UES Uruguay - Personal Data Leak / Negligence

This document contains sensitive information obtained when researching a large data leak of private personal data. It is confidential and cannot be shared outside of the researcher and the government authorities which were directly sent this report by the researcher. Please contact the researcher listed below for permission to share this document to other parties. All notified agencies are listed in this document.

This document has a table of contents, you can enable viewing of the table of contents for easier viewing.

META

> **revision:** `2019-07-23.007`
> documented **July 14, 2019**
> updated **July 23, 2019**

Researcher:

> **Jayson Minard**
> [redacted]
> in **Montevideo, Uruguay**
> discovery frame **June 20 - July 12**

Parties Informed:

Date	Party
Jul 12, 2019	**UES** - source of leak CEO **Sebastian Salveraglio** [redacted] IT Director **Daniel Fernandez** [redacted] via phone conversation w/presentation via follow-up emails including **link to Uruguayan law 18.331**
Jul 15, 2019	**AGESIC** informed via this document to: [redacted] Asesor Juridico de la Unidad Reguladora y de Control de Datos Personales [redacted] **File 2019-299**
	Banco Itaú *affected party*

Date	Party
July 16, 2019	Notification by WhatsApp, meeting TBD [redacted]
July 17, 2019	**Banco Itaú** *affected party* Notification by Meeting (*email follow-up*) [redacted]
July 23, 2019	**CERTuy** via cert@cert.uy on advice of [redacted] update to both AGESIC and CERTuy of latest document
TBD	Foreign authorities
TBD	Other Involved Parties
TBD	Public affected parties

Data Retained:

Acquired	Disposition	Description
June 20	temporarily retained evidence	Start crawl of UES package tracking data 858K delivery records Corporate data of UES Personal data of clients (name, address) *Retain until hand-off to AGESIC, then delete*
July 12	temporarily retained evidence	Start crawl of open FTP via HTTP site Full index page **listing ~2,088,195 images** Copied sample of **~146,000 actual images** Personal data of clients (name, address, cedula, aclaration, signature, private documents) *Retain until hand-off to AGESIC, then delete*
July 14	retained as part of reporting document	Included sample data/images in this documentation which are intentionally not sanitized when the target audience are the government reporting agencies. **Not for public disclosure.**

Overview

In late June of 2019, I noticed an obvious leak of personal information via the UES Uruguay http://ues.com.uy. In their package tracking page they show more details than neccessary to accomplish the job of tracking a package. They disclose the sender's name, receipient's name, and full address of the receipient.

On further inspection I saw other clues that led me to believe the thinking of the company in terms of data security and privacy was at least inmature, and likely lacking or absent. **And this led to a series of descoveries that uncovered millions of points of personal data exposed publically on the internet.**

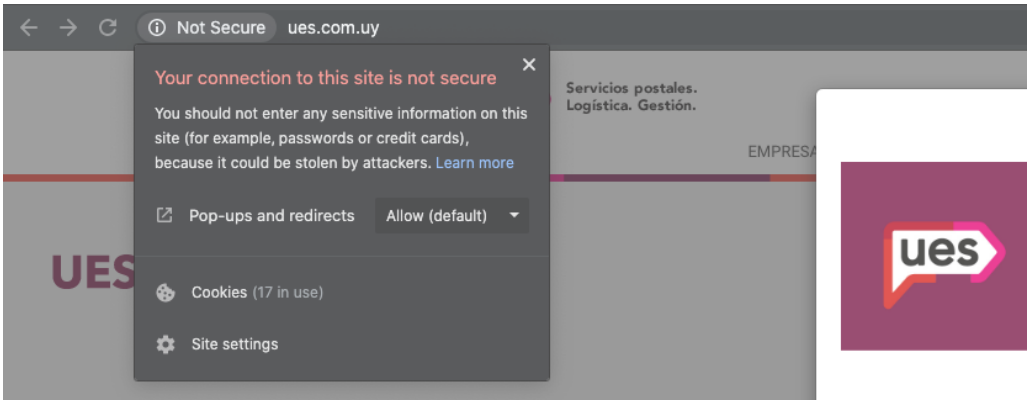
Note: No hacking was done, no penetration of the system, no steps required a login to the application. Any opinions related to legal matters are layman opinions and could be a misinterpretation of the law.

I will walk through each part of the site that exposes data, starting with simple and proceeding to the most dangerous security and data management problems with their systems.

Website in General

One the UES website, there is...

- No secure connection (HTTPS) for major portions of the site (including login pages)
- No terms of use on the site
- No terms of use on the tracking page
- No captchas
- No restrictions on volume or rate of requests
- No robots.txt guiding crawlers
- The use the verb "BUSCAR" on buttons indicating the form subtmital is a search function, which could attact search engine crawlers
- A friendly scan of the site by a security scanner would show most if not all securiry best practices are **not** followed.



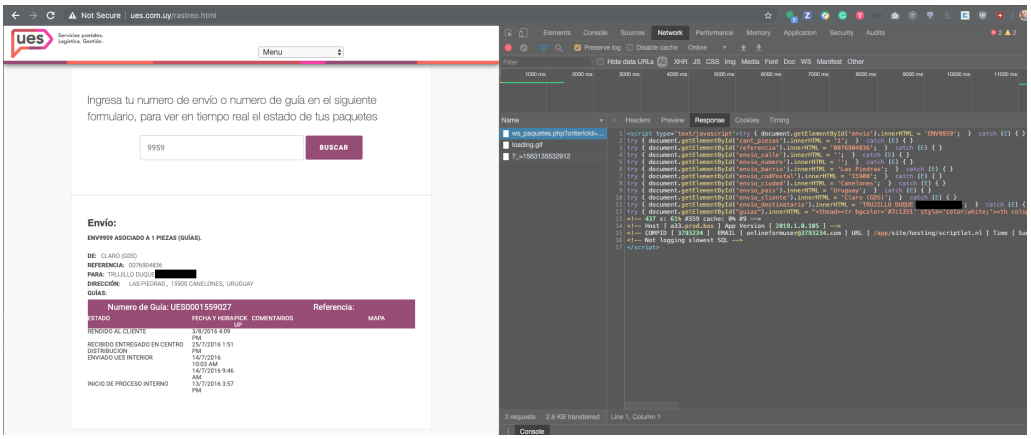
Package Tracking

Overview

The package tracking page presents publically available data, without restriction. When tracking a package on http://ues.com.uy/rastreo.html, it discloses:

- The name of the sender (typically an e-commerce company, but sometimes a mutualista or government agency)
- The reference number of the sender (more about the problem with this later)
- The personal data of the receiver (name, address, city, postal, state, country)
- The internal state movements of the company, beyond what is neccessary for the consumer

In addition the tracking number system uses predictable tracking numbers which is just a sequential numeric value that increases for each sent package. It also appeared that **all data** since their initial shipments is available online.



Their backend system is PHP based and instead of returning data to be rendered onto the page, it returns JavaScript code that is executed and modifies the original page to show the content. Therefore a crawler that has a headless browser with JavaScript support can easily crawl all of the content and extract the results. I started such a crawler to see how extensive the data set was. In around two weeks, the full captured data was available for analysis. Here is what was encountered.

- Numeric range of values from 160 consecutively through the current 1,034,999 for tracking numbers.
- Found data for **858,031 deliveries** of 940,217 packages in total
- Date range of packages run from **Feburary 2016 until current**.
- All of their delivery orders are prefixed with `ENV` followed by the sequential number.
- Each package shipped has a separate Guía, in the pattern of `UES` followed by a numeric value of fixed width.
- Limited number of senders of packages, around 368 who are mostly e-commerce clients.
- Fairly dirty name and address data for receipients with:
    - **338,256 distinct receipient names**
    - 308,827 unique addresses (after minimal cleaning)
    - **467,956 unique combinations of receipient names + addresses**

- Packages deliever to the following departamentos in Uruguay
    - Montevideo (494,548)
    - Canelones (93,909)
    - Maldando (36,223)
    - Colonia (25,810)
    - Salto (19,564)
    - Paysandu (19,086)
    - Rocha (17,868)
    - San Jose (16,797)
    - Tacuarembo (16,354)
    - Soriano (14,823)
    - Cerro Largo (13,601)
    - Rivera (13,339)
    - Florida (13,107)
    - Trienta y Tres (12,353)
    - Lavalleja (12,271)
    - Artigas (11,269)
    - Rio Negro (11,262)
    - Durazno (11,063)
    - Flores (4,620)
- Internal transition states and timings of UES delivery steps (fairly messy with 122 statuses)

This data is added by UES when they ship a package, with the data provided by the sender. The receipient may not be aware of their data being retained in a database, especially indefinitely. Given this open tracking system it is possible for bad actors to:

- Determine who is buying from which vendor
- Advertise to the clients of your competitor (as an e-commerce company)
- Build contact and marketing mailing lists from the persons present in the system
- Predict when packages were arrive at intended destinations and try to intercept them
- Analyze the comparital sales of each client e-commerce company
- Determine who is an “e-commerce ready” person for marketing purposes
- Analyze geographic buying patterns of Uruguayans (by postal zone, barrior, city, state, region)
- Analyze the internal state transitions and timings of the UES package delivery system

This database is unlawful in that:

(*ley 18.331 of 2008 and updates since*)

- it is not registered as a database containing personal data (*which in process includes a lot of questions that are indicators that you have designed a manageable and secure system*)
- it is holding data beyond the need and purpose of having the data
- the system exposes the private personal data publically
- any attempt to protect the data in other systems are defeated by this intentional public display of the data, and it is illegal to hold personal data without security

- there is no obvious manner to remove yourself from this database
- By design, not by accident, they are exposing personal data

By Uruguayan law, consent is not required to build a database if it is limited to names and addresses, but that does not exclude the other provisions of protecting the data and limiting the lifespan of data retention.

This data likely violates GDPR laws of Europe since Eurpean citizens use this service and are part of the database. Uruguay also has a law cooperating with some European other laws on data protection (*ley 19.030 - Approval of Convention 108*).

Regardless, UES has an obligation to inform AGESIC of the exposed data, and the affected parties who's personal data is exposed.

Affected parties:

Without clean data it is hard to be sure, but it is clear the 368 clients sending packages through this system have their data leaked and that of their customers. Also some large percentage of the 467,956 unique names+addresses represent affected parties. In effect, the "general public" of anyone ever receiving a package via UES is included, and of that set many may not even know the name of the delivery service that knocked at their door.

Actions:

UES was notified of this data leak on July 12, 2019. Attending were **Sebastian Salveraglio** (Generente General de UES) and **Daniel Fernandez** (IT Director, Innovation Manager). Others might have been present on the call as video was disabled on the part of UES.

as of July 12, 2019

Exact mechanisms of the leak, crawl of the data, implications of predictable tracking numbers, exposure of personal data, exposure of client data, exposure of UES corporate data were all disclosed.

as of July 14, 2019

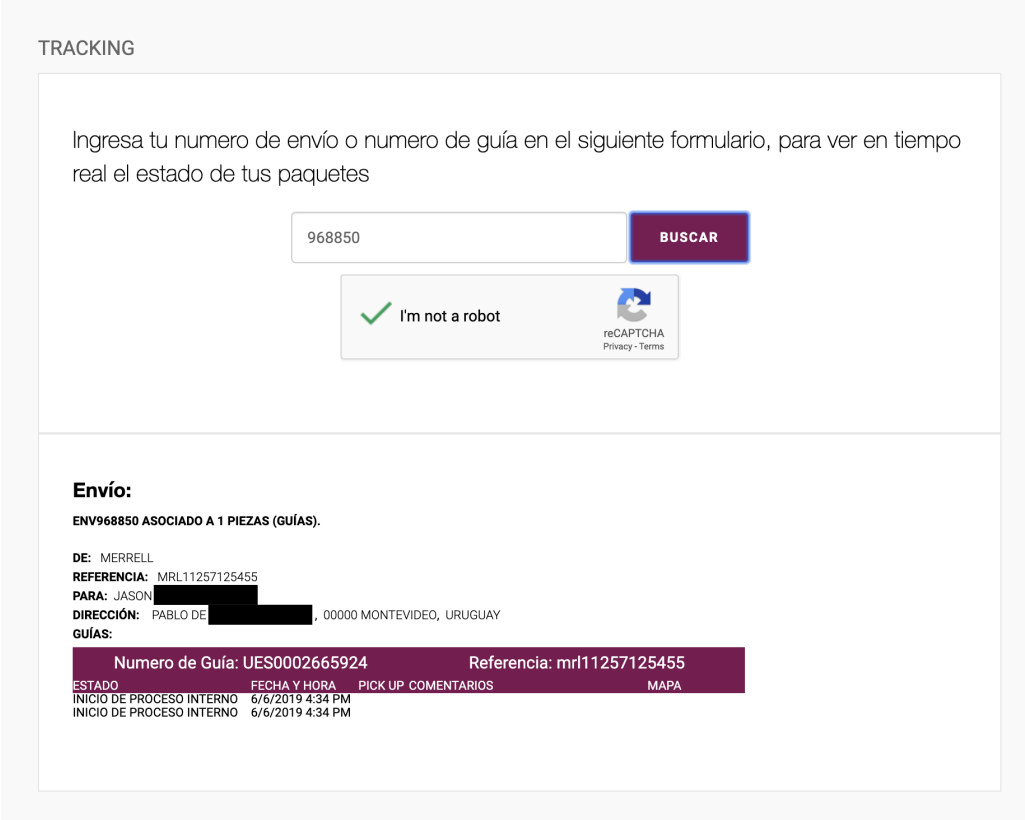
Researcher making this discovery and writing this document with intent to preserve evidence as to the scope and contents of the leak for reporting to government authorities, AGESIC, GDPR, etc. No further reporting is needed to UES.

as of July 15, 2019

AGESIC informed (*see notifications header*)

as of July 16, 2019

UES has added a captcha to the package tracking page preventing crawling without reCAPTCHA circumvention / solving. Full personal data is still showing:



as of July 19, 2019

[redacted]

as of July 21, 2019

UES terminates this aspect of the site and returns a message about site maintance or error 500's depending on the part of the site. [redacted]

"En Mano" Service and System

Overview

UES apparently has a delivery service for important documents for which the majority must be presented

“in hand” to the intended receipient or valid alternative receipient. This is used by banks, credit card companies, legal entities, and more. This feature is managed by the clients within the http://ues.com.uy/enmano section of the website. Which upon, visiting you are immediately presented with a login page.

The first thing to notice is that the page is **not running HTTPS** and therefore is a very insecure login page. The second thing to notice is how other pages redirect to this login page incorrectly. For example, crawling all links of the UES website, turns up some interesting pages within the `enmano` namespace. One of those is `lista_urusarios.php`. Going to that page redirects to the login page in a web browser. We can do the same within the `curl` command and receive this output:

```
ues > curl -vs "http://ues.com.uy/enmano/lista_usuarios.php" > temp.html
*   Trying 190.64.204.63...
* TCP_NODELAY set
* Connected to ues.com.uy (190.64.204.63) port 80 (#0)
> GET /enmano/lista_usuarios.php HTTP/1.1
> Host: ues.com.uy
> User-Agent: curl/7.64.1
> Accept: */*
>
< HTTP/1.1 302 Moved Temporarily
< Date: Mon, 15 Jul 2019 01:05:01 GMT
< Server: Apache
< X-Powered-By: PHP/5.6.40
< Expires: Thu, 19 Nov 1981 08:52:00 GMT
< Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
< Pragma: no-cache
< Set-Cookie: PHPSESSID=47b1a0328f329722e911fdbce7f9b5fa; path=/
< Location: index.php
< Transfer-Encoding: chunked
< Content-Type: text/html; charset=UTF-8
<
{ [2333 bytes data]
* Connection #0 to host ues.com.uy left intact
* Closing connection 0
```

Curl does not follow redirects by default, so even though we see the redirect header being returned, Curl stops at this point. Well, it doesn’t completely stop. **Curl consumes the stream of data coming from the server completely** and that is what we piped into the file `dump.html` with this command. And oddly enough, even though the logic of the page knew it was an invalid request that rquires login, **it still returned the full data** and a list of users of the system. A sample:

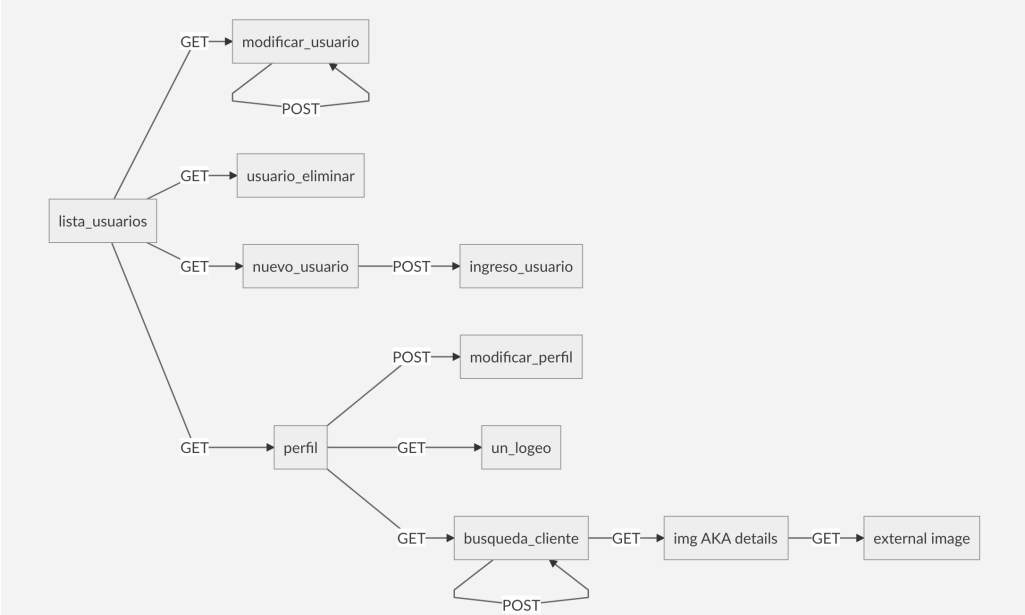
```
...
<tr>
    <td>OTERMIN</td>
    <td>Otermin, [redacted]</td>
    <td>2019-03-27 16:35:37</td>
    <td>No</td>
    <td>
        <div class="btn-group">
            <a href="modificar_usuario.php?i=1420" target="_blank" class="btn btn-primary" >
            <a onclick="return pregunta()" href="usuario_eliminar.php?i=1420" class="btn bg-
        </div>
    </td>
</tr>
<tr>
    <td>BONJOUR</td>
    <td>BONJOUR,[redacted]</td>
    <td>2019-04-01 17:18:50</td>
    <td>No</td><td>
        <div class="btn-group">
            <a href="modificar_usuario.php?i=1439" target="_blank" class="btn btn-primary" >
            <a onclick="return pregunta()" href="usuario_eliminar.php?i=1439" class="btn bg-
        </div>
    </td>
 </tr>
 ...
```

We find in total of *359 users*, and an interesting PHP warning when scanning the file:

```
Warning:  Cannot modify header information - headers already sent by (output started at /home/
```

A warning that should not have been ignored.

Now we are sure that parts of the site are not really protected by login at all. And we find the names of more pages within the system by scanning for other PHP pages referenced from the HTML. We have so far the following pages and actions:



And from these we start learning their vocabulary of field names, which are likely used to `POST` data or as query parameters. We have `i`, `usuario` / `usuarioid`, `email`, `nombre`, `contrasena` / `password` / `password2`, `permisos`, `coordina_urgente`, `empresa`, `funsion`, and `descripcion` fields so far. It also appears they have the concept of the user login `usuario` verses an internal ID `i` (range up to the value `1521`).

It appears that at least the creation of users is hardcoded to one client, the Banco Itau, so it is possible they are the only current client of this system, or the default on random `GET` of the pages. Other data suggerts other clients or previous clients given the digital imagry discussed later.

Light probing of `POSTS` absent of form data indicates that **no action other than `GET` redirects to the ineffectual login page**. Looking at the pages designed for `POST` we can still determine that you could modify data given no protection on any of those functions, and therefore this can surely can be used to silently **change the password of any user in the system without logging in**. At which point you have full user access to the system. Given no apparent system roles (inferred from the displayed data or forms), you could then create new users and play in the system endlessly consuming the millions of sensitive records available.

Changing a user password would be as simple as:

```
curl -vs -X POST \
  http://ues.com.uy/enmano/modificar_perfil.php \
  -H 'Accept: */*' \
  -H 'Cache-Control: no-cache' \
  -H 'Content-Type: application/x-www-form-urlencoded,text/plain' \
  -d 'usuario=SOMEUSERNAME&password=0123456&password2=0123456&nombre=The%20User&email=.'
```

Turning our attention to the `busqueda_cliente.php` page, this allows searching of all the client data. If you look at the form presented by `GET` of this page, and then `POST` the following fields: `reservation` (a date range in the form `10/07/2019 - 10/07/2019` - not Uruguay date format) , and `nro_cuenta` as the user login to retrieve data. Omitting the `nro_cuenta` parameter runs for a really long time and crashs with an error `500` possibly pulling data across all logins until it ran out of memory?

There appears to be a limit of `500` records returned, and paging was not explored, but limiting the query data to single days combining with other search properties can keep the results below this limit. The data is returned both in the HTML and as a data table, which appears to show part of this data at a time in paged form; with no links to return to the server for additional pages.

Each record is displayed in a table such as:

```
<tr>
    <td>ALUMI[redacted]</td>
    <td>COLORADO 1[redacted]</td>
    <td>Notificación Montevideo</td>
    <td>MONTEVIDEO-/-MONTEVIDEO</td><td>10/07/2019</td>
    <td>VISITADO NO ATIENDEN/SE DEJO COPIA</td>
    <td>
        <div class="btn-group">
            <a href="img.php?a=146700712" target="_blank" class="btn btn-warning">Detalles</a>
        </div>
    </td>
</tr>
```

which was rendered from the included data table:

```
<!-- /.box-body objeto = Array
(
    [0] => Array
        (
            [nro_acuse] => 146700712
            [nro_recibo] => 1160267
            [destinatario] => ALUMI[redacted]
            [direccion] => COLORADO 1[redacted]
            [localidad_nombre] => MONTEVIDEO-/-MONTEVIDEO
            [cliente_nombre] => Banco Itaú Uruguay S.A. Casa Central (WTC)
            [descripcion] => Notificación Montevideo
            [observaciones] => VISITADO NO ATIENDEN/SE DEJO COPIA
            [estado] => B
            [fecha_ingreso] => 2019-07-10 15:33:37.695
        )
```

Which provides a little bit more data than what is displayed.

Of note is the link for details of the form `img.php?a={nro_acuse}` . The CSS class of the button indicates status which is demo'd in the header part of the page.

```
<div class="box-header">
    <h3 class="box-title">Resultados de la búsqueda    -     Referencia :
    <button type="button" class="btn btn-default btn-flat">En gestión</button>  
    <button type="button" class="btn btn-warning btn-flat">Motivado</button>  
    <button type="button" class="btn btn-primary btn-flat">Entregado</button>  
    <button type="button" class="btn btn-info btn-flat">Con imagen</button>  
    <button type="button" class="btn bg-olive btn-flat">Reclamo</button>  
</div>
```

Therefore picking an item that has `btn-info` seems to tell us we might be awarded with some type of image. Taking a stab at `img.php?a=145986531` returns some of the same details with the addition of an image being displayed. In this case, the signature of the person receiving the letter. And in this case the source of that image is http://190.64.137.29/local/ftp/145986531.png.

This is not under a domain name (i.e. ues.com.uy), and therefore could not be sharing a cookie nor session with the main application, and there is no authorization information in the URL (which would be useless anyway). So going to the image link works without being logged in, since it is open to the public.

Given any item ID you can now predict the image on the HTTP server that appears to be fronting for an FTP server. Try the extension `png` for a signature or `jpg` for a document and you would be rewarded with the image.

But lookign more closely at that image URL. We go to the top level `http://190.64.137.29` of the server and get a nonsense response which looks like a health check. We go down one level to `http://190.64.137.29/local` and find a login page that UES has labelled with their name. It is not clear what it is protecting, because if you then go to `http://190.64.137.29/local/ftp` you receive... a list of **2+ million document images** due to **directory listing being enabled** for the server at this URL.

A web browser will truncate this list, but using `wget` will pull down the `412MB` or more directory listing showing **2,088,253** image files with dates ranging from 2018 until present.

```
ues > wget http://190.64.137.29/local/ftp
Warning: Failed to set locale category LC_NUMERIC to en_UY.
Warning: Failed to set locale category LC_TIME to en_UY.
Warning: Failed to set locale category LC_COLLATE to en_UY.
Warning: Failed to set locale category LC_MONETARY to en_UY.
Warning: Failed to set locale category LC_MESSAGES to en_UY.
--2019-07-14 22:51:38--  http://190.64.137.29/local/ftp
Connecting to 190.64.137.29:80... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: http://190.64.137.29/local/ftp/ [following]
--2019-07-14 22:51:38--  http://190.64.137.29/local/ftp/
Reusing existing connection to 190.64.137.29:80.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/html]
Saving to: 'ftp'

ftp [ <=> ] 412.61M  3.50MB/s    in 1m 59s

2019-07-14 22:54:16 (3.45 MB/s) - 'ftp' saved [432652177]

ues > cat ftp | grep href | wc -l
 2088253
```

Sampling these documents shows that the set contain:

- legal documents, including judgements, embargos, and more
- delivery receipts (the majority)
- invoices
- demand letters
- cheque orders
- and more

Many of these documents include peoples full names, addresses, federal ID number (in Uruguay your Cédula), their name written in their handwriting, and their signature in high resolution. And sometimes not the person intended to receive the docment, but rather the information of some bystander who happened to sign for the document and wrote their personal data on the receipt.

This level of data is not legal to obtain without explicit consent and store in a database. And none of these people have an expectation that receiving a document will include their private data into a computerized system in digitized form.

So what could a bad actor do with this set of data from the system and its matching digital documents? Let's brainstorm!

- Create fake petitions with real names, their address, their written name (aclaration), and their signature (firma)
- Determine who all has credit cards from specific banks
- Commit some sort of financial fraud, you have enough data!

- Create fake contracts
- Sell mailing lists of qualified people
- Expose people's wrong-doings from the legal documents you scraped up
- Intercept credit cards that are in transit to a known destination
- and more!

It appears there is data from many of the major banks and credit card companies of Uruguay in the system, along with the delivery receipts. There are legal judgements, tax related documents, invoices, earnings, cashed cheque information, and more.

Other Dangerous Issues:

This system also is suseptible to SQL injection on many of the `POST`able pages such as when modifying and deleting users. This causes risk of data loss and problems with the integrity of the system. It can also be used to inject a `sleep(N)` into the query `WHERE` clause causing the database server to denial-of-service itself. Not to mention injecting `' OR true` to cause deleting or updating all users.

All other best practices for protecting a website from attacks are also absent. Any friendly scan of the site returns a laundry list of possible security problems.

It should be noted that employees of clients of sites like this that are using sites with non-secure login pages should be refreshed on the dangers of doing so. Bank employees especially should be aware of this danger and notice the lack of security when logging into this page.

This database is unlawful in that:

(*ley 18.331 of 2008 and updates since*)

- it is not registered as a database containing personal data (*which in process includes a lot of questions that are indicators that you have designed a manageable and secure system*)
- they would need to include the digitized document as well as the textual data in that registration
- there is no consent from anyone for their data to be in this database, and this data exceeds the limits of what can be held without explicit consent. It also includes parties unrelated to the delivery (someone deliverying a legal document to another party which includes information of yet another party).
- it is holding data beyond the need and purpose of having the data (2018 data until now)
- more data is kept than neccessary to perform the intended action
- the system exposes the private personal data publically
- there is **no attempt** at security on the digital documents, and there is completely failed attempt at security on the data reading pages, and no security on the writing pages.
- there is no obvious manner to remove yourself from this database
- By design, not by accident, they are exposing personal data
- Negligence on the part of the software developers and their management is present here at the highest level.

By Uruguayan law this is clearly an illegal database. Also, this data likely violates GDPR laws of Europe since Eurpean citizens use this service and are part of the database. Uruguay also has a law cooperating with some European other laws on data protection (*ley 19.030 - Approval of Convention 108*).

Regardless, UES has an obligation to inform AGESIC of the exposed data, and the affected parties who's personal data is exposed. Which in this case, is the full "general public" without restriction.

Affected parties:

Given the open nature of the interface to this data, and the length of time every element has been unprotected, it must be assumed the data has leaked extensively. Anything exposed on the internet this long has been misused.

It would be possible to pull a list of directly affected parties from the sending company (many major financial groups in Uruguay) and the delivery receipient from the textual data. But it would be a mean feat to extract the data of all that were actually affected by the contents of the digitized images as well. This includes other parties mentioned in documents, other parties receipting letters, and more. There are **over 2 million digitized images**, many more textual elements of data, in a country with a population of 3.457 million. The textual data is accumulating at a rate of over 500 per day (*due to hard limit of search page, the exact number hasn't yet been determined, and if determined and update will be made to thise document*).

This also does not include all of the data and digitized documents from years past that are no longer present, but have been present for periods of a year or more within their time window of being active. The whole public must be informed because **anyone** could be a member of the affected class.

Actions:

UES was notified of this data leak on July 12, 2019. Attending were Sebastian Salveraglio (Generente General de UES) and Daniel Fernandez (IT Director, Innovation Manager). Others might have been present on the call as video was disabled on the part of UES.

When notified of the leak, the IT Director of UES asked "How can you be so sure there was a breach?" in which I responded "anything this public, and this available, for this long, has leaked." **If UES is assuming otherwise, they are acting with the wrong intent and focus.**

as of Jul 12, 2019

Summary of the leak including accessing pages without login and the thought that over 600,000 documents were likely exposed was known *at the time*. Full details of bypassing login (ignoring redirect) has not been disclosed nor have they asked about the mechanism. The existance of the open FTP via HTTP site has not been disclosed, as at the time the documents were found via the link returned in the application data and the FTP site had yet to be fully indexed and sampled.

Therefore they are aware of the problems in general, which pages are affected, and the type of data being leaked including digital documents of the categories shown in this document.

as of July 14, 2019

Researcher making this discovery and writing this document with intent to preserve evidence as to the scope and contents of the leak for reporting to government authorities, AGESIC, GDPR, etc. Plans are to present to AGESIC on the 15th.

as of July 15, 2019

AGESIC informed (*see notifications header*)

on July 15, 2019

UES has started to make visible changes to the system. As of evening of the 15th, the FTP server fronted by an HTTP server at `http://190.64.137.29/local/ftp` now returns HTTP code `403` forbidden. The image access is replaced by links of the pattern:

```
http://190.64.137.29/local/api_img.php?a=145986531.png&l=431b2b0200da5d8c1a72ee3466a54d9b33eaf

http://190.64.137.29/local/api_img.php?a=146684808.jpg&l=3e6b6ad15ecb2ee95485bb9be5f328acb12b8

http://190.64.137.29/local/api_img.php?a=146684731.jpg&l=de18da6cddd14e469c31487a73eb027929a65

http://190.64.137.29/local/api_img.php?a=146704466.jpg&l=c9fd78fb2a4d5b4534398723ee755154057fa

http://190.64.137.29/local/api_img.php?a=146688900.png&l=8673a7b8df48dc9ea08da0ea08aec8a0f10fe

http://190.64.137.29/local/api_img.php?a=ftp/146688812.jpg&l=3e5d368dea8779dc4713f01536c7db0f3
```

They made this change before fixing access to the pages that show the link format and therefore the new pattern is able to be seen. The new `l` parameter appears to be SHA1 (SHA 128) likely related to the image number.

on July 16, 2019

Starting notification process to Banco Itaú.

on July 17, 2019

Notification meeting with Banco Itaú, with [redacted].

as of July 19, 2019

as of July 21, 2019

[redacted]

as of July 21, 2019

I send further information about the vulnerabilities that UES did not seem to understand in the prior meeting because they had not fixed them prior to just taking down the site.

Example digitized documents: