Penetration test report and response

This document describes RealVNC's response to a penetration test report conducted by an independent security agency, NCC Group

December 2022

Version 2022.1

RealVNC response to the report

Our customers' security and privacy are of paramount importance to RealVNC. As such, our flagship VNC 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 November 2022 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 VNC Connect Portal (including SSO, API access keys and purchase flows), CMS website, Instant Support website and our public network infrastructure for VNC Connect.

We believe this to be an exceptionally positive report – with no critical, high or medium rated findings discovered. The two low findings, in the CMS website, have no connection to VNC Connect customer data; this is a marketing and sales site serving static information only. From the informational findings which were discovered, we have either improved them since discovery and/or we have a number of external means for prevention of abuse and active monitoring of the VNC Connect service, which are not visible outside our security and operations teams.

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 review, we engage with an additional external company, Cure53 for a full whitebox audit of our VNC Connect service. 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 VNC Connect security and compliance, visit our dedicated security page. If you have any further questions, please do not hesitate to contact us at enquiries@realvnc.com or via https://realvnc.com/contact-us

RealVNC

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

RealVNC- Web Application and Web Service Security Assessment

Real VNC Limited

Version 1.0 – December 6, 2022

Prepared By
Tatjana Sidorenko

Prepared For
Ben May

1 Executive Summary

This report presents the findings of the web application penetration test, WordPress and web service security assessments conducted on behalf of Real VNC Limited. The assessment was conducted between 14/11/2022 and 23/11/2022.

The system being assessed allowed users to use the remote access services, set up their teams with devices that needed support, download the client, view support logs and find out more information about the products. The API that allowed quick retrieval of the above data is also in scope.

Overview

The assessment established that the security posture of the applications and API in scope was mature and robust, with a relatively small number of issues identified. The majority of these security observations are reported for informational purposes, and hence excluded from the table below, whilst none of the issues assessed posed more than a low risk. The low risk issues were identified in the WordPress site. Despite the risk ratings, it is recommended that these issues are reviewed and addressed in line with a robust defence in depth approach to security.

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
Web Application Assessment	0	0	0	0	**0**
Web Service Assessment	0	0	0	0	**0**
WordPress Site Assessment	0	0	0	2	**2**
Total	**0**	**0**	**0**	**2**	**2**

Assessment Summary

Web Application Assessment

The discovered issues were all reported for information only. These included issues such as outdated JavaScript dependencies, weak password complexity rules, and inclusion of third-party libraries that were hosted externally.

WordPress Site Assessment

The most significant issue identified in the WordPress assessment was “Outdated WordPress plugins”. This gives an attacker a potential to exploit a vulnerability in the outdated plugin to potentially gain access to sensitive information or enable lateral movement. It is important to note that this issue is posing a low risk due to the lack of a proof-of-concept exploitation.

The remaining issues were all assessed to pose a low risk or are reported for information only.

Web Service Assessment

The only issue discovered during the Web Service Assessment phase was the CORS configuration. This allowed the specification of a remote host to be permitted to authenticate. With the current system configuration this issue cannot be exploited, since a different method of authentication is in place, hence it is reported for information only.

Despite the low severity of reported issues across the three phases, it is recommended that these are reviewed and addressed to bring the web applications and services within scope 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 Limited.

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 Limited
Project Reference	RVNC001
Proposal Reference	O-180242
Document Title	RealVNC- Web Application and Web Service Security Assessment
Author	Tatjana Sidorenko

Document History

Version	Issue Date	Issued by	Change Description
0.1	2022-11-11	Tatjana Sidorenko	Draft for NCC Group internal review only
0.2	2022-11-29	Michelle Simpson	Internal QA
1.0	2022-11-29	Tatjana Sidorenko	Released to client

Document Distribution List

Name	Role
Ben May	Head of Cyber Security, Real VNC Limited
Tatjana Sidorenko	Security Consultant, NCC Group
Jonny Cope	Mid Market Account Manager, NCC Group

4 Technical Summary

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

Scope

The security assessment was carried out in the production and staging environments and included the following sections:

- WordPress Web Application Assessment:
  - https://www.realvnc.com/
  - Assessed from an unauthenticated perspective
- Web application assessment of the Real VNC application:
  - https://manage.realvnc.com/
  - https://www.realvnc.help/
  - Payment assessed in staging (pre-production) only: https://s-manage.realvnc.com/
- Web service assessment of the Real VNC API:
  - https://s-connect-api.services.vnc.com/
  - 10 endpoints in scope
  - Documentation: https://docs.realvnc.com/api-access.html

Caveats

The payment functionality has been tested on the staging (pre-prod) environment due to the production environment not supporting test payments.

The `www.realvnc.com` only has four worker threads, hence some scans have been purposefully throttled to respect the limitation.

Rate limiting has been in place throughout the course of the assessment.

There have been updates pushed to the system, namely Python3 upgrade deployment and continuous WordPress updates. Because of this, some tested components may have additional unreported issues. Likewise, some of the reported issues may no longer be relevant.

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.

Post Assessment Cleanup

Any test accounts which were created for the purpose of this assessment should be disabled or removed, as appropriate, together with any associated content.

Revert any WAF/IDS/IPS/firewall changes which were made for the purposes of the assessment.

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.

Web Application Assessment

Title	Status	ID	Risk
Weak Password Complexity Requirements	New	YYD	Info
Third-Party Script Included Without Subresource Integrity Hash	New	3RV	Info
Wildcard SSL Certificate in Use	New	EME	Info
Outdated JavaScript Dependencies	New	4QA	Info
Username Enumeration via Forgotten Password Function	New	TWP	Info
Session Handling Configuration	New	DLP	Info

Web Service Assessment

Title	Status	ID	Risk
CORS Configuration	New	W4L	Info

WordPress Site Assessment

Title	Status	ID	Risk
Outdated WordPress Plugins	New	EW3	Low
Missing/Misconfigured Security-Related HTTP Headers	New	XTM	Low
Automated Abuse Protections	New	ECA	Info
Wordpress has XML-RPC Enabled	New	LNH	Info
Wildcard SSL Certificate in Use	New	TGW	Info
Outdated Version of JavaScript Dependencies	New	2VH	Info
Third Party JavaScript Library Inclusion	New	M6N	Info
Technical Information Disclosure	New	KFH	Info
WordPress Username Enumeration	New	XCK	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 – Web Application Assessment

Info

Weak Password Complexity Requirements

Overall Risk	Informational	**Finding ID**	NCC-RVNC001-YYD
Impact	Low	**Component**	Web Application Assessment
Exploitability	Low	**Category**	Authentication
		Status	New

Description

The password policy enforced by the VNC Portal application was not sufficiently robust [1]. Weak passwords can be easier to guess or to determine through a brute-force attack and could therefore lead to the compromise of user accounts, especially in the event of a password database breach.

The existing password policy was found to have the following weaknesses:

- Insufficient minimum length requirement (minimum of 8 characters)
- No password format specifications (such as requiring to include upper case or lower case characters, digits and symbols)
- Common dictionary words were allowed

As a result, it is possible for users to set their passwords to simple values such as `11111111`. If a user does use a weak password, an attacker could guess their password and gain access to their account. Alternatively, in the event of a password database breach, an attacker is more likely to recover a weak password from a brute-force attack.

Although the change password functionality did not allow the user to re-use the same password, this was not the case for the forgotten password functionality.

Recommendation

Ensure that a suitably strong password policy is in place, commensurate with any defined policies for the application, system, or organisation.

Passwords should be at least ten characters long, and should be forced to include at least one upper case and one lower case letter, at least one special character and at least one digit. However, consideration could be given to relaxing password complexity in favour of a higher minimum length, providing that suitable guidance is given. This is because an examination of any large scale password dump will show that the majority of users choose a password which is in line with the bare minimum required by a policy but is nevertheless weak. Therefore, any technical controls in this area should also be supported by efforts to educate users, both on the reasons for the policy and with practical tips for the creation of secure passwords.

When creating or changing user passwords, the application should perform checks for password complexity and reuse of previous credentials. Complexity checks can ideally be performed client side to provide immediate user feedback, using a tool like zxcvbn[2].

Do not require users to regularly update passwords, as this results in weaker passwords overall[3].

1. CWE-521: Weak Password Requirements: https://cwe.mitre.org/data/definitions/521.html
2. zxcvbn - GitHub: https://github.com/dropbox/zxcvbn.

Other defences to consider include:

- Detecting and responding to automated password attacks
- Blacklisting variations on common passwords, such as usernames, the application or the organisation
- Monitoring for unusual activity
- Making users aware of the last login event and encouraging them to report anything suspicious

Location

- https://manage.realvnc.com

3. Sophos - NIST's new password rules. What you need to know: https://nakedsecurity.sophos.com/2016/08/18/nists-new-password-rules-what-you-need-to-know/

Info

Third-Party Script Included Without Subresource Integrity Hash

Overall Risk	Informational	**Finding ID**	NCC-RVNC001-3RV
Impact	Low	**Component**	Web Application Assessment
Exploitability	Low	**Category**	Other
		Status	New

Description

The VNC Portal and help page 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[4].

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)[5] was introduced as a browser feature in most major browsers[6]. 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:

manage.realvnc.com

- https://cdn-3.convertexperiments.com/js/10021806-10025517.js
- https://js.hscta.net/cta/current.js
- https://static.zdassets.com/ekr/snippet.js?key=9bde55fa-1986-4904-b25e-51e2b0ae1a40
- https://static.zuora.com/Resources/libs/hosted/1.3.1/zuora-min.js
- https://www.paypalobjects.com/api/checkout.js

www.realvnc.help

- https://code.jquery.com/jquery-3.6.0.min.js

4. 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/
5. Mozilla - Subresource Integrity (SRI): https://developer.mozilla.org/en-US/docs/Web/Security/Subresource_Integrity
6. Can I use - Subresource Integrity: https://caniuse.com/#feat=subresource-integrity

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 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[7][8], 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
- https://realvnc.help

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

Info

Wildcard SSL Certificate in Use

Overall Risk	Informational	**Finding ID**	NCC-RVNC001-EME
Impact	Low	**Component**	Web Application Assessment
Exploitability	Low	**Category**	Cryptography
		Status	New

Description

The VNC Portal application 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. It was therefore notable that the certificate had the potential to be valid for both test and production environments.

The following wildcard certificates were found:

```
SSL Certificate:
Signature Algorithm: sha256WithRSAEncryption
RSA Key Strength:    2048

Subject:  *.realvnc.com
Altnames: DNS:*.realvnc.com, DNS:realvnc.com
Issuer:   DigiCert TLS RSA SHA256 2020 CA1
```

```
SSL Certificate:
Signature Algorithm: sha256WithRSAEncryption
RSA Key Strength:    2048

Subject:  *.realvnc.help
Altnames: DNS:*.realvnc.help, DNS:realvnc.help
Issuer:   DigiCert TLS RSA SHA256 2020 CA1
```

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.[91011]

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

Reproduction Steps

Visit an affected URL with a modern web browser. Inspect the certificate supplied by the site; Specifically, the `Subject` and `SAN` or `Subject Alternative Name` fields. Any entries in these fields containing wildcard characters (asterisks) indicate the use of wildcard certificates.

Location

- https://manage.realvnc.com
- https://s-manage.realvnc.com
- https://www.realvnc.help

9. The Risks in Wildcard Certificates: https://www.sslshopper.com/article-the-risks-in-wildcard-certificates.html
10. OWASP Transport Layer Protection Cheat Sheet: https://www.owasp.org/index.php/Transport_Layer_Protection_Cheat_Sheet
11. NCC Group Whitepaper on the Configuration of SSL/TLS Services: https://www.nccgroup.trust/uk/our-research/how-organisations-can-properly-configure-ssl-services-to-ensure-the-integrity-and-confidentiality-of-data-in-transit/

Info

Outdated JavaScript Dependencies

Overall Risk	Informational	**Finding ID**	NCC-RVNC001-4QA
Impact	None	**Component**	Web Application Assessment
Exploitability	None	**Category**	Patching
		Status	New

Description

The versions of some JavaScript libraries used by the VNC Portal application and help page were outdated, with one library containing a vulnerability that might be exploited under some scenarios. In the case of client-side JavaScript libraries, the disclosed vulnerabilities typically have low impact or in many cases are not exploitable, leading to minimal overall risk.

The following libraries were identified as being older than the current stable release version:

manage.realvnc.com

Library	Versions Used	Known Vulnerabilities Present?	Current Stable Version
jQuery	3.6.0	None at the time of writing.	3.6.1
jQuery UI	1.13.1	XSS	1.13.2

www.realvnc.help

Library	Versions Used	Known Vulnerabilities Present?	Current Stable Version
jQuery	3.6.0	None at the time of writing.	3.6.1

Observed URI locations are listed in the Affects table below.

Note that due to time constraints and library complexity, it was not possible to determine whether known vulnerabilities actually affected features of the library used by the website or not.

Please refer to the footnotes for a list of jQuery versions with known weaknesses. [12][13]

Recommendation

Consider adding a dependency-checking tool into the development and release workflow to automatically detect and report outdated dependencies. There are many free and paid tools which can assist with automatic checking, such as OWASP Dependency Checker[14], and `retire.js`[15][16]. However, be aware that automatically upgrading dependencies without auditing each dependency individually increases the chance that a malicious dependency is included in the application - a scenario which has become significantly more prevalent in recent years.

Location

- manage.realvnc.com:
  - https://static.realvnc.com/static/297113309/portal/js/corporate_combined.js

12. jQuery versions with known weaknesses: http://research.insecurelabs.org/jquery/test/
13. jquery-ui vulnerabilities: https://security.snyk.io/package/npm/jquery-ui
14. OWASP Dependency Checker: https://www.owasp.org/index.php/OWASP_Dependency_Check
15. Retire.js website: https://retirejs.github.io/retire.js/
16. Retire.js GitHub: https://github.com/RetireJS

- www.realvnc.help:
  - https://code.jquery.com/jquery-3.6.0.min.js

Info

Username Enumeration via Forgotten Password Function

Overall Risk	Informational	**Finding ID**	NCC-RVNC001-TWP
Impact	Low	**Component**	Web Application Assessment
Exploitability	Low	**Category**	Access Controls
		Status	New

Description

It was possible to enumerate users of the VNC Portal application through the forgotten password mechanism. This provided different responses depending on whether or not the supplied username was valid. An attacker could potentially use this difference in behaviour to compile a list of valid usernames which could be used as the basis for further attacks against the application.

When an invalid username was supplied the application returned, “There isn’t an account with the email address `<<email address>>`”. When a valid username was supplied the application returned, “We’ve sent an email with a secure link to change your password. Please check any Spam or Trash folder”.

This is demonstrated in the following two screenshots:

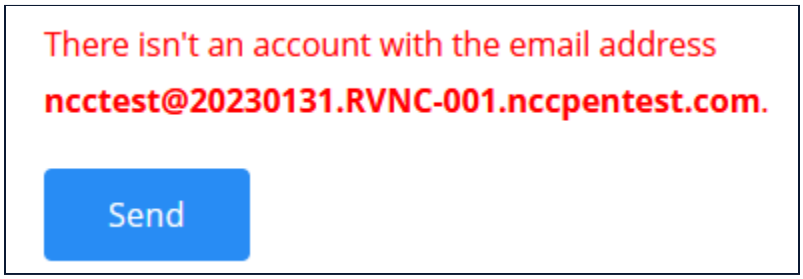

Figure 1: The message returned when the email address was not present in the database

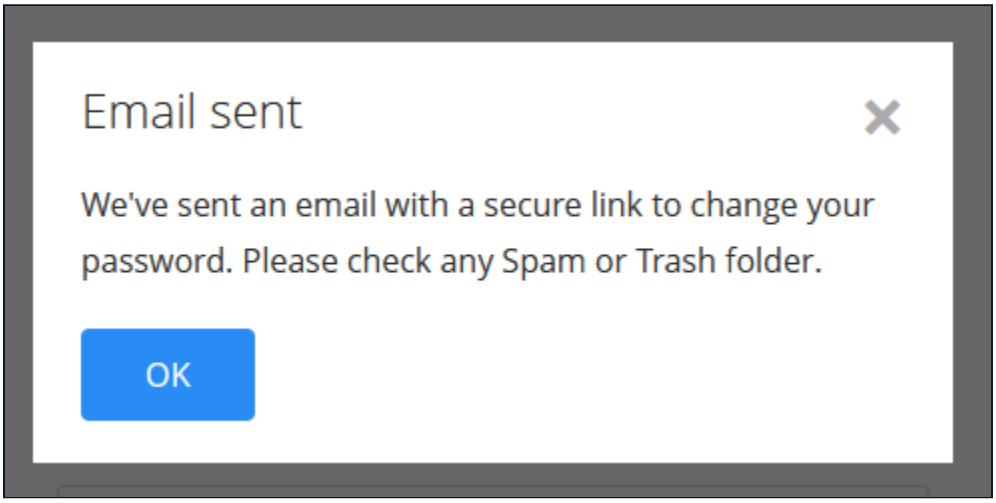

Figure 2: The message returned when an email address was present in the database

Recommendation

This issue is reported for information and awareness only. Direct changes to functionality in order to hide information on whether a given account is valid are not recommended (that is, changes to features such as login, account recovery, and password reset are not recommended). In general, the following measures are components of a strong abuse protection system [17][18]:

- Require multi-factor authentication for all users (the VNC Portal does offer this functionality to users)
- Proactively monitor and alert on abuse signals such as multiple failed logins
- Provide users with visibility and control of active sessions for their account, with information on the device and location of the login
- Implement CAPTCHAs or rate-limiting on actions which present a high risk of automated attacks (VNC Portal does have this functionality present)

Location

- https://manage.realvnc.com

17. Elie Bursztein - Account protections. A Google Perspective: https://elie.net/talk/account-protections-a-google-perspective
18. Ryan McGeehan - Account Takeover (ATO) Checklist: https://github.com/magoo/ato-checklist

Info

Session Handling Configuration

Overall Risk	Informational	**Finding ID**	NCC-RVNC001-DLP
Impact	Low	**Component**	Web Application Assessment
Exploitability	Low	**Category**	Configuration
		Status	New

Description

The VNC Portal application did not implement some recommended security-related session handling mechanisms. Concurrent logins were permitted, and the application did not invalidate the currently logged in session upon change of password.

Concurrent Logins

The VNC Portal did not prevent a particular user from logging in multiple times and creating multiple simultaneous sessions. Failure to prevent concurrent logins makes it harder for a user to identify that their account has been compromised, as illegitimate and legitimate use could occur at the same time.

Session Not Invalidated Upon Change of Password

The application did not invalidate the currently logged in session upon change of password. Failure to invalidate the session may permit a potentially compromised account to remain concurrently logged in and allow the attacker continued access to the compromised account.

Recommendation

The following recommendations should be considered:

Concurrent Logins

In general, implementing any change which forces users to only have a single active session is not recommended. Accessing an application across multiple devices or browsers is normally a common scenario which does not create a security risk for most web applications.

If permitting concurrent logins is not desirable, and considered to be a security risk, consideration could be given to configuring the application so that users are permitted to only use one session at a time. If the user authenticates again then any previously valid sessions should be immediately terminated, with an appropriate message displayed within both sessions.

Session Not Invalidated Upon Change of Password

Configure the web application to invalidate the current session upon a password change and log the user out. This will ensure that all malicious users who are concurrently logged in will automatically be logged out as well.

Location

- https://manage.realvnc.com

Info

CORS Configuration

Overall Risk	Informational	**Finding ID**	NCC-RVNC001-W4L
Impact	Low	**Component**	Web Service Assessment
Exploitability	Low	**Category**	Configuration
		Status	New

Description

The configuration of cross-origin resource sharing (CORS) on the server supporting the application was insecure. This could potentially allow an attacker to access sensitive content, or perform privileged actions within the VNC application.

CORS is a method of providing cross-domain access to resources by specifying exceptions that would not normally be permitted by the browser *Same Origin Policy*. The web browser issues a request that includes the `Origin` header. The server then responds with an `Access-Control-Allow-Origin` header. The value within `Access-Control-Allow-Origin` is used by the browser to determine whether the resource can be accessed. If the two values match then the origin is trusted.

A further header (`Access-Control-Allow-Credentials: true`) may be issued by the server, and in this case any cookies present in the request will be sent by the browser to the requested resource. With this configuration, users can be tricked into visiting a malicious web page that can make cross-domain requests for potentially sensitive information, normally only accessible in authenticated areas of an application. The attacker should receive this information in the response, since the CORS policy allows this.

There are a number of specific vulnerabilities related to CORS, all dependent on the value of the `Access-Control-Allow-Origin` header. The following configuration was observed during the assessment:

```
Access-Control-Allow-Origin: <arbitrary value>
Access-Control-Allow-Credentials: true
```

The value of the Origin header was reflected by the server in the `Access-Control-Allow-Origin` header, shown in the request and response pair below. This meant that any origin would be trusted, and any unauthenticated application content at the resource location could be accessed by an attacker. Additionally, the server permitted transmission of credentials (`Access-Control-Allow-Credentials: true`). Authenticated and potentially sensitive application data is therefore also likely to be at risk.

Request

```
GET /1.0/entryGroups HTTP/1.1
Host: s-connect-api.services.vnc.com
Content-type: application/json
Authorization: Bearer {REMOVED FOR BREVITY}
Origin: nccgroup.com
Connection: close
Content-Length: 0
```

Response

```
HTTP/1.1 200
Vary: Origin
Vary: Access-Control-Request-Method
```

```
Vary: Access-Control-Request-Headers
Access-Control-Allow-Origin: nccgroup.com
Access-Control-Expose-Headers: RealVNC-Set-Token, ETag, Link, Location, Retry-After, Via, WWW-
↳ Authenticate
Access-Control-Allow-Credentials: true
Content-Type: application/json
Date: Tue, 15 Nov 2022 17:12:29 GMT
Connection: close
Server: gateway/1.2.0-RELEASE
Content-Length: 776
```

As the API uses a bearer token for authentication through request headers, it is not possible to exploit this oversight in the CORS configuration. If the API was to move to a cookie based authentication system in the future, the current set up would pose a risk to the application's data. Due to this, the issue has been raised as informational.

Furthermore, the above was not applicable to the Update Entry endpoint:

- PATCH /1.0/entries/{entryId}

Adding an arbitrary value for the `Origin` header resulted in the server responding, "HTTP 403: Invalid CORS request".

Recommendation

CORS should be configured with whitelisting enabled to ensure only trusted origins can request application data [19][20].

Location

- https://s-connect-api.services.vnc.com

19. What is CORS (cross-origin resource sharing)? - PortSwigger: https://portswigger.net/web-security/cors
20. Cross Origin Resource Sharing - OWASP HTML5 Cheat Sheet: https://cheatsheetseries.owasp.org/cheatsheets/HTML5_Security_Cheat_Sheet.html#cross-origin-resource-sharing

9 Finding Details – WordPress Site Assessment

Low

Outdated WordPress Plugins

Overall Risk	Low	**Finding ID**	NCC-RVNC001-EW3
Impact	Low	**Component**	WordPress Site Assessment
Exploitability	Low	**Category**	Patching
		Status	New

Description

A number of plugins were found to be outdated, some of which were vulnerable. Although the vulnerabilities identified could not be exploited, any configuration changes in the future might significantly weaken the defensive posture of the organisation.

Vulnerable plugins are reported below:

Plugin	Version Used	Vulnerabilities	Current Release
sitepress-multilingual-cms (WPML)	4.5.12	CSRF (CVE-2022-45072)[21][22][23]	4.5.14[24]
ajax-search-pro	4.22	None at the time of writing.	4.23.3

```
[+] ajax-search-pro
 | Location: https://www.realvnc.com/wp-content/plugins/ajax-search-pro/
 | Last Updated: 2022-09-23T15:29:42.000Z
 | [!] The version is out of date, the latest version is 4.23.3
 |
 | Found By: Urls In Homepage (Passive Detection)
 | Confirmed By: Urls In 404 Page (Passive Detection)
 |
 | Version: 4.22 (80% confidence)
 | Found By: Readme - Stable Tag (Aggressive Detection)
 |  - https://www.realvnc.com/wp-content/plugins/ajax-search-pro/readme.txt
```

```
| [!] 1 vulnerability identified:
 |
 | [!] Title: WPML < 4.5.14 - CSRF
 |     Fixed in: 4.5.14
 |     References:
 |      - https://wpscan.com/vulnerability/8bf2529a-3fc3-47bb-959a-1f97bd6e4ec1
 |      - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2022-45072
 |      - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2022-45071
 |
```

21. WPML < 4.5.14 - CSRF - https://wpscan.com/vulnerability/8bf2529a-3fc3-47bb-959a-1f97bd6e4ec1
22. CVE-2022-45072 Detail - https://nvd.nist.gov/vuln/detail/CVE-2022-45072
23. WordPress WPML Multilingual CMS premium plugin <= 4.5.13 - Cross-Site Request Forgery (CSRF) vulnerability - https://patchstack.com/database/vulnerability/sitepress-multilingual-cms/wordpress-wpml-multilingual-cms-premium-plugin-4-5-13-cross-site-request-forgery-csrf-vulnerability-2?_s_id=cve
24. Category: WPML versions - https://wpml.org/changelog/

```
| Version: 4.5.12 (100% confidence)
| Found By: Meta Generator (Passive Detection)
|  - https://www.realvnc.com/en/, Match: 'WPML ver:4.5.12 stt'
| Confirmed By: Readme - Stable Tag (Aggressive Detection)
|  - https://www.realvnc.com/wp-content/plugins/sitepress-multilingual-cms/readme.txt
```

During the assessment period, it was not possible to find technical or proof of concept details for the WPLM CSRF vulnerability.

The WordPress base build was at its latest release version 6.1.1, at the time of writing.

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

Low

Missing/Misconfigured Security-Related HTTP Headers

Overall Risk	Low	**Finding ID**	NCC-RVNC001-XTM
Impact	Low	**Component**	WordPress Site Assessment
Exploitability	Low	**Category**	Configuration
		Status	New

Description

HTTP response headers which could be used to enhance the security posture of the VNC WordPress site were not used.

HTTP security headers

Name	Value	Setting secure
content-security-policy	frame-ancestors 'self';	✗
strict-transport-security	max-age=31536000; includesubdomains; preload	✓
x-content-type-options	Header not returned	✗
x-xss-protection	Header not returned	✗
x-frame-options	Header not returned	✗
cache-control	Header not returned	✗
access-control-allow-origin	Header not returned	✓

Figure 3: Misconfigured security headers produced using the RECX Security Analyser plugin for Google Chrome

X-Content-Type-Options

The `X-Content-Type-Options` HTTP header can be used to prevent web browsers from using content sniffing to discover a file's MIME type. Setting this header can help to protect against cross-site scripting attacks under certain circumstances.

Cache-Control

The `Cache-Control` HTTP header provides control over how responses can be cached either by proxies or by a user's browser. Using this response header can provide enhanced privacy by ensuring that sensitive content is not cached in a user's browser or intermediary proxy, where it could potentially be recovered by an attacker.

Content Security Policy

The Content Security Policy (CSP) header is a powerful mechanism for controlling the origin and behaviour of certain specified resources within a web page. Using this HTTP header can provide defence in depth from various attacks, most notably cross-site

scripting, but a degree of planning is required to address any potential conflicts with the application's implementation. Good support for CSP exists in modern browsers (thereby excluding Internet Explorer), but both the W3C standard and vendor implementations are continually evolving, and therefore support for certain individual directives will depend on the exact browser and version in use.

CSP also includes the `frame-ancestors` directive to defend against 'clickjacking' attacks, which involve an attacker using multiple transparent or opaque layers to trick a user into interacting with a web page as it appears to be displayed, when in fact the user is interacting with a different page. Use of CSP is now the recommended approach (rather than the older `X-Frame-Options` HTTP header).

X-XSS-Protection

The `X-XSS-Protection` HTTP header is supported by some older browsers and will force the enabling of any built-in cross-site scripting filters. These filters are available in Microsoft Edge up to Windows 10 1809 (October 2018) and Chrome versions up to version 78. While the built-in filters cannot be relied on solely to defend the application against input validation issues, they are a valuable addition to the defence profile of the application. It should be noted that if this header is enabled without `mode=block` then there is an increased risk that otherwise non-exploitable cross-site scripting vulnerabilities may become exploitable.

Recommendation

Consideration should be given to implementing these features by returning the following HTTP headers:

- X-Content-Type-Options: `nosniff`
- Cache-control: `no-store, no-cache`
- X-XSS-Protection: `1; mode=block`

Additionally, consider defining a list of trusted locations from which JavaScript code can be executed (along with many other restrictions) using the `Content-Security-Policy` header. As this header has many options, it should be tailored to each specific application after appropriate testing.[25]

Location

- https://www.realvnc.com

25. OWASP - List of Useful HTTP Headers: https://cheatsheetseries.owasp.org/cheatsheets/HTTP_Headers_Cheat_Sheet.html

Info

Automated Abuse Protections

Overall Risk	Informational	**Finding ID**	NCC-RVNC001-ECA
Impact	None	**Component**	WordPress Site Assessment
Exploitability	Low	**Category**	Other
		Status	New

Description

The public-facing WordPress site made use of CAPTCHA[26][27] controls to help prevent against automated attacks, including account sign up and forgotten password. However, a couple of forms were identified in which such controls were not implemented.

The following data collection forms were affected:

- https://www.realvnc.com/en/connect/competitor-comparisons/
- https://www.realvnc.com/en/newsletter/

Note that both forms submitted the data to an out of scope domain – forms.hsforms.com.

Recommendation

Consideration should be given to implementing CAPTCHA controls on the forms listed above.

A balanced approach would be to monitor for excessive activity and then, should a threshold be triggered, add a CAPTCHA to the affected form. In this way a CAPTCHA would not be requested under normal usage conditions.

Location

- https://www.realvnc.com

26. CAPTCHA - Wikipedia: https://en.wikipedia.org/wiki/CAPTCHA
27. Blocking Brute Force Attacks; Sidebar: Using CAPTCHAS - OWASP: https://owasp.org/www-community/controls/Blocking_Brute_Force_Attacks#sidebar-using-captchas

Info

Wordpress has XML-RPC Enabled

Overall Risk	Informational	**Finding ID**	NCC-RVNC001-LNH
Impact	Low	**Component**	WordPress Site Assessment
Exploitability	Low	**Category**	Configuration
		Status	New

Description

The WordPress XML-RPC interface[28][29] was accessible from the public Internet, and its exposure unnecessarily increases the overall attack surface of the website. Security best practice recommends that such an interface should only be available to trusted IP addresses.

The XML-RPC interface was identified on the following endpoint:

- POST https://www.realvnc.com/xmlrpc.php

The interface was active as demonstrated below for querying the `system.listMethods`:

Request

```
POST /xmlrpc.php HTTP/2
Host: www.realvnc.com
…
<methodCall>
<methodName>system.listMethods</methodName>
<params></params>
</methodCall>
```

Response

```
HTTP/2 200 OK
…
<?xml version="1.0" encoding="UTF-8"?>
<methodResponse>
  <params>
    <param>
      <value>
      <array><data>
  <value><string>system.multicall</string></value>
  <value><string>system.listMethods</string></value>
  <value><string>system.getCapabilities</string></value>
  <value><string>translationproxy.updated_job_status</string></value>
  <value><string>translationproxy.test_xmlrpc</string></value>
  <value><string>translationproxy.get_languages_list</string></value>
…
```

However, the risk was reduced by a number of interfaces not being enabled. For example, attempting to query the following methods (which could be used to perform credential brute-forcing) resulted in the server responding with, “XML-RPC services are disabled on this site”.

- wp.getUserBlogs

28. XML-RPC Support - Wordpress Codex: https://codex.wordpress.org/XML-RPC_Support
29. New Brute Force Attacks Exploiting XMLRPC in WordPress - Sucuri Blog: https://blog.sucuri.net/2014/07/new-brute-force-attacks-exploiting-xmlrpc-in-wordpress.html

- wp.getCategories
- metaWeblog.getUsersBlogs

This is demonstrated in the following HTTP request / response to the wp.getUserBlogs method:

Request

```
POST /xmlrpc.php HTTP/2
Host: www.realvnc.com
…
<methodCall>
<methodName>wp.getUsersBlogs</methodName>
<params>
<param><value>ncctest</value></param>
<param><value>password1</value></param>
</params>
</methodCall>
```

Response

```
HTTP/2 405 Method Not Allowed
…
<?xml version="1.0" encoding="UTF-8"?>
<methodResponse>
  <fault>
    <value>
      <struct>
        <member>
          <name>faultCode</name>
          <value><int>405</int></value>
        </member>
        <member>
          <name>faultString</name>
          <value><string>XML-RPC services are disabled on this site.</string></value>
        </member>
      </struct>
    </value>
  </fault>
</methodResponse>
```

It was noted that the `pingback.ping` method was enabled, which could be used to make WordPress send arbitrary request to any host / port; although in this instance it did not appear possible to make WordPress scan the internal network. The screenshot below is using the `pingback.ping` method to ping an NCC Group managed server.

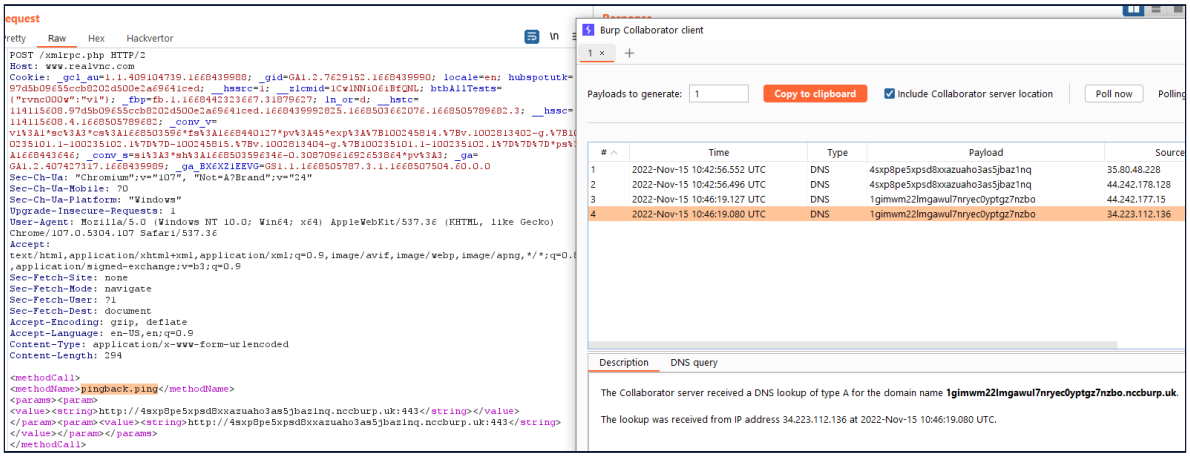

Figure 4: The XMLRPC pingback.ping method was enabled

Recommendation

Consider disabling the WordPress XML-RPC interface if it is not a business requirement.

Location

- https://www.realvnc.com/xmlrpc.php

Info

Wildcard SSL Certificate in Use

Overall Risk	Informational	**Finding ID**	NCC-RVNC001-TGW
Impact	Low	**Component**	WordPress Site Assessment
Exploitability	Low	**Category**	Cryptography
		Status	New

Description

The TLS service 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:

```
Common Name (CN)            *.realvnc.com  (request w/o SNI didn't succeed)
subjectAltName (SAN)        *.realvnc.com realvnc.com
```

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.[30][31]

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

Reproduction Steps

Visit an affected URL with a modern web browser. Inspect the certificate supplied by the site; Specifically, the `Subject` and `SAN` or `Subject Alternative Name` fields. Any entries in these fields containing wildcard characters (asterisks) indicate the use of wildcard certificates.

Location

- https://www.realvnc.com

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

Info

Outdated Version of JavaScript Dependencies

Overall Risk	Informational	**Finding ID**	NCC-RVNC001-2VH
Impact	Undetermined	**Component**	WordPress Site Assessment
Exploitability	Undetermined	**Category**	Patching
		Status	New

Description

The versions of some JavaScript libraries used by the VNC WordPress site were outdated and contained vulnerabilities that might be exploited under some scenarios. In the case of client-side JavaScript libraries, the disclosed vulnerabilities typically have low impact or in many cases are not exploitable, leading to minimal overall risk.

The following libraries were identified as being older than the current stable release version:

Library	Versions Used	Known Vulnerabilities Present?	Current Stable Version
jQuery	1.9.1	XSS	3.6.1
jQuery UI	1.12.1	XSS	1.13.2

Note that due to time constraints, an exhaustive check of JavaScript libraries was not completed. In addition, due to library complexity, it was not possible to determine whether known vulnerabilities in the instances identified above were actually exploitable.

Older versions of jQuery may be vulnerable to a number of publicly released vulnerabilities that have been assigned CVE numbers. It should also be noted that development stopped on the 1.X.X and 2.X.X branches of jQuery in 2016, with only critical security updates being released since. See footnote for information on the release of version 3[32].

Recommendation

Update the version of the affected libraries used by the web application to the latest stable and secure version available. Perform any testing necessary to ensure that this does not break or conflict with required functionality.

Embed tools such as Retire.js[33][34] and/or the OWASP Dependency Checker[35] into the development and release pipeline to gain insight into where outdated libraries are in use.

Location

- https://www.realvnc.com

32. jQuery versions with known weaknesses: http://research.insecurelabs.org/jquery/test/
33. Retire.js website: https://retirejs.github.io/retire.js/
34. Retire.js GitHub: https://github.com/RetireJS
35. OWASP Dependency Checker: https://www.owasp.org/index.php/OWASP_Dependency_Check

Info

Third Party JavaScript Library Inclusion

Overall Risk	Informational	**Finding ID**	NCC-RVNC001-M6N
Impact	Low	**Component**	WordPress Site Assessment
Exploitability	Low	**Category**	Other
		Status	New

Description

The https://www.realvnc.com website included JavaScript code from multiple external sources, which are consequently in a position to modify the behaviour of the site (whether by direct malicious intent or indirectly through a compromise). Such modification could potentially extend to the injection of malicious content (for example, crypto-coin mining scripts).

The following external scripts were referenced:

Location	External file(s)
/	• https://s39923.pcdn.co/wp-content/plugins/ajax-search-pro/js/min/external/simplebar.js • https://s39923.pcdn.co/wp-content/plugins/ajax-search-pro/js/min/plugin/optimized/asp-addons-elementor.js • https://s39923.pcdn.co/wp-content/plugins/ajax-search-pro/js/min/plugin/optimized/asp-autocomplete.js • https://s39923.pcdn.co/wp-content/plugins/ajax-search-pro/js/min/plugin/optimized/asp-core.js • https://s39923.pcdn.co/wp-content/plugins/ajax-search-pro/js/min/plugin/optimized/asp-ga.js • https://s39923.pcdn.co/wp-content/plugins/ajax-search-pro/js/min/plugin/optimized/asp-live.js • https://s39923.pcdn.co/wp-content/plugins/ajax-search-pro/js/min/plugin/optimized/asp-load.js • https://s39923.pcdn.co/wp-content/plugins/ajax-search-pro/js/min/plugin/optimized/asp-prereq.js • https://s39923.pcdn.co/wp-content/plugins/ajax-search-pro/js/min/plugin/optimized/asp-results-vertical.js • https://s39923.pcdn.co/wp-content/plugins/ajax-search-pro/js/min/plugin/optimized/asp-settings.js • https://s39923.pcdn.co/wp-content/plugins/ajax-search-pro/js/min/plugin/optimized/asp-wrapper.js • https://s39923.pcdn.co/wp-content/plugins/elementor-pro/assets/js/frontend.min.js • https://s39923.pcdn.co/wp-content/plugins/elementor-pro/assets/js/preloaded-elements-handlers.min.js • https://s39923.pcdn.co/wp-content/plugins/elementor-pro/assets/js/webpack-pro.runtime.min.js • https://s39923.pcdn.co/wp-content/plugins/elementor-pro/assets/lib/smartmenus/jquery.smartmenus.min.js

Location	External file(s)
	• https://s39923.pcdn.co/wp-content/plugins/elementor-pro/assets/lib/sticky/jquery.sticky.min.js • https://s39923.pcdn.co/wp-content/plugins/elementor/assets/js/frontend-modules.min.js • https://s39923.pcdn.co/wp-content/plugins/elementor/assets/js/frontend.min.js • https://s39923.pcdn.co/wp-content/plugins/elementor/assets/js/preloaded-modules.min.js • https://s39923.pcdn.co/wp-content/plugins/elementor/assets/js/webpack.runtime.min.js • https://s39923.pcdn.co/wp-content/plugins/elementor/assets/lib/dialog/dialog.min.js • https://s39923.pcdn.co/wp-content/plugins/elementor/assets/lib/share-link/share-link.min.js • https://s39923.pcdn.co/wp-content/plugins/elementor/assets/lib/swiper/swiper.min.js • https://s39923.pcdn.co/wp-content/plugins/elementor/assets/lib/waypoints/waypoints.min.js • https://s39923.pcdn.co/wp-content/plugins/gp-premium/general/js/smooth-scroll.min.js • https://s39923.pcdn.co/wp-content/themes/generatepress-child/js/accordion-fix.js?ver=1.6 • https://s39923.pcdn.co/wp-content/themes/generatepress-child/js/advanced-testimonial-swiper.js?ver=1.6 • https://s39923.pcdn.co/wp-content/themes/generatepress-child/js/client-pricing.js?ver=1.0.2 • https://s39923.pcdn.co/wp-content/themes/generatepress-child/js/combined-download.js?ver=1.5 • https://s39923.pcdn.co/wp-content/themes/generatepress-child/js/custom-case-study-card-swiper.js?ver=1651581505 • https://s39923.pcdn.co/wp-content/themes/generatepress-child/js/custom-swiper.js?ver=1647938070 • https://s39923.pcdn.co/wp-content/themes/generatepress-child/js/download.js?ver=1.5 • https://s39923.pcdn.co/wp-content/themes/generatepress-child/js/expandable-image-box.js?ver=1.6 • https://s39923.pcdn.co/wp-content/themes/generatepress-child/js/expandable-radio-button.js?ver=1.6 • https://s39923.pcdn.co/wp-content/themes/generatepress-child/js/expandable-table.js?ver=1.6 • https://s39923.pcdn.co/wp-content/themes/generatepress-child/js/features-table.js?ver=1.6 • https://s39923.pcdn.co/wp-content/themes/generatepress-child/js/icon-tabs.js?ver=1.6 • https://s39923.pcdn.co/wp-content/themes/generatepress-child/js/mega-menu.js?ver=1.6

Location	External file(s)
	• https://s39923.pcdn.co/wp-content/themes/generatepress-child/js/tab-sections.js?ver=1.6 • https://s39923.pcdn.co/wp-content/themes/generatepress/assets/js/menu.min.js • https://s39923.pcdn.co/wp-includes/js/dist/hooks.min.js • https://s39923.pcdn.co/wp-includes/js/dist/i18n.min.js • https://s39923.pcdn.co/wp-includes/js/dist/vendor/regenerator-runtime.min.js • https://s39923.pcdn.co/wp-includes/js/dist/vendor/wp-polyfill.min.js • https://s39923.pcdn.co/wp-includes/js/imagesloaded.min.js • https://s39923.pcdn.co/wp-includes/js/jquery/jquery-migrate.min.js • https://s39923.pcdn.co/wp-includes/js/jquery/jquery.min.js • https://s39923.pcdn.co/wp-includes/js/jquery/ui/core.min.js • https://static.zdassets.com/ekr/snippet.js?key=9bde55fa-1986-4904-b25e-51e2b0ae1a40
• `/en` • `/en/contact-us/` • `/en/discover/windows-remote-desktop-software/` • `/en/discover/ios-and-android-remote-access-software/` • `/en/raspberrypi/` • `/en/discover/linux-remote-desktop-software/`	• https://js.hsforms.net/forms/embed/v2.js
• `/en/connect/download/vnc/` • `/en/connect/download/combined/` • `/en/connect/download/viewer/`	• https://cdn-3.convertexperiments.com/js/10021806-10025517.js

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. Reports discussing successful attacks of this nature are readily available.[36]

This issue can be observed by simply visiting an affected URL using a modern browser. View the HTML source of the page, paying attention to HTML script tags, which have a `src` attribute.

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 specifies an encoded hash of the expected file, which conforming browsers will verify; for example:[37]

```
<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, see the footnotes, 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://www.realvnc.com

36. **Breach incidents:** https://www.csoonline.com/article/3253572/what-is-cryptojacking-how-to-prevent-detect-and-recover-from-it.html https://blog.sucuri.net/2014/11/the-dangers-of-hosted-scripts-hacked-jquery-timers.html https://web.archive.org/web/20160310071704/https://www.maxcdn.com/blog/bootstrapcdn-security-post-mortem/

37. Subresource Integrity (SRI): https://www.troyhunt.com/protecting-your-embedded-content-with-subresource-integrity-sri/ https://developer.mozilla.org/en-US/docs/Web/Security/Subresource_Integrity https://hacks.mozilla.org/2015/09/subresource-integrity-in-firefox-43/ https://www.w3.org/TR/SRI/ https://report-uri.com/home/sri_hash

Info

Technical Information Disclosure

Overall Risk	Informational	**Finding ID**	NCC-RVNC001-KFH
Impact	Low	**Component**	WordPress Site Assessment
Exploitability	Low	**Category**	Data Exposure
		Status	New

Description

A HTTP header produced by the public-facing WordPress site provided version information to the application gateway (ARES) used by the hosting provider, Pagely. The ARES Gateway was located between the wider Internet and the WordPress application, managing the filtering and routing of requests and responses. An attacker may use this information to gain a greater understanding of the underlying technologies involved and tailor further attacks to these specific products, e.g. a WAF bypass. Therefore, it is considered good practice to exclude information such as this from HTTP responses.

The HTTP header in the response was as follows:

- Server: Pagely-ARES/1.10.7

Recommendation

If possible, the Pagely ARES should be reconfigured so that software version information is not included in HTTP responses.

Location

- https://www.realvnc.com/

Info

WordPress Username Enumeration

Overall Risk	Informational	**Finding ID**	NCC-RVNC001-XCK
Impact	None	**Component**	WordPress Site Assessment
Exploitability	Low	**Category**	Other
		Status	New

Description

It was possible to enumerate users of the VNC WordPress site through the WordPress '`authors`' functionality. This functionality makes it possible to enumerate a list of all the usernames in use within the application, which could then be used as the basis for further attacks targeting these users and the application.

When permalinks are enabled, WordPress provides a functionality related to author archives that can be abused to list all the users of the application. This functionality was located at the following URL:

- https://www.realvnc.com/?author=1

All valid usernames could be retrieved by iterating through the author ID. The following table summarises some of the users found by abusing this functionality:

Author ID	Name
/?author=1	flickerleap
/?author=2	zonica
/?author=3	michael
/?author=4	hayley
/?author=5	matthew
/?author=8	asif
/?author=10	ben
/?author=11	realvnc
/?author=17	bogdan
/?author=20	nickc
/?author=22	agbrealvnc-com

WordPress also allows user enumeration through the following URLs:

- https://www.realvnc.com/en/news/author/ben

If the username supplied (highlighted in the example above) existed, the application answered with `200 OK`. If it did not exist, the application returned a `404 Not Found` error.

Furthermore, usernames could be enumerated by querying:

- https://www.realvnc.com/en/wp-json/wp/v2/users/?per_page=100&page=1

The above username enumeration can be automated using the tool WPScan.

It was noted that the WordPress log in page was not accessible on the public Internet. Attempting to access the following paths resulted in a `HTTP 403 Forbidden` response:

- /wp-admin/login.php
- /wp-admin/wp-login.php
- /wp-login.php

Whilst attempting other likely candidates, such as /login.php, returned `HTTP 404 Not Found`.

Recommendation

WordPress does not allow administrators to disable this functionality if permalinks are enabled. To address this issue it is recommended that the Apache ModRewrite module (or similar) should be used. The following rewrite rules can be placed in an `.htaccess` file to disallow the author enumeration issue:

```
RewriteCond %{REQUEST_URI} !^/wp-admin [NC]
RewriteCond %{QUERY_STRING} author=\d
RewriteRule ^ /? [L,R=301]
```

Similarly, the following rules can disable the second author enumeration issue:

```
RewriteCond %{REQUEST_URI} ^/+author/\w [NC]
RewriteRule ^ /? [L,R=301]
```

The WPScan[38] tool can be used to verify that this (and other WordPress vulnerabilities) have been effectively addressed. The following commands can be used to scan for username enumeration:

```
$> wpscan --url https://www.realvnc.com --enumerate u
```

Location

- https://www.realvnc.com/

38. WPScan: https://wpscan.org/

10 Contact Info

The team from NCC Group has the following primary members:

- Tatjana Sidorenko – Consultant
  tatjana.sidorenko@nccgroup.com
- James Wilde – Consultant
  James.Wilde@nccgroup.com