SMS OTP is not Secure Two Factor Authentication! Now what?

Aaron Poffenberger

BSidesDFW

2016-11-05

Overview

- Introduction
- The Problem
- Common Solutions
- What's Wrong with SMS OTP?
- Alternatives to SMS OTP
- Code Examples
- Conclusion
- Resources

Introduction

- Software developer
  - 17+ years professionally
  - Security software developer
  - Design and implement secure APIs
- InfoSec
  - Software vulnerability assessment
  - Auditing
  - CISSP 2005+
  - US Army
- IT Background
  - IT Admin
  - ISP (dial-up land)
  - DevOps

Introduction

- Founding member, Houston dc713
- Founding member, Houston Area Hackers Anonymous
- OpenBSD user
- Amateur radio enthusiast
- Electronics hobbyist

About You

- Red Team

About You

- Red Team
- Blue Team

About You

- Red Team
- Blue Team
- Developer

About You

- Red Team
- Blue Team
- Developer
- How many developers also raised your hands for red or blue teams?

The Problem

- Many sites have poor security, poor password crypto
- Bad guys break in and steal both
- Users reuse user ids and passwords across sites
- Bad guys use these credentials to move from less valuable to more valuable sites

Common Solutions

- Password-expiry intervals
- Password complexity requirements
- Two-Factor Authentication

A Brief Overview of Two-Factor Authentication

- Biometric
- Single-purpose devices (RSA SecurID)
- Smart-phone apps (Google Authenticator)
- Short Message Service one-time pad (SMS OTP)

Survey

- Who thinks SMS OTP is Secure?

Survey

- Who thinks SMS OTP is Secure?
- Who thinks SMS OTP is Good Enough?

Survey

- Who thinks SMS OTP is Secure?
- Who thinks SMS OTP is Good Enough?
- Who thinks SMS OTP is “better than nothing”?

Survey

- Who thinks SMS OTP is Secure?
- Who thinks SMS OTP is Good Enough?
- Is SMS OTP “better than nothing”?

Let’s see whether we can change a few opinions.

What is SMS OTP Two Factor Authentication?

- Sending a code, PIN or other one-time-use authentication token to a user's cell phone using the short-message service.
- The token is entered **without transformation** as a second authentication factor.
- Similar to sending authentication token via email.

Typical Use Cases

- Second factor during authentication
- Password reset OTP

SMS OTP Advantages

- In 2016 nearly everyone has a phone
- Nearly real time
- Cellphones are unique to users
- Easy for users to configure and use

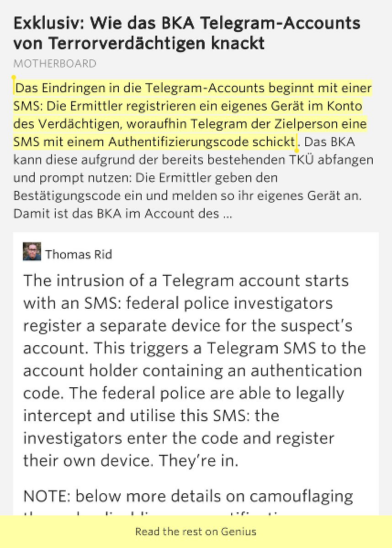
What's Wrong with SMS OTP?

- SMS standard does not require encryption
- SMS encryption standards have all been cracked
  - Can be decoded in real time using COTS hardware and software
- No forward secrecy
- Net effect: **Authentication code is sent in the clear**
- False sense of security
- Deprecated in current draft of NIST Special Publication 800-63-3, Digital Authentication Guideline
  - To be removed in future version

Defending SMS OTP

- The attacks aren't easy to effect
- SMS OTP is easy for end users to use and understand
- "Our system needs good enough, not perfect"
- "Our Site doesn't have anything of value"
- "SMS OTP is better than nothing"

Lawful Intercept



Exklusiv: Wie das BKA Telegram-Accounts von Terrorverdächtigen knackt

MOTHERBOARD

Das Eindringen in die Telegram-Accounts beginnt mit einer SMS: Die Ermittler registrieren ein eigenes Gerät im Konto des Verdächtigen, woraufhin Telegram der Zielperson eine SMS mit einem Authentifizierungscode schickt. Das BKA kann diese aufgrund der bereits bestehenden TKÜ abfangen und prompt nutzen: Die Ermittler geben den Bestätigungscode ein und melden so ihr eigenes Gerät an. Damit ist das BKA im Account des ...

Thomas Rid

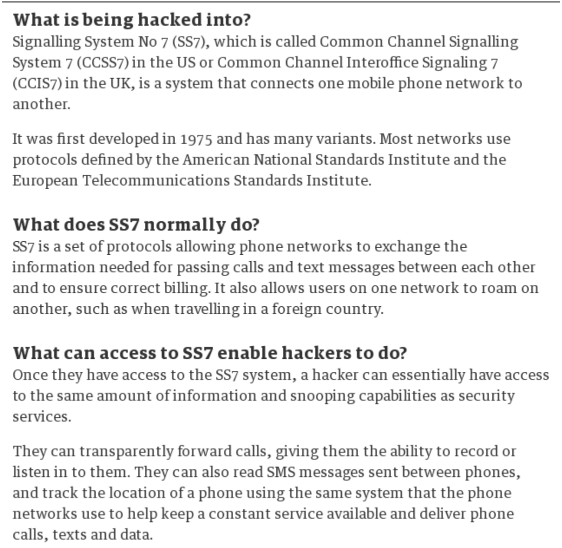
The intrusion of a Telegram account starts with an SMS: federal police investigators register a separate device for the suspect's account. This triggers a Telegram SMS to the account holder containing an authentication code. The federal police are able to legally intercept and utilise this SMS: the investigators enter the code and register their own device. They're in.

NOTE: below more details on camouflaging

Read the rest on Genius

Figure 1:

Using SS7 to Hack GSM Network



What is being hacked into?

Signalling System No 7 (SS7), which is called Common Channel Signalling System 7 (CCSS7) in the US or Common Channel Interoffice Signaling 7 (CCIS7) in the UK, is a system that connects one mobile phone network to another.

It was first developed in 1975 and has many variants. Most networks use protocols defined by the American National Standards Institute and the European Telecommunications Standards Institute.

What does SS7 normally do?

SS7 is a set of protocols allowing phone networks to exchange the information needed for passing calls and text messages between each other and to ensure correct billing. It also allows users on one network to roam on another, such as when travelling in a foreign country.

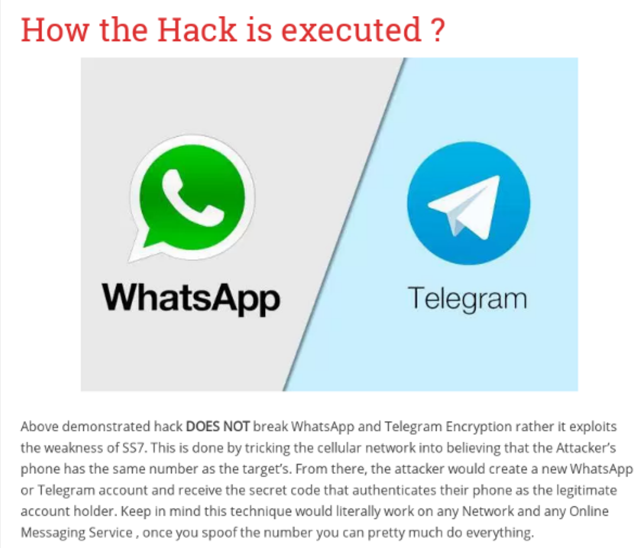
What can access to SS7 enable hackers to do?

Once they have access to the SS7 system, a hacker can essentially have access to the same amount of information and snooping capabilities as security services.

They can transparently forward calls, giving them the ability to record or listen in to them. They can also read SMS messages sent between phones, and track the location of a phone using the same system that the phone networks use to help keep a constant service available and deliver phone calls, texts and data.

Figure 2:

Using SS7 to Hack WhatsApp

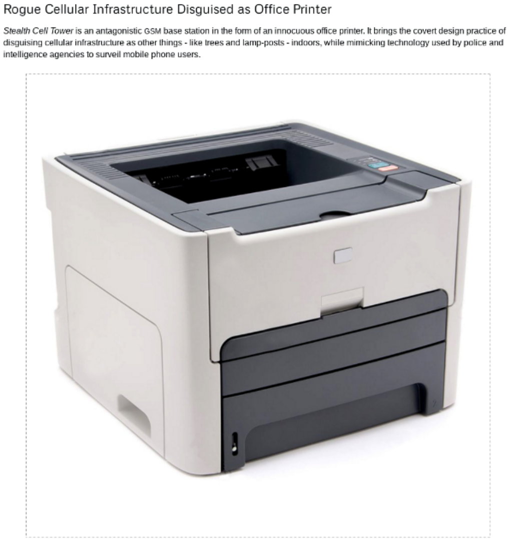


How the Hack is executed ?

Above demonstrated hack **DOES NOT** break WhatsApp and Telegram Encryption rather it exploits the weakness of SS7. This is done by tricking the cellular network into believing that the Attacker's phone has the same number as the target's. From there, the attacker would create a new WhatsApp or Telegram account and receive the secret code that authenticates their phone as the legitimate account holder. Keep in mind this technique would literally work on any Network and any Online Messaging Service , once you spoof the number you can pretty much do everything.

Figure 3:

Intercepting GSM Transmissions



Rogue Cellular Infrastructure Disguised as Office Printer

Stealth Cell Tower is an antagonistic GSM base station in the form of an innocuous office printer. It brings the covert design practice of disguising cellular infrastructure as other things - like trees and lamp-posts - indoors, while mimicking technology used by police and intelligence agencies to surveil mobile phone users.

Figure 4:

Intercepting GSM Transmissions

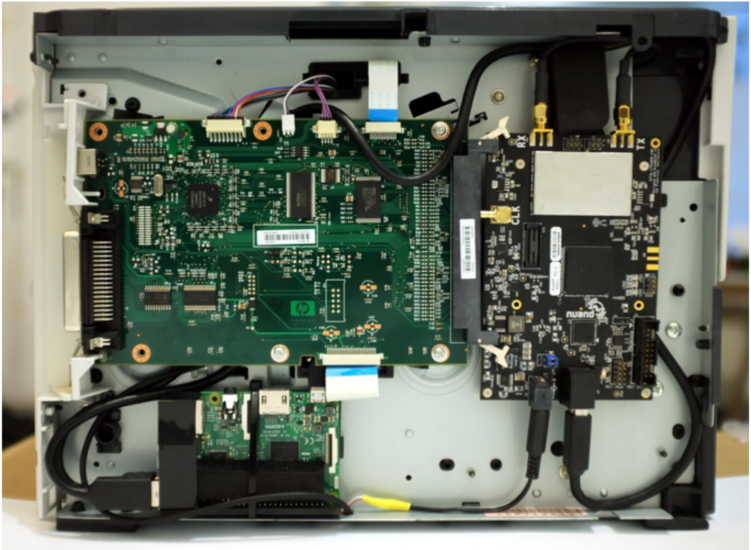
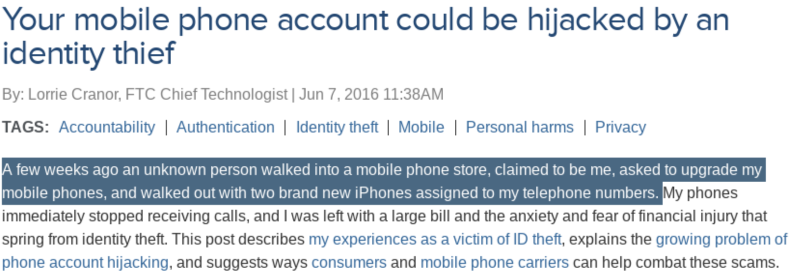


Figure 5:

Identity Theft



Your mobile phone account could be hijacked by an identity thief

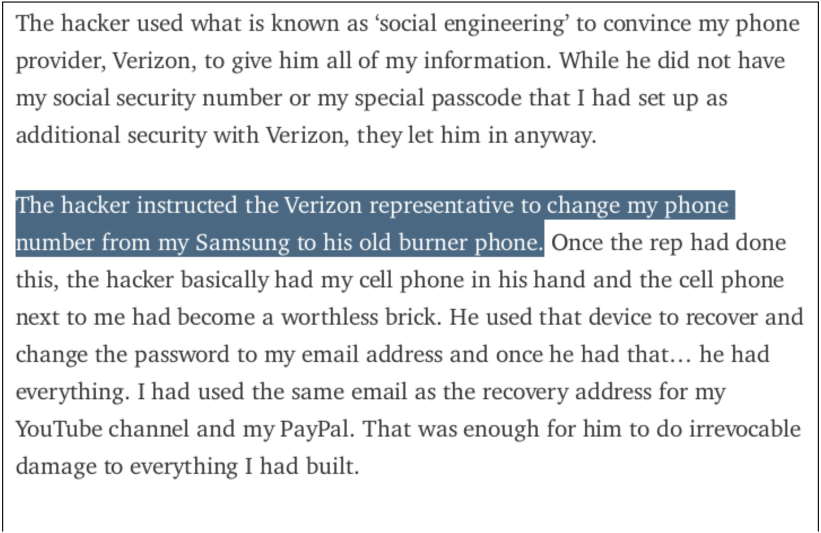
By: Lorrie Cranor, FTC Chief Technologist | Jun 7, 2016 11:38AM

TAGS: Accountability | Authentication | Identity theft | Mobile | Personal harms | Privacy

A few weeks ago an unknown person walked into a mobile phone store, claimed to be me, asked to upgrade my mobile phones, and walked out with two brand new iPhones assigned to my telephone numbers. My phones immediately stopped receiving calls, and I was left with a large bill and the anxiety and fear of financial injury that spring from identity theft. This post describes my experiences as a victim of ID theft, explains the growing problem of phone account hijacking, and suggests ways consumers and mobile phone carriers can help combat these scams.

Figure 6:

Social Engineering I

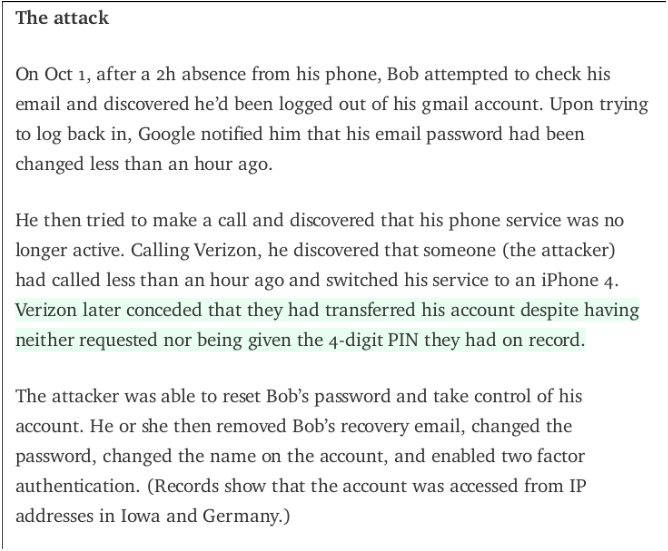


The hacker used what is known as ‘social engineering’ to convince my phone provider, Verizon, to give him all of my information. While he did not have my social security number or my special passcode that I had set up as additional security with Verizon, they let him in anyway.

The hacker instructed the Verizon representative to change my phone number from my Samsung to his old burner phone. Once the rep had done this, the hacker basically had my cell phone in his hand and the cell phone next to me had become a worthless brick. He used that device to recover and change the password to my email address and once he had that… he had everything. I had used the same email as the recovery address for my YouTube channel and my PayPal. That was enough for him to do irrevocable damage to everything I had built.

Figure 7:

Social Engineering II



The attack

On Oct 1, after a 2h absence from his phone, Bob attempted to check his email and discovered he'd been logged out of his gmail account. Upon trying to log back in, Google notified him that his email password had been changed less than an hour ago.

He then tried to make a call and discovered that his phone service was no longer active. Calling Verizon, he discovered that someone (the attacker) had called less than an hour ago and switched his service to an iPhone 4. Verizon later conceded that they had transferred his account despite having neither requested nor being given the 4-digit PIN they had on record.

The attacker was able to reset Bob's password and take control of his account. He or she then removed Bob's recovery email, changed the password, changed the name on the account, and enabled two factor authentication. (Records show that the account was accessed from IP addresses in Iowa and Germany.)

Figure 8:

Survey Revisited

Who wants to change their answers?

- Who thinks SMS OTP is Secure?
- Who thinks SMS OTP is Good Enough?
- Is SMS OTP “better than nothing”?

“We Need Good Enough, Not Perfect”

If what your protecting is important enough to need two factors of authentication, doesn’t it need a secure second factor?

Or are you cargo culting?

"Our Site doesn't have anything of value"

Everyone is thinking about lateral movement through systems. **Who's thinking about lateral movement through people and organizations?**

That is:

Who can the bad guys get to by going through the people and organizations they're connected to?

Your site might be the first step. The users themselves may be the value.

If you're going to do security, do it right for the scenario under consideration.

“SMS OTP is Better than Nothing”

SMS OTP is not “better than nothing”.

“SMS OTP is Better than Nothing”

SMS OTP is not “better than nothing”.

Depending on the site and the attack, **it’s worse than nothing**.

Alternatives to SMS OTP

- None
- Proprietary Protocol / Write You Own
- Third-party app using secure protocol
- In-App

None

- Default solution for most sites
- Do you really need two-factor authentication?
- Seriously, do you need two factor?
- What problem are you trying to solve?
- Is it a security problem or a usability problem? Both?

None – Pros

- User only has to remember login id and password
- Shared responsibility for security of account
- Great for users who don't have cell phones

None – Cons

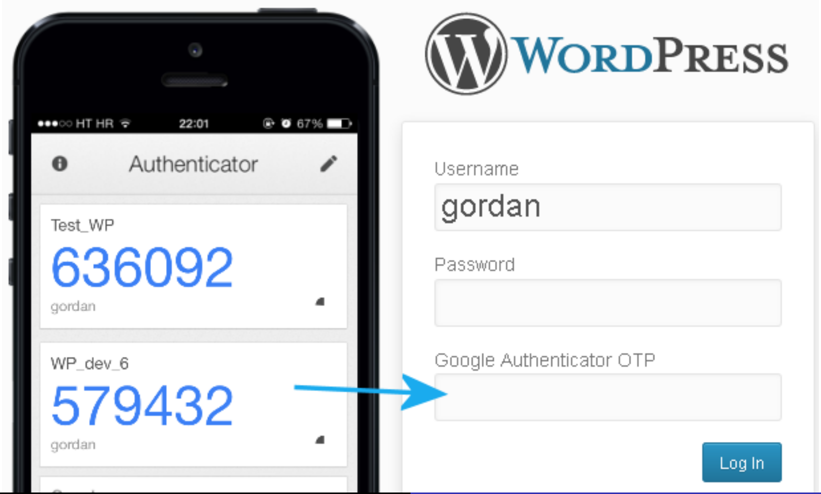
- Places onus of security on user
- Need lost-id and lost-password processes
- Can't defend against lateral movement across sites when credentials are compromised
- May require user to store one or more "emergency" passwords

Proprietary Protocol / Write Your Own

- Don't!
- Stick to well-reviewed protocols with multiple, compatible implementations
- You wouldn't devise your own crypto algorithm . . . would you?

Third-party Apps

Delegate second-factor token generation to another app or service provider like Google Authentication or Authy.



Username

gordan

Password

Google Authenticator OTP

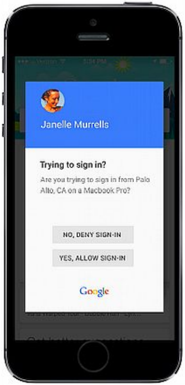
Third-party Apps – Pros

- User may already have app installed
- Main contenders (Google Authenticator and Authy) standards based
- Authy can syncrhonize shared secrets across devices and has a desktop app
- Doesn't require developing a custom app just for two factor
- CLI tools (for the geeks)

Third-party Apps – Cons

- Often requires per-site shared secrets
- Not as easy as sending a text message
- Losing shared secret can lead to lockout
- May require user to store one or more “emergency” passwords
- Replacing shared secrets is even harder (more steps) than changing a password

In-App – Google Two Step



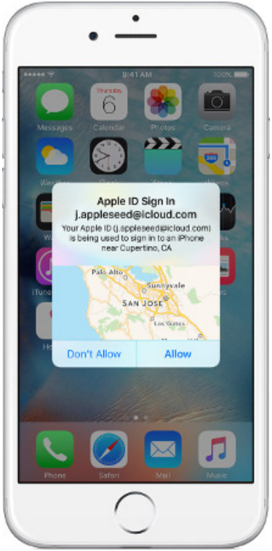
Janelle Murrells

Trying to sign in?

Are you trying to sign in from Palo Alto, CA on a Macbook Pro?

Figure 10:

In-App – Apple ID Sign In



Apple ID Sign In

j.appleseed@icloud.com

Your Apple ID (j.appleseed@icloud.com) is being used to sign in to an iPhone near Cupertino, CA

Figure 11:

In-App – Pros

- Secure-able
- Good option for sites whose users typically or must install the app (banks)
- Easy for users to use and understand

In-App – Cons

- Requires user to install a unique app per site
- Not as easy as sending a text message
- Deleting app or lost phone can lead to lock out
- May require user to store one or more “emergency” passwords
- Need a re-keying strategy for shared secret

In-App Solutions

- Authorization Request / Approve (Apple and Google style)
- Counter based: HOTP - RFC 4226
- Time based: TOTP - RFC 6238

HOTP Up Close

[A]n algorithm to generate one-time password values, based on Hashed Message Authentication Code (HMAC). - RFC 4226

$$HMAC(K, m) = H\Big((K' \oplus opad) \mid\mid H\big((K' \oplus ipad) \mid\mid m\big)\Big)$$

Figure 12: (Wikipedia)

- Easy to implement, libraries for most languages (see Resources)
- Uses keyed-hash message authentication codes where the message to be hashed is a non-repeating counter
- Output is 6 or 8 digits

TOTP Up Close

[A]n extension of the One-Time Password (OTP) algorithm, namely the HMAC-based One-Time Password (HOTP) algorithm, as defined in RFC 4226, to support the time-based moving factor. - RFC 6238

- HOTP using timestamps as the message to be hashed
- Easy to implement, libraries for most languages (see Resources)

Python – Time Based

```
import pyotp

totp = pyotp.TOTP('base32secret3232')
totp.now() # => 492039

# OTP verified for current time
totp.verify(492039) # => True
time.sleep(30)
totp.verify(492039) # => False

# HOTP example similar
# See examples in pyotp
```

Python – Using Google Authenticator Secret Key

```
import pyotp

totp = pyotp.TOTP("JBSWY3DPEHPK3PXP")
print("Current OTP:", totp.now())
```

Conclusion

- SMS OTP is **NOT** Secure Two-Factor Authentication
- Sites should consider whether they need two-factor
- Two-factor authentication can be done via cell phone
- Third-party apps using secure protocols are available
- Secure protocols for in-app two-factor authentication exist
- Libaries for secure two-factor authentication are plentiful and easy to use

Questions

- Contact:
  - Twitter: @akpoff
  - Email: akp@hypernote.com
  - Blog: http://akpoff.com/
  - Github: https://github.com/akpoff/

Resources

- Very select and incomplete list of resources
- Google is your friend
- InfoSec people are your friends

Short Message Service Specification

- Specification
- A5/1 Stream Cipher (54-bit)
- A5/2 Stream Cipher (the weaker version of A5/2)
- A5/3 "Kasumi" block cipher (better but not great)

Intercepting SMS Messages

- Lawful intercept
  - Thomas Rid's Summary Tweet
- Hacking WhatsApp using SS7 - Live Demo
  - Explaination of the live demo the SS7 hack
  - Guardian article explaining SS7 hack
- New Attack Allows Intercepting or Blocking of Every LTE Phone Call And Text
- Slides from Ruxcon presentation
- Rogue Cellular Infrastructure Disguised as Office Printer

Social Engineering and Identity Theft

- Your mobile phone account could be hijacked by an identity thief
- Adding a phone number to your Google account can make it LESS secure
- Getting Hacked As An Internet Creator

RFCs and Standards

- Draft Special Publication 800-63-3: Digital Authentication Guideline
- RFC 2104 - HMAC
- RFC 4226 - HOTP
- RFC 6238 - TOTP
- Wikipedia Explanation of HMAC
- Wikipedia Explanation of HOTP
- Wikipedia Explanation of TOTP

TOTP/HOTP Clients

- Authy
- Google Authenticator
- oathgen - A command line HOTP and TOTP client
- OATH Toolkit

TOTP/HOTP Libraries

- C/C++ - Google Authenticator
- C/C++ - OpenOTP
- C# - OATH.NET
- C# - OTP
- Haskell - One Time Password
- Java - libotp
- Java - oath
- Javascript - JS-OTP

TOTP/HOTP Libraries (cont.)

- Node.js - speakeasy
- Perl - Authen::OATH
- Python - pyotp
- Python - oath
- php - otphp
- php - Otis
- Ruby - rotp