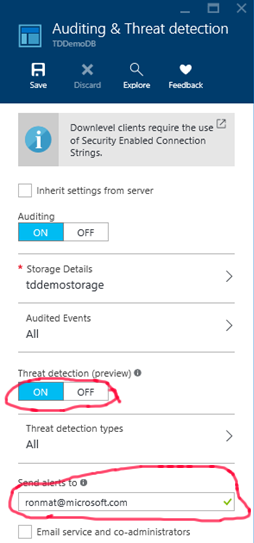
Hi, my name is **\_\_\_\_\_\_**, and I’d like to show you our new ‘Threat Detection” feature.

Threat Detection is a new security feature for Azure SQL database that detects suspicious database activities indicating possible malicious intent to access, breach or exploit data in the database

First I’m going to simulate a particular type of threat – a SQL injection attack – on a demo application. Then I’ll show how the new Threat Detection feature detects this suspicious event as it occurs and helps me explore the relevant events using the SQL audit log. SQL injection is one of the most common threats to data-driven applications on the Internet. It works by injecting malicious SQL statements into input fields of vulnerable web applications.

For this demo, I will be using a [**simple CRM application**](https://mybadwebsite.azurewebsites.net/Home/Login) **that uses Azure SQL Database as its backend** database. This web application accepts user input and uses it to construct SQL statements without properly validating that input, which unfortunately means it’s vulnerable to SQL injection attacks



**STEP1: Setup**

I’ll start by setting up Threat Detection for my database using the Azure Portal.

Setting up Threat Detection is really easy. I’ll navigate to my database in the portal, and then select the Auditing & Threat detection option in the settings blade. You’ll notice that I already have Auditing turned on for this database, which is a prerequisite for Threat Detection. Then simply click a button to turn on Threat Detection, and specify which email accounts should receive the security notifications.

**Now I have enabled Threat Detection for my database.**

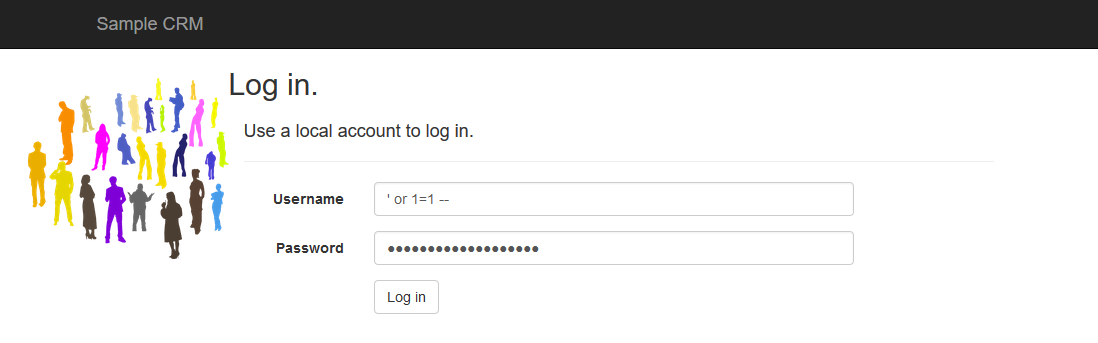
**STEP2: SQL Injections**

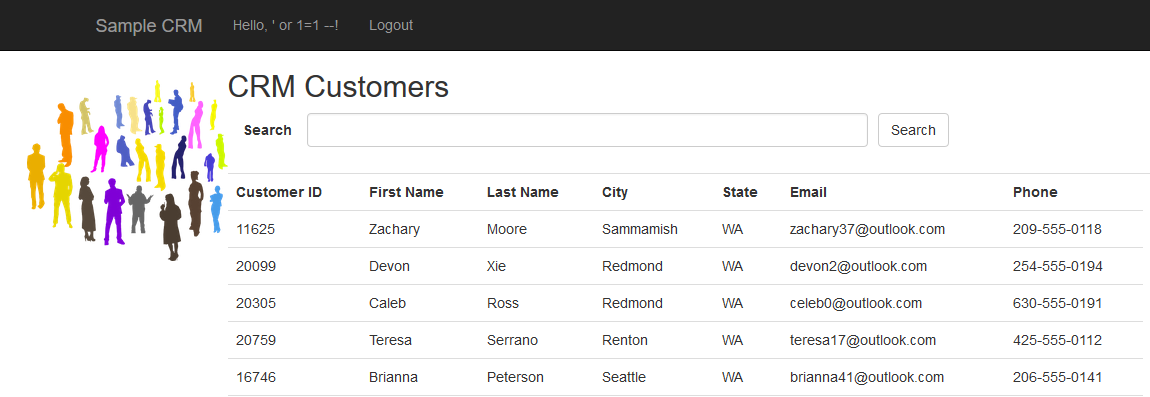
**Now for the fun part. I’m going to simulate the SQL Injection attack manually.**

In particular, I will be using a common SQL injection pattern in the user name field, which exploits application vulnerabilities that don’t properly validate the user’s input.

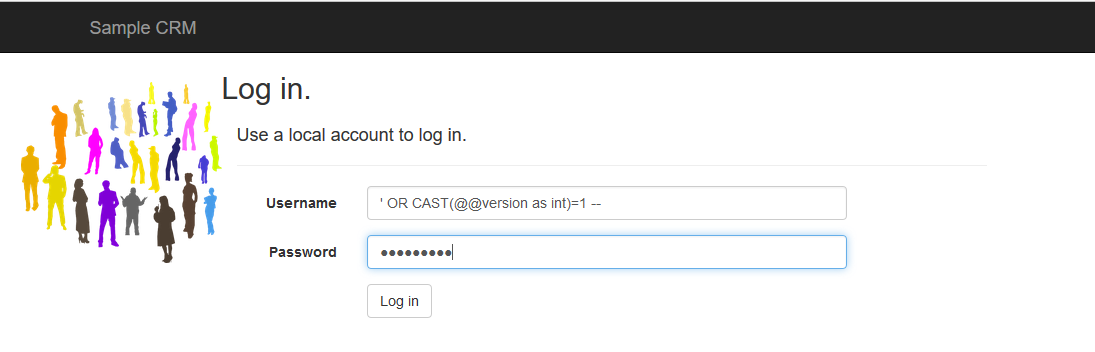
As you can see, these attempts bypass application login checks and allow me to *execute a variety of SQL statements that can retrieve and damage the database.*

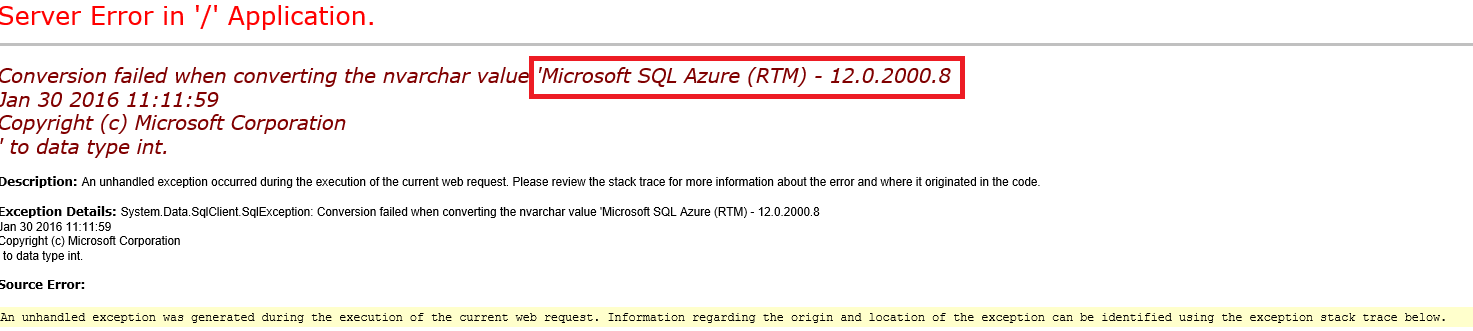
Bypass application login - ' OR 1 = 1 --





Generates exception with SQL server version        - ‘OR CAST(@@version as int) = 1 --

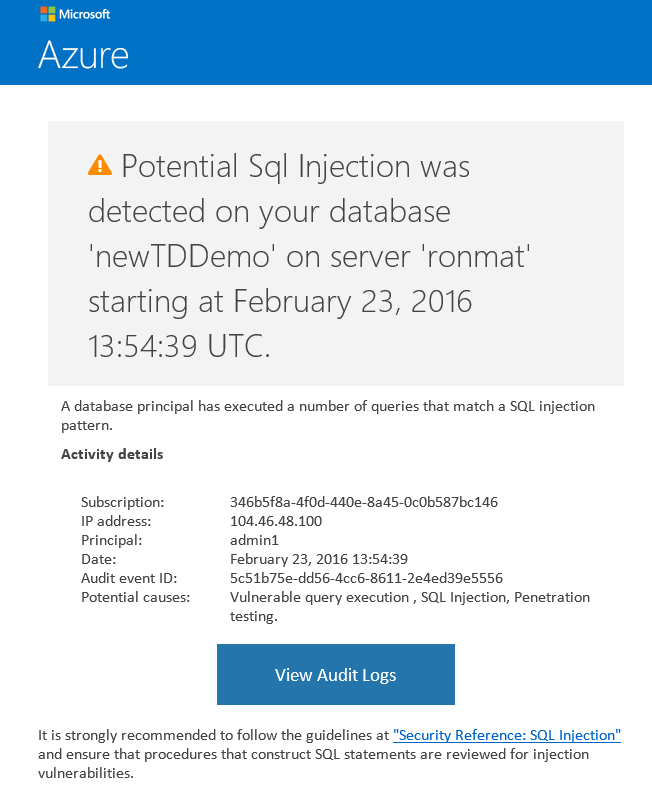




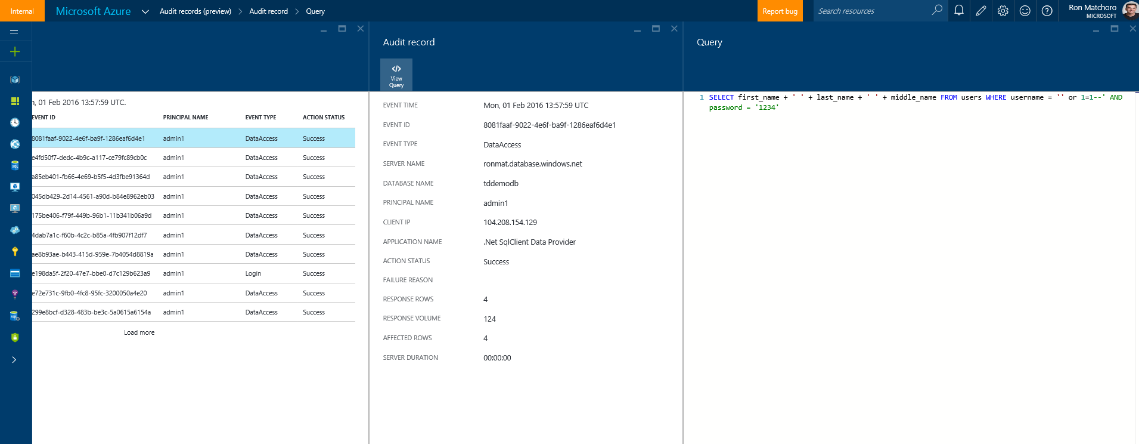
**STEP3: Detect & Explore**

**So the bad news is my app just got attacked. The good news is that Threat Detection alerts me to this in real-time, and helps me analyze and ultimately fix the issue. Let’s walk through that experience.**

I go to my inbox, and I get an email notification upon detection of the SQL injection on my database, which provides information on the security event, database name, server name and the event time. In addition, it provides information on possible causes and recommended actions to investigate and mitigate the potential threat to the database.



I will click on the Azure SQL Auditing Log button, which will launch the Azure portal and immediately bring me to the relevant Auditing records around the time of the suspicious event. I can click on the audit records to view more details on the suspicious database activities such as the SQL statement that was used to breach data from my database. Armed with this information, I can now go find and fix the vulnerable query in my application.



I can also click **Open in Excel** to open a pre-configured excel template for deeper analysis of the audit log around the time of the suspicious event.

So that’s our ‘Threat Detection’ feature. It’s incredibly easy to use, and it’s helping our customers combat one of the most common threats to their applications.

**Thank you very much.**