Josh Gillman

CIT 480 - Senior Capstone

Cloud Based Security System

Research Topic – Text Notifications

With an ever more connected society today, it’s hard to find someone without their phone, tablet, or another smart device on a daily basis. With this in mind, and with the goal of creating a security system where a user can have peace of mind with a place & forget camera, one feature that we would like to implement is the ability for a user to subscribe to text notifications so that they are aware of any activity captured by the camera as it happens.

This can be done in multiple ways using the agreed upon Linux platform for the web based portion of the system, but with Amazon Web Services plugins, this can be implemented into the site code to seamlessly integrate this feature. According to the AWS documentation, Amazon’s SNS can be used to send messages to any SMS-enabled devices. This can be done either using console commands or by using one of the AWS SDKs. For our purposes, we will be looking to use the SDK plugins as they will be used to augment our notification features and can be automated with code.

According to the documentation: “…an SMS message can be sent by using one of AWS SDKs, use the action in that SDK that corresponds to the “Publish” request in the Amazon SNS API. With this request, you can send an SMS message directly to a phone number.” (Sending an SMS Message - Amazon Simple Notification Service). Additional values can be set with more exact commands:

* AWS.SNS.SMS.SenderID
  + A custom ID that contains up to 11 alphanumeric characters, including at least one letter and no spaces. The sender ID is displayed as the message sender on the receiving device. (Sending an SMS Message - Amazon Simple Notification Service)
* AWS.SNS.SMS.MaxPrice
  + The maximum amount in USD that you are willing to spend to send the SMS message. Amazon SNS will not send the message if it determines that doing so would incur a cost that exceeds the maximum price. (Sending an SMS Message - Amazon Simple Notification Service)
* AWS.SNS.SMS.SMSType
  + The type of message that is being sent:
    - Promotional (default) – Noncritical messages, such as marketing messages. Amazon SNS optimizes the message delivery to incur the lowest cost. (Sending an SMS Message - Amazon Simple Notification Service)
    - Transactional – Critical messages that support customer transactions, such as one-time passcodes for multi-factor authentication. (Sending an SMS Message - Amazon Simple Notification Service)

These attributes can be set in the Java SNS SDK by coding a map that will associate the attribute keys with “MessageAttributeValue” objects, using this block of code:

Map<String, MessageAttributeValue> smsAttributes =

**new** HashMap<String, MessageAttributeValue>();

smsAttributes.put("AWS.SNS.SMS.SenderID", **new** MessageAttributeValue()

.withStringValue("mySenderID") //The sender ID shown on the device.

.withDataType("String"));

smsAttributes.put("AWS.SNS.SMS.MaxPrice", **new** MessageAttributeValue()

.withStringValue("0.50") //Sets the max price to 0.50 USD.

.withDataType("Number"));

smsAttributes.put("AWS.SNS.SMS.SMSType", **new** MessageAttributeValue()

.withStringValue("Promotional") //Sets the type to promotional.

.withDataType("String"));

A complete example of a message can be seen in the documentation, and for our purposes, this snippet of code will be used as a template for sending messages to our subscribers:

**public** **static** **void** **main**(String[] args) {

AmazonSNSClient snsClient = **new** AmazonSNSClient();

String message = "My SMS message";

String phoneNumber = "+1XXX5550100";

Map<String, MessageAttributeValue> smsAttributes =

**new** HashMap<String, MessageAttributeValue>();

//<set SMS attributes>

sendSMSMessage(snsClient, message, phoneNumber, smsAttributes);

}

**public** **static** **void** **sendSMSMessage**(AmazonSNSClient snsClient, String message,

String phoneNumber, Map<String, MessageAttributeValue> smsAttributes) {

PublishResult result = snsClient.publish(**new** PublishRequest()

.withMessage(message)

.withPhoneNumber(phoneNumber)

.withMessageAttributes(smsAttributes));

System.out.println(result); // Prints the message ID.

}

The idea will be to integrate the information from the user’s profile into the message send command, where “String phoneNumber” can be replaced with a variable set by the user, and the message will provide a hyperlink to their profile with the snippet of video captured that set off the alert.

The inclusion of this feature will bring about more questions, such as when to send the message to the user, either from the trigger being sent from the site or the camera, and then the sensitivity of the text inclusion, where to find out whether or not there is a way to calibrate when a user will receive a message or not. More research will have to be done to figure out the rest of these details in regards to this feature, but in regards to implementation, this will be a very simple feature to insert into our application.

Works Cited

Sending an SMS Message - Amazon Simple Notification Service. "Sending An SMS Message - Amazon Simple Notification Service". Docs.aws.amazon.com. N. p., 2017. Web. 16 Jan. 2017.