



Creating One Time Password (OTP) infrastructures using Open Source software

Giuseppe “Gippa” Paternò

**Visiting Researcher
Trinity College Dublin**



Fàilte
(welcome)

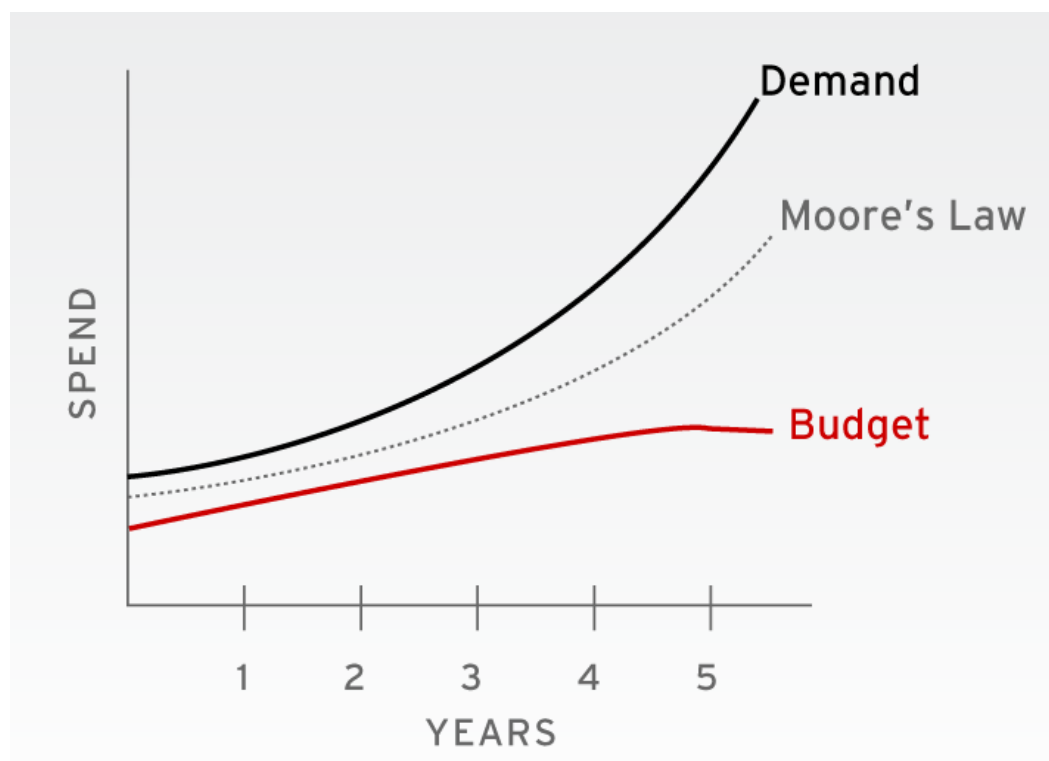


Who am I

- Visiting Researcher at Trinity College Dublin (Ireland)
- Solution Architect and EMEA Security Expert in Red Hat
- Previously Security Solution Architect in Sun and also in IBM
- Red Hat Certified Security Specialist (RHCSS), Red Hat Certified Architect (RHCA) and Cisco Certified Network Professional (CCNP)
- Part of the italian security community *sikurezza.org*
- Published books and whitepapers
- Forensic analysis for local govts
- More on:
 - <http://www.scss.tcd.ie/Giuseppe.Paterno/>
 - <http://www.gpaterno.com/>
 - <http://www.linkedin.com/in/gpaterno>



Global IT scenario



- Even more in this recession phase, the IT budget is getting lower and lower
- The projects (demand) are increasing with significantly less money available



Lowering TCO

"The economic crisis is going to be a catalyst for open source, much like the technology crash of 2001 catapulted Linux front and center"

Laurie Wurster, a Gartner analyst.

The adoption of Open Source
software can lower the TCO
... and increase your security!



How Open Source can increase Security?



Open Source = Open Standards = Choice



The OATH Alliance

- The Initiative for Open Authentication (OATH)
- Open alliance of vendors
 - ActiveIdentity, Vasco, Gemalto, Aladdin, ...
- <http://www.openauthentication.org/>
- Created a common algorithm for one time password tokens (HOTP)
 - A common “protocol” for the interoperability of the several implementations available



What is HOTP

- An HMAC-Based One-Time Password Algorithm (HOTP)
- A common shared algorithm that is meant to facilitate the adoption of two-factor authentication
- Algorithm published as RFC 4226
- The complete standard on:
 - <http://www.rfc-editor.org/>



HOTP: Internals

The algorithm is:

$$\text{HOTP}(K,C) = \text{Truncate}(\text{HMAC-SHA-1}(K,C))$$

- | | |
|------------|--|
| K | Shared key between client and server |
| C | 8-byte counter value synchronized between client and server |
| Truncate() | Perform a dynamic truncation and reduction of the string to extract a 4-byte dynamic binary code.

The result must extract minimum a 6-digit code, but also 7 and 8-digit code |



Anatomy of HOTP

- The shared key between the OTP peers (token and authenticator) is an hexadecimal string
 - The length is a SHA-1 digest
- Example of generating a new HMAC 6-digit shared key:

```
dd if=/dev/random bs=4096 count=1 2>/dev/null |  
sha1sum | awk '{print $1}'
```



HOTP implementations

- Both commercial and open source implementations available
- Most of the hardware tokens adhere to the HOTP algorithm
- Few software implementations, most of which proprietary/closed source
- Some software client available:
 - J2ME, iPhone and Windows Mobile
 - Publically available algorithm makes it simple to implement a client



How does it fit all together?



The software

- An open source OTP server:
 - Only one server implementation available (OTPD), formelly from TRI-D Systems
 - Now I made it available on <http://otpd.googlecode.com>
- FreeRADIUS, the popular radius server for Linux
- Two tested freely available client:
 - oathdsss.jar (DSSS) for Java MIDP (Nokia)
 - iToken (Quest Software) for iPhone



OTPD server

- It handle the validation of the One Time Passwords
 - Uses files and LDAP as repository
- Keeps the state of the OTP token (counter)
- Supported tokens:
 - HOTP
 - CRYPTOCard
 - Plain old x9.9 (based on DES, unsecure!)
- It listen to authentication requests



FreeRADIUS

- Well known high-performance open source RADIUS server
 - Handle authentication and accounting
 - Plug-in based
- One of the plug-in is **rlm_otpd**
 - Developed by TRI-D Systems
 - Communicate via Unix sockets with the OTPD server to verify an OTP token

The soft-token

- An OTP token in software
- Less “secure” than an hardware
 - What if my laptop is stolen?
- A compromise is using a soft-token on a mobile platform
 - Easy to manage
 - Lower costs
 - Better security over a “fat” client on laptops/desktops
 - Available for most mobile phones



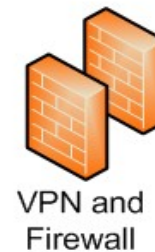
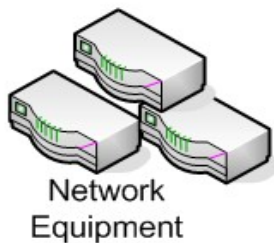
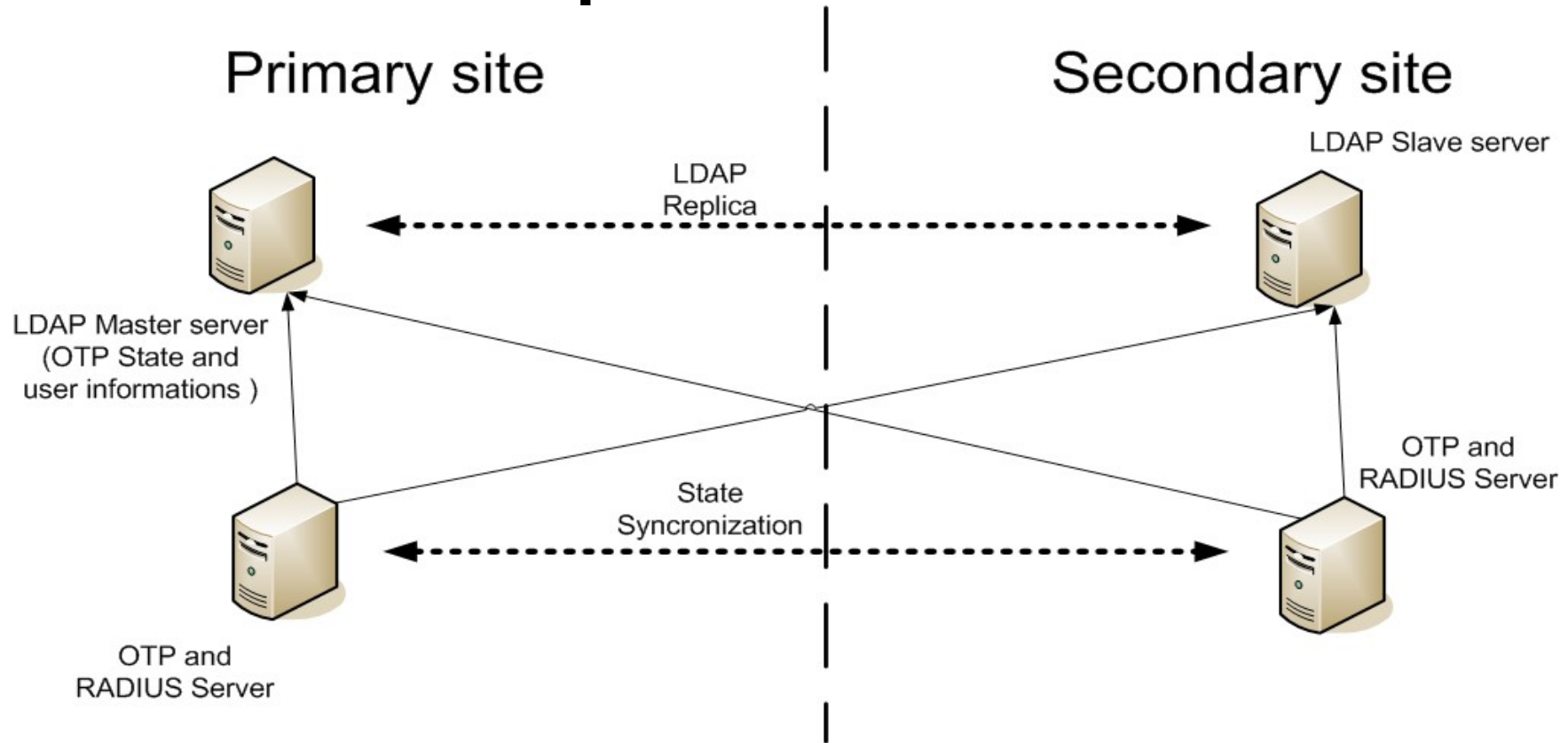


What can I authenticate?

- Any RADIUS compliant system, ex:
 - VPN systems
 - Wireless LANs
 - Routers/network equipments
 - Core UNIX systems (through pam_radius)
 - Captive portals
- Any application can use the RADIUS protocol:
 - common APIs available in C, PHP, Python, Ruby, Java (J2EE)



Enterprise scenario





Demo scenario

- Authentication server:
 - OTP Server
 - FreeRADIUS Server
- Client UNIX
- Web application (PHP)
- Centralized Web Single Sign-On (CAS)



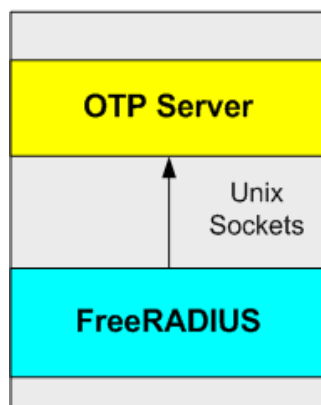
Demo (the clients)

- Client Unix
 - Interactive log-in
 - Leverage the pam_radius module
- Web Single Sign-On
 - Based on Yale CAS
 - Customized to login through RADIUS
- PHP web application
 - Dummy application to demonstrate CAS' capabilities with OTP integration
 - Virtually every application can leverage CAS architecture

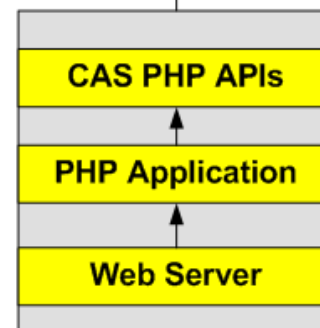
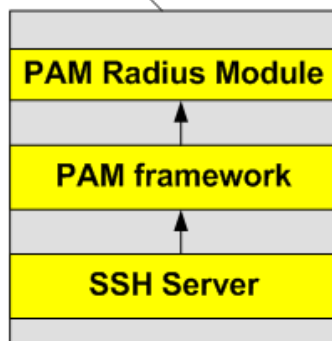
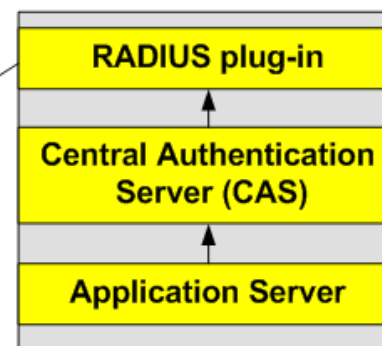


Demo scenario (big picture)

Authentication Server

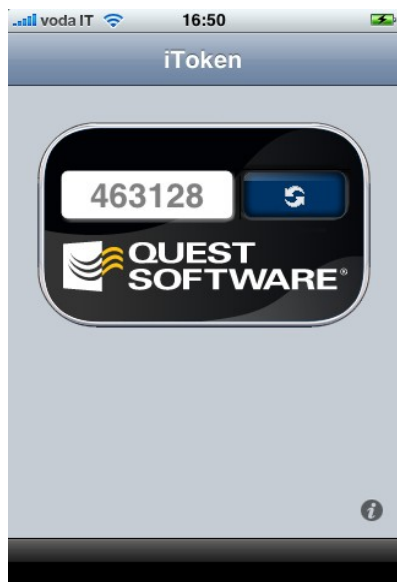


Web Single Sign-On



Unix Server

Web Server

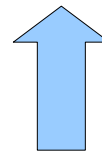




Interactive log-in



OTP/RADIUS Server



Authentication
Request
(RADIUS)



Log-on
request

```
BusyBox v1.2.1 (2006.12.10-00:34+0000) Built-in shell (ash)
Enter 'help' for a list of built-in commands.

-----
 | .-----.| | | | | .-----.| | | | | | | |
 | -  ||  |  _|| | | | |  ||  |
 |-----|| | | | | | | | | | | |
 |   | | WIRELESS  FREEDOM
 |   | | WHITE RUSSIAN (0.9) -----
 |   | | * 2 oz Vodka  Mix the Vodka and Kahlua together
 |   | | * 1 oz Kahlua  over ice, then float the cream or
 |   | | * 1/2oz cream milk on the top.
 |-----|

root@Quantumbase:~$ cd
root@Quantumbase:~$ cd /
root@Quantumbase:/$ ls
bin  dev  etc  jffs  lib  mnt  proc  row /sbin  top  usr  var  www
root@Quantumbase:/$
```



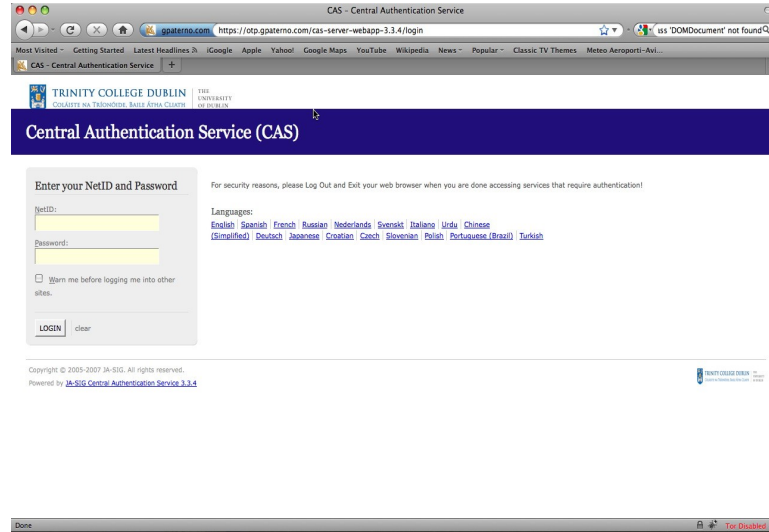
Web Application



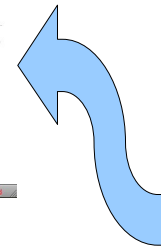
OTP/Radius Server



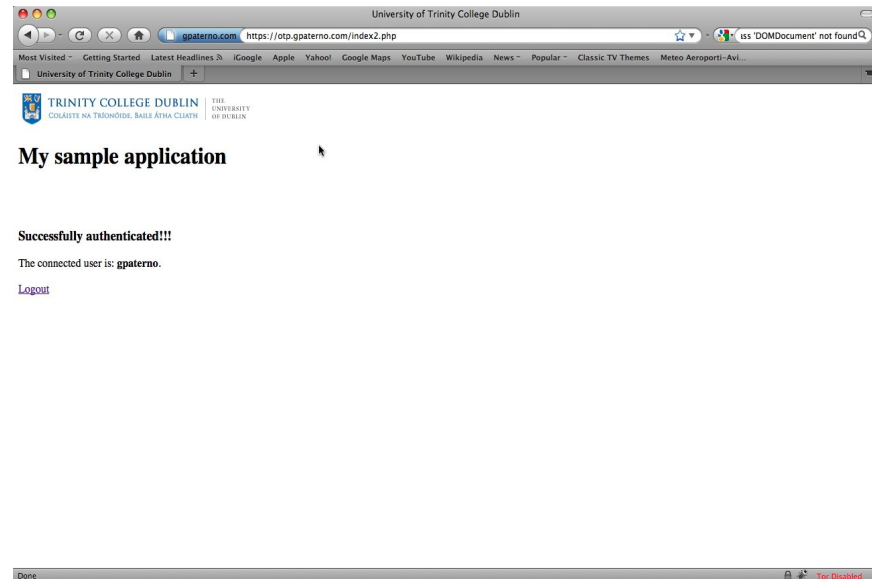
Authentication Request (RADIUS)



Redirect to CAS' Single Sign-on Portal



Web Access





Demo now!



Thank you!!

Giuseppe “Gippa” Paternò
Visiting Researcher
Trinity College Dublin

paternog@cs.tcd.ie

<http://www.scss.tcd.ie/Giuseppe.Paterno/>

<http://www.gpaterno.com/>