PENETRATION TEST REPORT AND RESPONSE

This document comprises the penetration test report conducted by an independent security agency, NCC Group, and RealVNC®'s response to it.

Version 2023.1
December 2023

RealVNC® response to the report

Our customers' security and privacy are of paramount importance to RealVNC®. As such, our flagship RealVNC® Connect remote access software is built from the ground up with security and privacy in mind. We understand that our online infrastructure, website, and the software itself must adhere to the highest security standards and must remain ahead of the curve as active security threats emerge.

To maintain a level of assurance, in December 2023 we engaged NCC Group, an independent security agency, to conduct our annual penetration test – to test our infrastructure and produce an objective report. This document addresses any potential issues uncovered in their reports, which can be found appended to this document.

The scope for this test was the RealVNC® Connect Portal (including SSO, API access keys and purchase flows), CMS website, On-Demand Assist website and our public network infrastructure for RealVNC Connect.

We believe this to be an exceptionally positive report with no critical, high or medium rated findings discovered.

Although only Low and Informational, a number of the findings are either not publicly discoverable (NCC Group had network access exceptions in place) or have a long series of unrealistic pre-requisites, meaning they do not affect real-world usage. Additionally, we have a number of external mechanisms for prevention of abuse and active monitoring of our services via our 24/7 Security Operations Centre and Technical Operations team, which are not visible externally.

We are proud of the positive feedback received from NCC Group during the engagement and from the documented report. This penetration test is a baseline security review, something we conduct on an annual basis, but we believe companies should go above and beyond to prove their security to customer. This is why in addition to this yearly blackbox penetration test, we periodically engage with an additional external security specialist, Cure53, for full whitebox security audits of our services. To read more about that process, please see https://www.realvnc.com/en/blog/cure53-security-audit-reaffirms-realvnc-strong-security-stance.

We continuously monitor and assess both internal and external environmental changes, which may affect our security posture. To learn more about RealVNC Connect security, visit our dedicated security page at https://www.realvnc.com/en/connect/security/ and to learn about RealVNC's Information Security visit https://trust.realvnc.com.

Web Application and Services Security Assessment

Real VNC

Version 1.0 – December 20, 2023

Prepared By
Ricardo Martin Rodríguez
Alberto Méndez

Prepared For
Ben May

1 Executive Summary

This report presents the findings of the Web Application and Services Security Assessment conducted on behalf of Real VNC. The assessment was conducted between 11/12/2023 and 14/12/2023.

The system being assessed allowed users to manage remotely control devices through a web application for audit and centralize access.

Overview

It was apparent that security had been a consideration in the development and deployment of this application as the risk from various common security flaws had been effectively mitigated. Nevertheless, a number of issues were identified which expose Real VNC to risk. The most significant issues are discussed in the Assessment Summary below.

The following table breaks down the issues which were identified by component and severity of risk (issues which are reported for information only are not included in the totals):

Component	Critical	High	Medium	Low	Total
CMS Application Security Assessment	0	0	0	1	**1**
Real VNC API Security Assessment	0	0	0	1	**1**
Real VNC Portal Security Assessment	0	0	0	0	**0**
Total	**0**	**0**	**0**	**2**	**2**

Assessment Summary

All issues were all assessed to pose a low risk or are reported for information only. Nevertheless, it is recommended that these are reviewed and addressed so as to bring the web applications and services into line with security best practice. It is important to recognise that even low risk issues can be exploited in combination with other issues as part of a wider attack which seeks to compromise an environment or application. In addition, resolving lower risk issues can have the dual benefit of reducing the attractiveness of systems to opportunistic attackers as well as enhancing the overall security posture.

More detailed information on each of the issues which were identified is included in the Finding Details section of this report.

Strategic Recommendations

Although no significant risks were identified in this assessment, it is recommended that the issues outlined in this report are reviewed in line with a suitably robust defence in depth approach which continuously monitors the organisation's security posture.

2 Table of Contents

3 Document Control

Client Confidentiality

This document contains Client Confidential information and may not be copied without written permission.

Proprietary Information

The content of this document should be considered proprietary information and should not be disclosed outside of Real VNC.

NCC Group gives permission to copy this report for the purposes of disseminating information within your organisation or any regulatory agency.

Document Data

Data Classification	Client Confidential
Client Name	Real VNC
Project Reference	RVNC002
Proposal Reference	O-205206
Document Title	Web Application and Services Security Assessment
Author	Ricardo Martin Rodríguez

Document History

Version	Issue Date	Issued by	Change Description
0.1	2023-12-11	Ricardo Martin Rodríguez	Draft for NCC Group internal review only
0.2	2023-12-20	Philip Marsden	Revised QA
1.0	2023-12-20	Ricardo Martin Rodriguez	Released to client

Document Distribution List

Name	Role
Ben May	Project Sponsor, Real VNC

4 Technical Summary

NCC Group was contracted by Real VNC to conduct a security assessment of the systems within scope in order to identify security issues that could negatively affect Real VNC's business or reputation if they led to the compromise or abuse of systems.

Scope

The security assessment was carried out in the UAT and live environments and included the following sections. The URLs within the scope of each section are listed below:

- Real VNC API Security Assessment:
  - https://s-docs.realvnc.com/api-access/openapi.json (SHA256: 62ccfba70bac1f3b36a0f91a9b3c59cb9310c4b772e6b01d37b391028f807a12)
- CMS Application Security Assessment:
  - https://www.realvnc.com/ (CMS web application)
- Real VNC Portal Security Assessment
  - https://s-manage.realvnc.com/
  - https://manage.realvnc.com

Caveats

Checks that would have a high probability of causing disruption to the named hosts were excluded. Denial of service attempts were excluded for the same reason.

5 Table of Findings

For each finding, NCC Group uses a composite risk score that takes into account the severity of the risk, application's exposure and user population, technical difficulty of exploitation, and other factors.

Title	Status	ID	Risk
Outdated WordPress Plugin	Fixed	9RJ	Low
No Online Certificate Status Protocol (OCSP) Stapling	New	7HA	Low
Lack of Input Validation	New	GUP	Info
Default Pages Present	New	K6Q	Info
WordPress Username Enumeration	New	2Q9	Info
Third-Party Script Included Without Subresource Integrity Hash	New	LVG	Info
Wildcard SSL Certificate in Use	New	QVV	Info
No Certification Authority Authorisation (CAA) Record	New	Y2B	Info
Reflected Cross-Site Scripting (XSS)	New	6GA	Info

6 Risk Ratings

The table below gives a key to the ratings used throughout this report to provide a clear and concise risk scoring system.

It should be stressed that quantifying the overall business risk posed by any of the issues found in any test is outside our remit. This means that some risks may be reported as high from a technical perspective but may, as a result of other controls unknown to us, be considered acceptable.

Risk Rating	CVSS Score	Explanation
Critical	9.0 - 10	A vulnerability was discovered that has been rated as critical. This requires resolution as quickly as possible.
High	7.0 - 8.9	A vulnerability was discovered that has been rated as high. This requires resolution in the short term.
Medium	4.0 - 6.9	A vulnerability was discovered that has been rated as medium. This should be resolved as part of the ongoing security maintenance of the system.
Low	1.0 - 3.9	A vulnerability was discovered that has been rated as low. This should be addressed as part of routine maintenance tasks.
Info	0 - 0.9	A discovery was made that is reported for information. This should be addressed in order to meet leading practice.

Impact

Impact reflects the effects that successful exploitation has upon the target system or systems. It takes into account potential losses of confidentiality, integrity and availability, as well as potential reputational losses.

Rating	Description
High	Attackers can read or modify all data in a system, execute arbitrary code on the system, or escalate their privileges to superuser level.
Medium	Attackers can read or modify some unauthorized data on a system, deny access to that system, or gain significant internal technical information.
Low	Attackers can gain small amounts of unauthorized information or slightly degrade system performance. May have a negative public perception of security.

Exploitability

Exploitability reflects the ease with which attackers may exploit a finding. It takes into account the level of access required, availability of exploitation information, requirements relating to social engineering, race conditions, brute forcing, etc, and other impediments to exploitation.

Rating	Description
High	Attackers can unilaterally exploit the finding without special permissions or significant roadblocks.
Medium	Attackers would need to leverage a third party, gain non-public information, exploit a race condition, already have privileged access, or otherwise overcome moderate hurdles in order to exploit the finding.
Low	Exploitation requires implausible social engineering, a difficult race condition, guessing difficult-to-guess data, or is otherwise unlikely.

7 Finding Details

Low

Outdated WordPress Plugin

Overall Risk	Low	**Finding ID**	NCC-RVNC002-9RJ
Impact	Undetermined	**Component**	CMS Application Security Assessment
Exploitability	Undetermined	**Category**	Patching
		Status	Fixed

Description

One WordPress plugin from the CMS web application was outdated and affected by one publicly reported security vulnerabilities. This indicates there may be a gap within the security patching process.

As shown below, the version in use by the Elementor plugin was 3.18.0, which was affected by CVE-2023-48777. [1] This issue which was exposed by the outdated software could be leveraged by an authenticated attacker (with the Contributor+ role) to gain remote code execution through a file upload issue.

```
$ curl --silent https://www.realvnc.com/wp-content/plugins/elementor/assets/js/admin-
↳ feedback.js | grep v3.18.0

/*!elementor - v3.18.0 - 08-12-2023*/ [...SNIP...]
```

It is worth noting that as per discussions with Real VNC, the Elementor plugin was updated during the security assessment. Therefore, it has been marked as “Fixed”.

Recommendation

Investigate the software patching and update policy and ensure that updates are applied to all software installations, including third party applications, on a regular basis. Consideration should be given to enabling the auto-update functionality within the affected third party software, to ensure that updates are applied quickly and regularly.

Location

- https://www.realvnc.com/

1. Elementor < 3.18.2 - Contributor+ Arbitrary File Upload to RCE via Template Import: https://wpscan.com/vulnerability/a6b3b14c-f06b-4506-9b88-854f155ebca9

Low

No Online Certificate Status Protocol (OCSP) Stapling

Overall Risk	Low	**Finding ID**	NCC-RVNC002-7HA
Impact	Low	**Component**	Real VNC API Security Assessment
Exploitability	Low	**Category**	Cryptography
		Status	New

Description

The TLS certificates offered by the web services endpoint did not offer OCSP stapling, a technology that builds upon certificate revocation technology and is used to provide a more reliable and secure method for a client to determine the revocation status of any particular TLS certificate.

With traditional certificate revocation methods, it is up to the client to retrieve Certificate Revocation Lists (CRLs). As this is a resource intensive process this can result in a failure to retrieve the required information and also can negatively affect the user experience.

Standard OCSP performs similar checks to traditional certificate revocation list (CRL) methods, but is considered more reliable and does not require large lists of information to be retrieved. However, the client is still required to communicate with the CA directly, which can lead to issues similar to those found with traditional CRL methods.

As shown below, the output of TestSSL [2] indicated that the endpoint did not offer OCSP stapling.

```
OCSP URI                      http://ocsp.digicert.com
OCSP stapling                 not offered
```

OCSP stapling removes the responsibility from the client to verify with the CA directly. Instead the server regularly polls the CA and obtains time stamped signed data. This is then stapled onto the TLS response to the client so that the client can verify if the connection is legitimate. This reduces the resource overhead of the technique and so reduces much of the negative consequences associated with traditional CRLs and regular OCSP. [3] [4]

Recommendation

Although OCSP stapling has a number of benefits, as a technology is not widely adopted and browser support varies. It is recommended that the technology is reviewed to determine if it can be deployed to the existing platform.

Location

- s-connect-api.services.vnc.com:443/tcp

2. TestSSL: https://testssl.sh/
3. Everything You Need to Know About OCSP, OCSP Stapling and OCSP Must-Staple - https://www.thesslstore.com/blog/ocsp-ocsp-stapling-ocsp-must-staple/
4. RFC 6960 - Online Certificate Status Protocol – OCSP https://tools.ietf.org/html/rfc6960

Info

Lack of Input Validation

Overall Risk	Informational	**Finding ID**	NCC-RVNC002-GUP
Impact	Undetermined	**Component**	Real VNC API Security Assessment
Exploitability	Undetermined	**Category**	Configuration
		Status	New

Description

One user supplied input within the web services within scope did not appear to consistently enforce input validation. Consequently, it was possible to submit payloads relevant to a number of vulnerabilities that were then accepted by the application. Whilst these were not directly exploitable, any back-end systems, or other systems that consume data being processed could be targeted (it is worth noting that although this value was reflected in the https://manage.realvnc.com web application, it was not possible to escalate it to other vulnerabilities such as cross-site scripting).

The request and response below show a cross-site scripting (XSS) payload submitted within the `name` parameter, which was accepted. This could then be observed within the output of the response.

Request:

```
PATCH /1.0/entries/26d6d272b46f4cab9eebe8096ffb9d49 HTTP/1.1
Host: s-connect-api.services.vnc.com
User-Agent: curl/8.3.0
Accept: */*
Authorization: Bearer ey<redacted>yg
Connection: close
Content-Length: 44
Content-Type: application/json

{
  "name": "<script>alert(1)</script>"
}
```

Response:

```
HTTP/1.1 200
Vary: Origin
Vary: Access-Control-Request-Method
Vary: Access-Control-Request-Headers
ETag: "172"
Content-Type: application/json
Content-Length: 666
Date: Tue, 12 Dec 2023 11:56:21 GMT
Connection: close
Server: gateway/2.4.0-RELEASE

{
  "entryId" : "26d6d272b46f4cab9eebe8096ffb9d49",
  "serverId" : "UmdLYq-9GuwL-xRpUMr",
  "name" : "<script>alert1</script>",
```


```
  "serverLastSeenTime" : "2022-11-14T16:09:59.897213Z",
  "data" : {
  [...SNIP...]
}
```

It is worth noting that this issue also affected another API endpoint (refer to Supplemental Data - Lack of Input Validation for further details). The lack of validation in the `expiry` field of the HTTP Post request allowed creating malformed JSON Web Tokens (JWT).

Recommendation

Ensure that documentation relating to the use of this service explains the potential content of response data, and covers the importance of handling this safely. [5] [6]

Input validation involves the application rejecting any characters which are invalid for the field in question, preferably by whitelisting a limited set of characters (in a telephone field, for example, the whitelisted characters could be 0-9, parentheses, and hyphens). In addition to character and format validation, the input length and range of numerical parameters should also be validated wherever it is applicable. This strategy can also help in mitigating other flaws which stem from a failure to sanitise input, such as SQL or HTTP header injection attacks.

Output encoding requires the encoding (or escaping) of all special characters (such as those used in HTML and JavaScript) in potentially malicious data. Most web programming languages have libraries which will do this automatically. Note that the correct escaping of the output depends on the location in which the data is to be used within the response. If this is within the main body of the document, HTML entities must be escaped. If the input is to be used within a script inside a string, the quotes used for that string must be escaped. In general, it is important to ensure that it is not possible for the data to include whatever sequence is used to demark the end of that data and the beginning of something else.

Location

- “Update entry” API Endpoint: https://s-connect-api.services.vnc.com/1.0/entries/`<entryId>`
- “Create session” API Endpoint: https://s-connect-api.services.vnc.com/1.0/sessions

5. OWASP - Improper Data Validation: https://owasp.org/www-community/vulnerabilities/Improper_Data_Validation
6. OWASP - Input Validation Cheat Sheet: https://cheatsheetseries.owasp.org/cheatsheets/Input_Validation_Cheat_Sheet.html

Info

Default Pages Present

Overall Risk	Informational	**Finding ID**	NCC-RVNC002-K6Q
Impact	None	**Component**	CMS Application Security Assessment
Exploitability	Low	**Category**	Configuration
		Status	New

Description

Default web server content existed on the CMS (WordPress) application within scope. This took the form of files and directories which were created when the web server software was installed. This content can represent a security risk, as it may provide an attacker with information which will be useful in other attacks (such as details of the versions of web server software in use), and in some cases, may even contain vulnerabilities itself.

The following default files were found in the application within scope. An example is provided below.

- /license.txt
- /readme.html

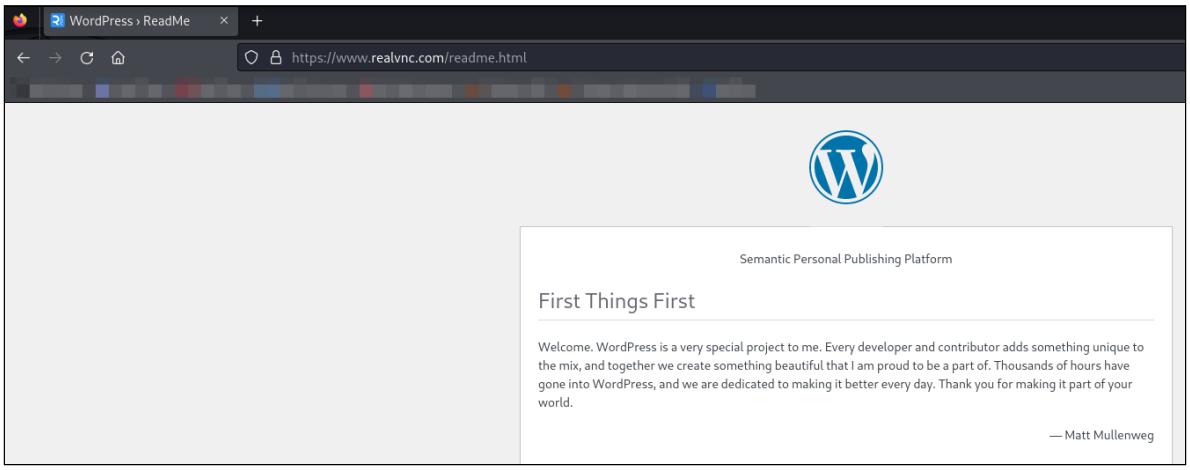


Figure 1: Accesible WordPress readme.html File

Note that even when no clear security risk is apparent, this finding could indicate that server-side content and settings are in a default or unhardened state.

In addition, it is worth noting that the application was protected by the Cloudflare WAF (Web Application Firewall), which was disabled for testing purposes. Therefore, an attacker would need to circumvent the protections enforced by the WAF in order to access these files.

As a result of all of these protections enforced in the public application, this issue was rated as informational.

Recommendation

All default pages should be removed if they are not required for normal operation of the site, in order to reduce the attack surface of the server and eliminate information leaks. If this content is required then access to these pages should be restricted to authorised users. [7]

7. CWE-200: Information Exposure: https://cwe.mitre.org/data/definitions/200.html

Ensure that appropriate hardening of server-side software and content has been performed.

Location

- https://www.realvnc.com/

Info

WordPress Username Enumeration

Overall Risk	Informational	**Finding ID**	NCC-RVNC002-2Q9
Impact	None	**Component**	CMS Application Security Assessment
Exploitability	Low	**Category**	Other
		Status	New

Description

It was possible to confirm whether or not a username or email address was already registered to a user account in the affected WordPress application. As a result, an attacker can more easily identify which usernames are valid for the application. This information could be useful in further attacks, such as phishing or account takeover via brute-force password guessing.

On WordPress [8] this can be done using the author permalinks technique or through the API. More specifically, the following usernames were obtained through the Yoast Seo author sitemap, [9] oEmbed API (Author URL) and author ID brute forcing. The following users were identified:

- jc
- david
- flickerleap
- hayley
- matthew
- asif
- ben
- bogdan
- nickc
- janndel-rosariorealvnc-com
- realvnc
- jan-intiarealvnc-com

In addition, as shown below, given that the application disclosed whether a username existed or not, it was possible to brute force the application in order to find valid users (note the length of the HTTP response when the user is valid).

8. WordPress User Enumeration: https://hackertarget.com/wordpress-user-enumeration/
9. Customize the Author Sitemap: https://yoast.com/help/how-to-exclude-author-pages-from-sitemap/

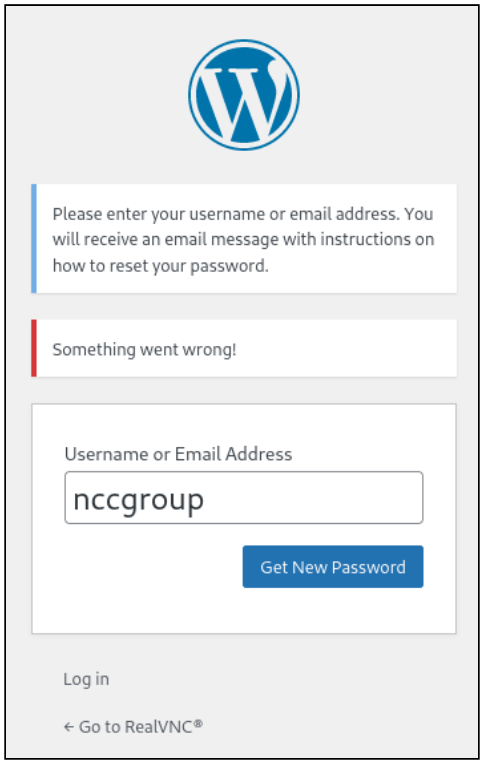


Figure 2: Application Disclosing Username Was Not Registered

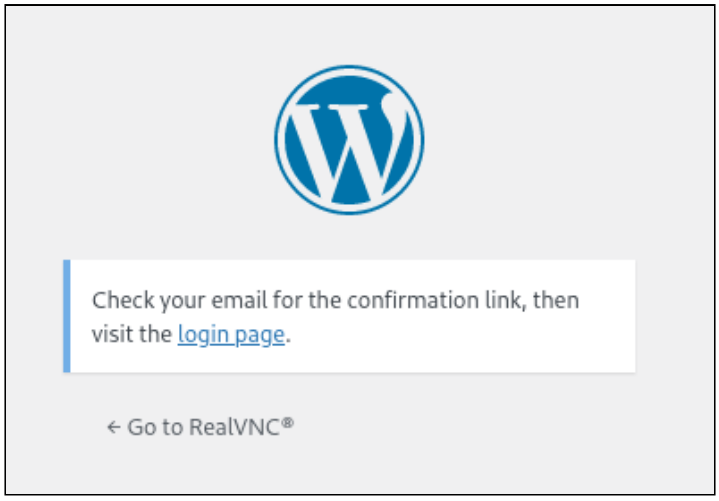
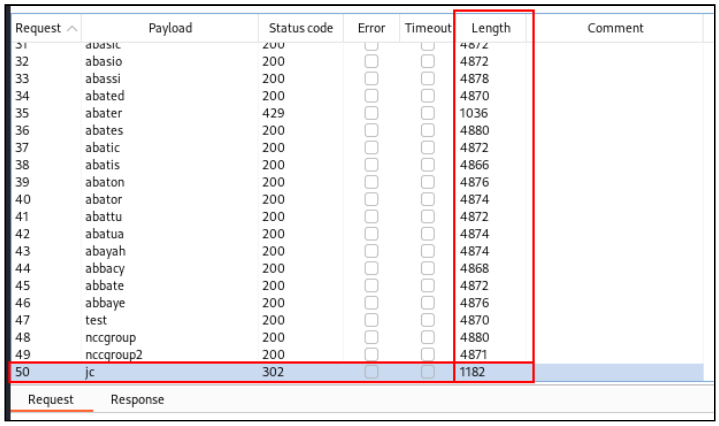


Figure 3: Application Disclosing Username Was Registered



Request	Payload	Status code	Error	Timeout	Length	Comment
31	abasic	200			4872	
32	abasio	200			4872	
33	abassi	200			4878	
34	abated	200			4870	
35	abater	429			1036	
36	abates	200			4880	
37	abatic	200			4872	
38	abatis	200			4866	
39	abaton	200			4876	
40	abator	200			4874	
41	abattu	200			4872	
42	abatua	200			4874	
43	abayah	200			4874	
44	abbacy	200			4868	
45	abbate	200			4872	
46	abbaye	200			4876	
47	test	200			4870	
48	nccgroup	200			4880	
49	nccgroup2	200			4871	
50	jc	302			1182	

Request Response

Figure 4: Brute Force Attack Against the Login Form

In addition, it is worth noting that the application was protected by the Cloudflare WAF (Web Application Firewall), which was disabled for testing purposes. Therefore, an attacker would need to circumvent the protections enforced by the WAF in order to leverage this issue.

As a result of all of these protections enforced in the public application, this issue was rated as informational.

Recommendation

It is recommended to prevent WordPress [10] username enumeration using different techniques like create an ".htaccess" file to block the permalinks used for this. This can be also mitigated using some plugins [11] [12] including the API access.

Location

- https://www.realvnc.com/

10. Prevent WordPress Username Enumeration: https://www.jinsonvarghese.com/prevent-wordpress-username-enumeration/
11. WP Hardening Plugin: https://wordpress.org/plugins/wp-security-hardening/
12. Stop User Enumeration Plugin: https://wordpress.org/plugins/stop-user-enumeration/

Info

Third-Party Script Included Without Subresource Integrity Hash

Overall Risk	Informational	**Finding ID**	NCC-RVNC002-LVG
Impact	Low	**Component**	Real VNC Portal Security Assessment
Exploitability	Low	**Category**	Other
		Status	New

Description

The VNC Portal used JavaScript code from multiple external sources. This creates the risk that a compromise of the third-party script host could result in a compromise of the application's users. Specifically, if an attacker compromises the third-party host, they could replace the script with a malicious script that completely controls user accounts. Third-party JavaScript has been documented as source of site compromise in the past. [13]

Including external JavaScript libraries implies not only trust that the host of the libraries will not modify them in a way that breaks functionality or introduces vulnerabilities, but also that the host is itself sufficiently secure. If the third party host comes under attack, the attacker could potentially use the targeted library as a vector to attack users of the application.

In order to mitigate this risk, Subresource Integrity (SRI) [14] was introduced as a browser feature in most major browsers. [15] This feature allows web applications to specify a hash of a script included with a `<script>` tag in order to verify the file has not been modified. Unfortunately, this feature has only received limited support from the vendors who most commonly provide hosted JavaScript. If the vendor does not support SRI, then the only choices may be to keep the functionality as-is, or to remove the script and associated functionality.

The following external scripts were referenced:

- https://cdn.cookielaw.org/consent/4d3612a3-53c3-402d-89a9-8189152577d9-test/OtAutoBlock.js
- https://cdn.cookielaw.org/scripttemplates/otSDKStub.js
- https://static.zuora.com/Resources/libs/hosted/1.3.1/zuora-min.js
- https://www.paypalobjects.com/api/checkout.js

Recommendation

Ideally, active content such as JavaScript, CSS, HTML, Java or Flash code should be hosted locally, rather than be included from third party hosts. If external hosting is preferred – usually for the performance gains delivered by content delivery networks (CDNs) – it is recommended that only reputable third parties are used and that, in the case of script and CSS files, the `Subresource Integrity` (SRI) attribute is added to force an integrity check. SRI

13. The JavaScript Supply Chain Paradox: SRI, CSP and Trust in Third Party Libraries: https://www.troyhunt.com/the-javascript-supply-chain-paradox-sri-csp-and-trust-in-third-party-libraries/
14. Mozilla - Subresource Integrity (SRI): https://developer.mozilla.org/en-US/docs/Web/Security/Subresource_Integrity
15. Can I use - Subresource Integrity: https://caniuse.com/#feat=subresource-integrity

specifies an encoded hash of the expected file, which conforming browsers will verify; for example:

```
<script src="//some.other.site.com/jquery/jquery.min.js" integrity="sha384-I6F5OKECLVtK/
↳BL+8iSLDEHowSAfUo76ZL9+kGAgTRdiByINKJaqTPH/QVNS1VDb" crossorigin="anonymous"></script>
```

In this case, should the hash of the file received by the browser from the third party not match the value specified by the first party, the script will not be loaded. For more information on SRI implementation and browser support, please see, [16] [17] but note that SRI:

- Requires the `crossorigin` attribute
- Cannot check the integrity of scripts that are loaded dynamically
- Provides no effective protection if the first party page is delivered over HTTP
- Will prove problematic with resources that change without notice (and therefore it may be preferable to reference a specific version rather than the 'latest' version)

Location

- https://manage.realvnc.com/

16. Subresource Integrity - W3C recommendation: https://www.w3.org/TR/SRI/
17. Create your SRI hash: https://report-uri.com/home/sri_hash

Info

Wildcard SSL Certificate in Use

Overall Risk	Informational	**Finding ID**	NCC-RVNC002-QVV
Impact	Low	**Component**	Real VNC API Security Assessment
Exploitability	Low	**Category**	Cryptography
		Status	New

Description

The SSL services offered by the web services within scope used a wildcard SSL certificate. Such certificates offer a cost-effective means of extending SSL/TLS coverage across multiple servers and applications. However, although wildcard certificates are cryptographically no weaker than dedicated certificates, the effective security level is reduced to that of the weakest application or component.

The following wildcard certificate was found:

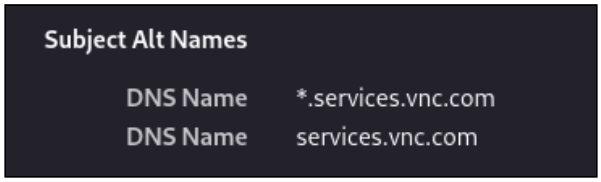


Figure 5: Wildcard Certificate in Use

Should an attacker be able to compromise one server or application that uses a wildcard certificate and recover the certificate's private key, it would then be possible to mount a man-in-the-middle attack against any SSL/TLS enabled service in any of the subdomains covered by the wildcard certificate, even if they have a different certificate installed.

Note that Extended Validation Certificates cannot be issued for wildcard certificates.

Recommendation

If possible, make use of a separate certificate for each application or service.

If it is not cost-effective to deploy a separate certificate for each application or service, consider using Subject Alternative Names to allow a certificate to cover multiple hostnames. This would require a new certificate to be issued.

Where certificates are reused, consider the security domains in which they operate. For example, a certificate used for a publicly accessible web forum application of low business importance should not also be used for a business critical web application that processes payments or otherwise handles sensitive information. A similar separation should be considered between test and production environments. [18] [19]

Ensure that incident response processes account for the use of wildcard certificates in the event of a server or application compromise.

Location

- s-connect-api.services.vnc.com:443/tcp

18. The Risks in Wildcard Certificates: https://www.sslshopper.com/article-the-risks-in-wildcard-certificates.html
19. OWASP Transport Layer Protection Cheat Sheet: https://www.owasp.org/index.php/Transport_Layer_Protection_Cheat_Sheet

Info

No Certification Authority Authorisation (CAA) Record

Overall Risk	Informational	**Finding ID**	NCC-RVNC002-Y2B
Impact	Informational	**Component**	Real VNC API Security Assessment
Exploitability	Informational	**Category**	Configuration
		Status	New

Description

The web services endpoint did not have an associated DNS Certification Authority Authorisation (CAA) Resource Record. CAA is a type of DNS record standardised in 2013 aimed at reducing the risk of certificate mis-issue. A CAA record allows application owners to specify which Certificate Authorities (CAs) are allowed to issue certificates containing their domain name.

Ordinarily a CA is permitted to issue certificates for any public DNS domain name if the applicant can validate control of that domain name. Utilising a CAA reduces the risk that a bug in any CAs validation process would allow them to issue a certificate to another third party.

As shown below, the output of TestSSL [20] indicated that the endpoint did not have an associated DNS CAA record.

```
DNS CAA RR (experimental)    not offered
```

Recommendation

A CAA Resource Record should be created that contains only the expected and authorised CAs for the domain. An example CAA record may appear as below. [21] [22]

```
example.org. CAA 128 issue "letsencrypt.org"
```

Location

- s-connect-api.services.vnc.com:443/tcp

20. TestSSL: https://testssl.sh/
21. RFC 6844 - DNS Certification Authority Authorization (CAA) Resource Record: https://tools.ietf.org/html/rfc6844
22. AWS Configure a CAA Record: https://docs.aws.amazon.com/acm/latest/userguide/setup-caa.html

Info

Reflected Cross-Site Scripting (XSS)

Overall Risk	Informational	**Finding ID**	NCC-RVNC002-6GA
Impact	Low	**Component**	Real VNC Portal Security Assessment
Exploitability	Low	**Category**	Data Validation
		Status	New

Description

The Real VNC Portal was vulnerable to reflected, or non-persistent, cross-site scripting (XSS) attacks. This type of vulnerability occurs when data provided by a web client is used immediately by server-side scripts to generate a page of results for the user. If unvalidated user-supplied data is included in the resulting page without full and proper HTML escaping, client-side executable code may be injected into the dynamic page.

In this case of a POST request, a victim user would have to first be persuaded to visit an otherwise unrelated site which then launched the attack using a form, and in addition, the attack should find a way to spoof the `Referer` header to inject the JavaScript payload.

Reflected cross-site scripting vulnerabilities are typically used to launch site impersonation or phishing attacks, in which unsuspecting users are lured to malicious sites via links that appear legitimate. The attacker is then free to present the user with what appears to be genuine content, in an attempt, for example, to capture authentication credentials. Another common method of exploitation is to capture the session token of the victim user, allowing their session to be hijacked by the attacker.

In this specific case, the Real VNC Portal allowed authenticated users to specify promotional codes in the purchase workflow. As shown in the HTTP request below, the value of the `Referer` header was reflected in the `href` attribute of the `<a>` HTML tag.

The affected instance of XSS was not accessible until users logged in, meaning that the victim would have to be persuaded to click a malicious link while authenticated to the web portal. An attacker could exploit this issue to steal a promotion code among other things.

Request:

```
POST /purchase/check_promo_code HTTP/1.1
Host: s-manage.realvnc.com
Cookie: <redacted>
Content-Length: 72
Sec-Ch-Ua: "Not_A Brand";v="8", "Chromium";v="120"
Accept: */*
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
X-Requested-With: XMLHttpRequest
Sec-Ch-Ua-Mobile: ?0
User-Agent: Mozilla/5.0
Origin: https://s-manage.realvnc.com
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: vbscript:msgbox(96861088)mfeck7
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
```

```
Priority: u=1, i
Connection: close

promo_code=PENTEST23&csrf_token=25f46726a3eff198c2dec8c634431f4e4a12cdfd
```

Response:

```
<div class="rich-text">
  <p>
    Sorry, this form has expired, please <a href="vbscript:msgbox(96861088)mfeck7">go back</a>
    ↳ and try again.
  </p>
</div>
```

It is worth noting that this issue is only exploitable in Internet Explorer 11 and below this version, and it would be also required for an attacker to circumvent the WAF protection and the CSRF (Cross-site request forgery) protection. Due to the numerous challenges and specific conditions required to exploit this XSS vulnerability, its severity has been downgraded.

> The last version of Internet Explorer to support VBScript was Internet Explorer 11. [23] In 2019, Microsoft disabled VBScript by default in Internet Explorer 11 on Windows 7, Windows 8, and Windows 8.1.

Recommendation

Reliable avoidance of cross-site scripting vulnerabilities should consist of two stages - input validation and output encoding. [24] [25] [26]

Input validation involves the application rejecting any characters which are invalid for the field in question, preferably by adding an allowlist for a limited set of characters (in a telephone number field, for example, the allowlisted characters could be 0-9, parentheses, and hyphens). This strategy can also help in mitigating other flaws which stem from a failure to sanitise input, such as SQL or HTTP header injection attacks.

Output encoding requires the encoding of all special characters (such as those used in HTML and JavaScript) in potentially malicious data. This is generally done directly before display by web applications (or client-side script), and many programming languages have built-in functions or libraries which provide this encoding (also called quoting or escaping in this context). Note that the correct encoding of the output depends on the location that the data is to be used within the response. In the case of it being within the main body of the document, HTML entities must be encoded. If the input is to be used within a script inside of a string, the quotes used for that string must be escaped. In general, it is important to ensure that it is not possible for the data to include whatever sequence is used to demark the end of that data and the beginning of something else.

23. Disabling VBScript execution in Internet Explorer 11: https://blogs.windows.com/msedgedev/2017/04/12/disabling-vbscript-execution-in-internet-explorer-11/
24. OWASP XSS References: https://owasp.org/www-community/attacks/xss/, https://cheatsheetseries.owasp.org/cheatsheets/Cross_Site_Scripting_Prevention_Cheat_Sheet.html
25. OWASP Top 10 2017 – Cross-Site Scripting: https://owasp.org/www-project-top-ten/2017/A7_2017-Cross-Site_Scripting_(XSS)
26. CWE-079: Improper Neutralisation of Input during Web Page Generation ('Cross-site Scripting'): https://cwe.mitre.org/data/definitions/79.html

The application should be reviewed and, if necessary, modified, to handle malicious data properly. The specific instances identified in this finding should be addressed, and the application code base should also be examined for any similar issues which may exist.

Location

- https://s-manage.realvnc.com/purchase/check_promo_code

8 Supplemental Data - Lack of Input Validation

The “Create session” API endpoint from the web services within scope allowed the user to create JSON Web Tokens (JWT) for authenticating the HTTP requests. To create the JWT, two mandatory parameters must be sent in the HTTP Post request (`accessKey` and `accessKeyId`). The `expiry` parameter, which specified the length of time the session token should be valid for was optional and was configured to only allow one hour at most.

However, the `expiry` parameter was not validated for certain conditions, and it was possible to introduce negative values in an ISO 8601 format, creating a malformed JWT with a negative expiration time.

As shown below, it was possible to introduce -999999999 hours in the `expiry` parameter, creating a malformed JWT which expired in -114155 years.

Request:

```
POST /1.0/sessions HTTP/1.1
Host: s-connect-api.services.vnc.com
Content-Length: 107
Content-Type: application/json

{
  "accessKey":"Tt<redacted>zC",
  "accessKeyId":"Sy<redacted>ij",
  "expiry":"PT-999999999H"
}
```

Response:

```
HTTP/1.1 201
Vary: Origin
Vary: Access-Control-Request-Method
Vary: Access-Control-Request-Headers
Content-Type: application/json
Content-Length: 1052
Date: Mon, 11 Dec 2023 11:47:31 GMT
Server: gateway/2.3.0-RELEASE

{
  "token" : "ey<redacted>lA",
  "expiresIn" : -3599999996400
}
```

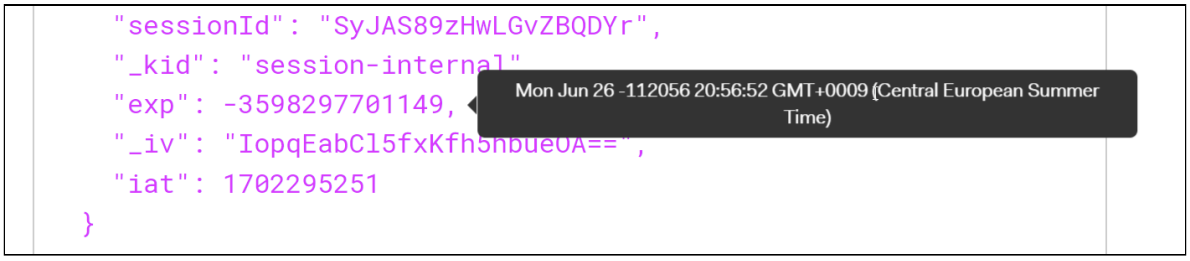


Figure 6: Resulted JWT Expiry Time

RealVNC®'s remote access and management software is used by hundreds of millions of people worldwide in every sector of industry, government and education. Our software helps organizations cut costs and improve the quality of supporting remote computers and applications. RealVNC® is the original developer of VNC remote access software and supports an unrivalled mix of desktop and mobile platforms. Using our software SDKs, third-party technology companies also embed remote access technology direct into their products through OEM agreements.

www.realvnc.com