Sri Lanka Institute of Information Technology

IT21299902 (ZAKEY M.S.M.A)

Bug bounty Report 10.

Domain: curl.com

Web security – IE2062

B.Sc. (Hons) in Information Technology Specialization in cyber security.

Declaration:

- I hereby certify that no part of this assignment has been copied from any other work or any other source. No part of this assignment has been written/produced for me by another person.

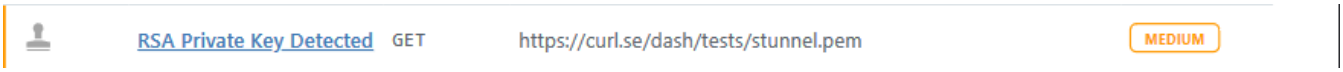
- I hold a copy of this assignment that I can produce if the original is lost or damaged.

1. Using components with known vulnerabilities.

- **RSA private key compromised.**

I have identified many vulnerabilities in under domain **www.curl.com/**. I figured out it has vulnerabilities that listed by OWSAP Top 10. And the overall website risk level is **critical**. I used net sparker to identify vulnerabilities in the following domain.

I hope this message finds you well. I am writing to report a vulnerability I discovered under the domain of **https://curl.se/dash/tests/stunnel.pem** during my participation in the bug bounty program. Please find below the details of the vulnerability for your review and further action.



Vulnerability title: RSA private key compromised.

Severity: medium

OWASP classification 2013: A3

CVSS 3.0 score: 7.7

CVSS 3.1 score: 7.7

CVSS string: CVSS:3.0/AV: N/AC: L/PR: L/UI: N/S: C/C: H/I: N/A: N

Effected components: Application backend.

Web server.

API endpoints.

Date of Discovery: 2023/05/16

Date of Report: 2023/05/18

2. Vulnerability Description.

Under this OWSAP category I have identified vulnerabilities sensitive data exposure that can occur due to bad implementation security under domain of **https://curl.se/dash/tests/stunnel.pem**

When attempting to connect onto a secure server, the client application uses a digital signature to demonstrate that you own the private key. The server then verifies that the signature is valid and that the public key is approved for your username. If everything is okay, access is allowed to you. If this key is compromised, then there is no point in using the secure mechanism. Anyone can access your secure server with this key.

3. Impact assessment.

When the private key is not password-protected, anyone who obtains it can access all your accounts.

1. Even if it is password-protected, an attacker with average computational power can test a vast variety of password combinations. Your passphrase can probably be cracked in a matter of seconds if it contains a dictionary word.

2. The attacker may be able to impersonate authorized users, obtain access to user accounts without authorization, or fake digital signatures if the leaked RSA key is used for user authentication or digital signatures. As a result, users may perform unlawful activities on their behalf, such as takeover of accounts, data manipulation, or fraudulent transactions.

3. In SSL/TLS certificates, RSA keys are frequently used to provide secure communication between clients and your web server. An attacker may be able to decrypt and intercept sensitive data sent via HTTPS connections if the RSA key is compromised. As a result, user conversations are no longer secure or private, leaving critical information vulnerable to interceptions or man-in-the-middle assaults.

4. An attacker may edit the application code, introduce malicious code, or disseminate modified copies of your application if the compromised RSA key is utilized for code signing or application integrity checks. This jeopardizes the reliability and integrity of your program and may cause the spread of malware or unauthorized changes.

Because of this vulnerability marked as medium It’s a must to address it with a good solution.

4. Steps to Reproduce.

1. Find the web application endpoint or feature that is causing the RSA secret key to appear in the response body.

2. To intercept and inspect HTTP requests and replies, set up a web proxy tool like Burp Suite or OWASP ZAP.

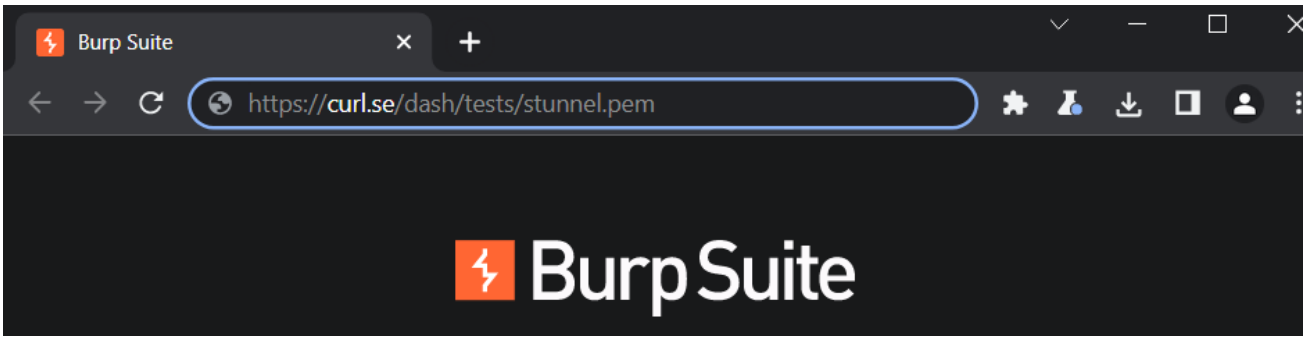
3. Set up your web browser to use the web proxy. The traffic between your browser and the web application can then be recorded and analyzed thanks to this.

4. Use the proxy tool to start recording HTTP traffic, then send a request that results in a response that contains the RSA secret key. This could entail submitting a form, using an API, or carrying out a certain task inside of your program.

5. Examine the response in the proxy tool after the request has been captured. Check the response body for the RSA secret key or any other private information that shouldn't be shared.

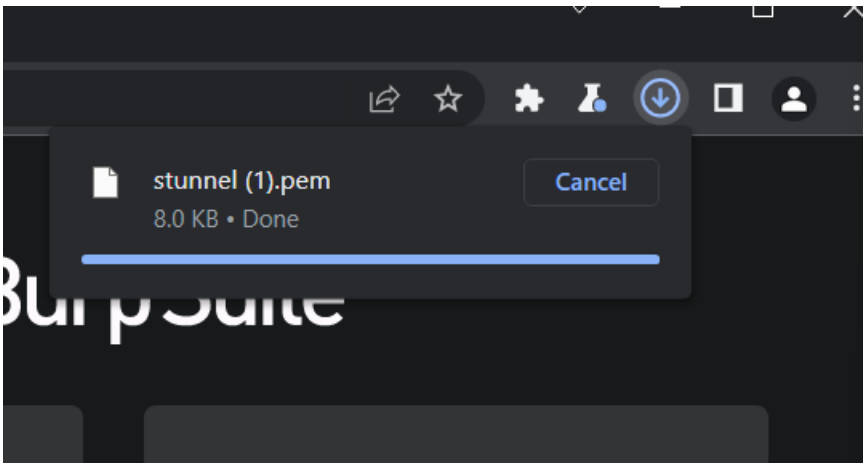
6. To ensure that the RSA secret key is consistently disclosed in the response body, test various scenarios or inputs.

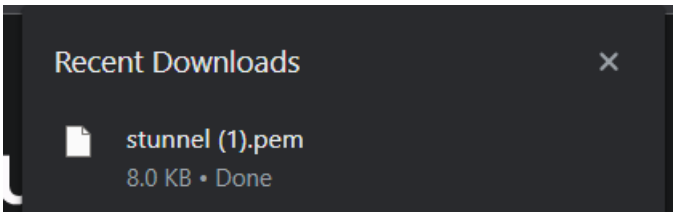
5. Proof of concept.



Pretty Raw Hex \n

```
GET /dash/tests/stunnel.pem HTTP/2
Host : curl.se
Sec-Ch-Ua : " Not A;Brand";v="99", "Chromium";v="104"
Sec-Ch-Ua-Mobile : ?0
Sec-Ch-Ua-Platform : "Windows"
Upgrade-Insecure-Requests : 1
User-Agent : Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/104.0.5112.102
Safari/537.36
Accept :
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,i
mage/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=
0.9
Sec-Fetch-Site : none
Sec-Fetch-Mode : navigate
Sec-Fetch-User : ?1
Sec-Fetch-Dest : document
Accept-Encoding : gzip, deflate
Accept-Language : en-US,en;q=0.9
```





```
[something]
# The key
# the certificate
# some dhparam
-----BEGIN PRIVATE KEY-----
MIIEvQIBADANBgkqhkiG9w0BAQEFAASCBKcwggSjAgEAAoIBAQCrCrAD0Hb+Xs4V
3mHV45FvfNa7yiaOeL4mNdGmWfHVPFU+CSzsoNSvDjxaorWweFGVYoCAcchOn1lZ
k0ASsqnOss0Xi58n8+PPI3gG0gYjX5sg7EJ3Zq2kXoK0TZRy6hNkcvzLgyzXoYv1
LkzTwYiyyJgZX++Y/GKAs2fMHyP8XzjNgm4tltk1k/4pomllwN9Fqz+sFxgAgEq3
ybq4Xym7xKwWl8xXNBDJNmVsPtiJRcilQoR8Xs0a6PE+VbMhD9A2E/LEL7lzQfqH
qtxE1mSW5FpQ+Uqf4KLnafStWs86IOWnCeLP6BmhAK6ouyICNFyzz7UkTHa/renx
uNOGun2TAgMBAAECggEAH0BsKb5Ax7h90jwYRzL141d9isFkaxq/r46c2FbN24bT
EmstxKycP8ILoAnjxbMuQOvHC/D+RvNRqY7Aocn4Qdakp50wvuWOpc3Ww/RC/9qb
pxfUCyn9Jy/HlPcp3RdM5MknzG2S13Fid7F2gyh0+CmztMs1JZBT1S0ylXbJJfbY
1pdlHcf9oEbYo36vGd9rtJHAFzsFfwua0idl76XYuOnR3bpOkHl1B5cJ8jpOliPv
VTmzn0cIgAmk7IByHHqGQ0u30PFiElI9kEbkKWoxAM1hq1pFU58jQhvp0ZkjVENL
bSFB2B4DbyosxPlbUgvJCN4B7nclqzYqBdrrk6/ZLQKBgQC1lDrPSGIGXLwvkZYS
xc0wtaCC7u6m7zV8rzh5HGcEoVvtmya/VyoZR8KGIpSor8COIkZqFtan6C77C3MH
wClbu2Kn3FkGb76D5U2Xwl38zepzjn8Z5qXc3bZfccrsDY1gXPicgsmcKUY9xV5/
T0RjESDKB+xxkJpCjia6klm2NQKBgQDxJNuqB6frDYKaj7mW/rvyHqkeT94J6eDY
```

6. Proof of existence of vulnerability

Request

```
GET /dash/tests/stunnel.pem HTTP/1.1
Host: curl.se
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: en-us,en;q=0.5
Cache-Control: no-cache
Referer: https://curl.se/dash/tests/
User-Agent: Mozilla/5.0 (Windows NT 10.0; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.
77 Safari/537.36
X-Scanner: Netsparker
```

Response

Response Time (ms) : 1640.3241 Total Bytes Received : 9016 Body Length : 8150 Is Compressed : No

```
HTTP/1.1 200 OK
X-Cache: MISS, MISS
X-Timer: S1684758746.943941,VS0,VE214
Age: 0
Cache-Control: max-age=1800
ETag: "1fd6-5f0c77b87bbec"
Strict-Transport-Security: max-age=31536000
Server: nginx/1.21.1
X-Content-Type-Options: nosniff
Connection: keep-alive
Expires: Mon, 22 May 2023 13:02:26 GMT
X-Frame-Options: SAMEORIGIN
Accept-Ranges: bytes
X-Cache-Hits: 0, 0
Content-Length: 8150
Via: 1.1 varnish, 1.1 varnish
alt-svc: h3=":443";ma=86400,h3-29=":443";ma=86400,h3-27=":443";ma=86400
Last-Modified: Tue, 27 Dec 2022 04:14:05 GMT
Content-Type: application/x-pem-file
X-Served-By: cache-bma1655-BMA, cache-qpg1246-QPG
Content-Security-Policy: default-src 'self' curl.haxx.se www.curl.se curl.se www.fastly-insights.com fa
stly-insights.com; style-src 'unsafe-inline' 'self' curl.haxx.se www.curl.se curl.se
Date: Mon, 22 May 20
```

```
_value     = Edel Curl Arctic Illudium Research Cloud
commonName                     = "Common Name"
commonName_value               = localhost
```

```
[something]
# The key
# the certificate
# some dhparam
-----BEGIN PRIVATE KEY-----
MIIEvQIBADANBgkqhkiG9w0BAQEFAASCBKcwggSjAgEAAoIBAQCrCrAD0Hb+Xs4V
3mHV45FvfNa7yiaOeL4mNdGmWfHVPFU+CSzsoNSvDjxaorWweFGVYoCAcchOn1lZ
k0ASsqnOss0Xi58n8+PPI3gG0gYjX5sg7EJ3Zq2kXoK0TZRy6hNkcvzLgyzXoYv1
LkzTwYiyyJgZX++Y/GKAs2fMHyP8XzjNgm4tltk1k/4pomllwN9Fqz+sFxgAgEq3
ybq4Xym7xKwWl8xXNBDJNmVsPtiJRcilQoR8Xs0a6PE+VbMhD9A2E/LEL7lzQfqH
qtxE1mSW5FpQ+Uqf4KLnafStWs86IOWnCeLP6BmhAK6ouyICNFyzz7UkTHa/renx
uNOGun2TAgMBAAECggEAH0BsKb5Ax7h90jwYRzL141d9isFkaxq/r46c2FbN24bT
EmstxKycP8ILoAnjxbMuQOvHC/D+RvNRqY7Aocn4Qdakp50wvuWOpc3Ww/RC/9qb
pxfUCyn9Jy/HlPcp3RdM5MknzG2S13Fid7F2gyh0+CmztMs1JZBT1S0ylXbJJfbY
1pdlHcf9oEbYo36vGd9rtJHAFzsFfwua0idl76XYuOnR3bpOkHl1B5cJ8jpOliPv
VTmzn0cIgAmk7IByHHqGQ0u30PFiElI9kEbkKWoxAM1hq1pFU58jQhvp0ZkjVENL
bSFB2B4DbyosxPlbUgvJCN4B7nclqzYqBdrrk6/ZLQKBgQC1lDrPSGIGXLwvkZYS
xc0wtaCC7u6m7zV8rzh5HGcEoVvtmya/VyoZR8KGIpSor8COIkZqFtan6C77C3MH
wClbu2Kn3FkGb76D5U2Xwl38zepzjn8Z5qXc3bZfccrsDY1gXPicgsmcKUY9xV5/
```

```
T0RjESDKB+xxkJpCjia6klm2NQKBgQDxJNuqB6frDYKaj7mW/rvyHqkeT94J6e...
Certificate:
Data:
Version: 3 (0x2)
Serial Number: 16717980999635 (0xf3475519fd3)
Signature Algorithm: sha256WithRSAEncryption
Issuer:
countryName
…
```

7. Proposed remedy.

1. The first and most important step is to delete the RSA private key from the web application's response. Check that no server response, including HTTP response bodies, headers, or any other network transmission, contains the private key.

2. Instead of including the private key in the response from the web application, securely store it on the server or in a separate key management system. To ensure that only authorized individuals can access and handle the private key, implement the necessary access controls.

3. Use secure communication channels: Make that the web application encrypts data while it is being transmitted between the server and the client via a secure communication channel, such as HTTPS/TLS. This provides defense against listening devices and man-in-the-middle attacks.

4. Follow standard practices for key management, such as routinely updating and rotating the private key, limiting access to only authorized users, and implementing safe backup and recovery methods. To safely manage and store the private key, think about utilizing a hardware security module (HSM).