Report for:

Web Application and External Infrastructure Security Assessment Sample Report

ClientName

March 23, 2021

Version: 3.0

Prepared by: NCC Consultant

Email: ncc.consultant@nccgroup.com

Telephone: +44 (0)161 209 5200

Assured Service Provider

in association with National Cyber Security Centre

APPROVED SCANNING VENDOR

NCC Group PLC - Security Testing Audit and Compliance

XYZ Building,
2 Hardman Boulevard,
Spinningfields,
Manchester,
M3 3AQ
https://www.nccgroup.com

1 Executive Summary

This report presents the findings of the Web Application and External Infrastructure Security Assessment conducted on behalf of Client Company Ltd (ClientName). The assessment was conducted between 15/03/2021 and 19/03/2021 and was authorised by ClientName.

The system being assessed allowed users to log into a website to create an insurance claim, upload supporting information (such as photographs or Word documents) and review the progress of the claim. The system contained a considerable amount of Personally Identifiable Information (PII).

1.1 Overview

Several of the issues identified in the web application were assessed to expose the organisation to unnecessary risk. These were largely associated with the web application, with a lesser degree of risk associated with the external infrastructure.

Two separate application vulnerabilities could allow any user who had logged into the application to extract data from the back-end database. This data is expected to include PII such as customer names, addresses and contact information, together with details of current and resolved insurance claims. As such, any unauthorised access to this data could have regulatory implications. If a compromise which resulted in the disclosure of sensitive information became public knowledge, this could also result in reputational damage.

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
Phase 1 –Web Application Assessment	1	1	1	2	**5**
Phase 2 –External Infrastructure Assessment	0	0	1	5	**6**
Total	**1**	**1**	**2**	**7**	**11**

1.2 Assessment Summary

The most significant issue was assessed to pose a critical risk. A user who was logged into the web application could use SQL injection to extract information from the underlying database. This information is expected to be of considerable sensitivity, unauthorised disclosure of which would adversely affect the business.

The web application included a feature which allowed users to upload files providing evidence in support of claims. However, a weakness in this mechanism was assessed to pose a high risk. Users could upload a wide range of file types, including some for which there is not expected to be a business requirement. In addition, no anti-virus scanning of the uploaded files was performed. This could result in the upload of malicious files or malware. In this specific instance, it would allow an attacker to upload a file which could allow them to run commands of their own choosing on the server. This might allow them to access the sensitive information held on that server.

The web application was also found to be vulnerable to reflected cross-site scripting. Attacks of this nature can be used to target other legitimate users of an application and are typically used to recover sensitive information such as login credentials or other personal information.

The most significant vulnerability identified in the external infrastructure was the presence of administration interfaces that were accessible from the Internet. Login pages to privileged functionality should be restricted so that they are only available from a limited set of trusted IP addresses. Exposing them publicly greatly increases the opportunity for them to be attacked; for instance, with password attacks designed to gain unauthorised access. It is worth noting that the risk of automated password attacks had been mitigated in this instance by the use of an effective account lock out mechanism.

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

1.3 Strategic Recommendations

The SQL injection and reflected cross-site scripting issues which were identified have a common cause: insufficient validation of the input submitted from a web browser to the back-end systems. It is seldom possible to identify all the instances of this issue within a black box assessment of this type and so it is strongly recommended that a thorough code review of the application should be performed. It is important that any remedial activity should be applied to the entire application rather than just those instances documented here. Use of standard libraries to perform input validation or output encoding as appropriate to the application context would help to ensure that these mechanisms were consistently applied while also minimising the associated code maintenance overhead.

As this is a live system, consideration should also be given to performing a forensic investigation to establish (if possible) whether this issue has already been exploited and the extent of any compromise which may have occurred.

Review the file upload mechanism and ensure that only those file types for which there is a business need can be uploaded. In this case, this is expected to be image files, Word documents and possibly PDFs. Files not conforming to these types should be rejected or quarantined pending further investigation. In addition, it should be ensured that effective real time anti-virus scanning is performed on all uploaded files to prevent malware being uploaded and distributed.

Steps should be taken to ensure that access to privileged or administrative functionality is suitably restricted. Exposure of these interfaces to the Internet unnecessarily increases the opportunity for attack. Access should be restricted to trusted internal IP addresses only.

It is recommended that the issues set out in this report should be addressed by a structured programme of remedial actions which are prioritised in accordance with the perceived risk to the organisation. 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.

2 Table of Contents

3 Technical Summary

NCC Group was contracted by ClientName to conduct a security assessment of the web application and its associated external infrastructure in order to identify security issues that could negatively affect ClientName's business or reputation if they led to the compromise or abuse of systems.

3.1 Scope

The security assessment was carried out in the live environment and included:

- Web application assessment
- External infrastructure assessment

The IP address and URL within the scope of this test is listed below:

- 192.001/1/111 (www.clientwebsite.com)

3.2 Caveats

Due to the nature of the environment, 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.

3.3 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.

3.4 Risk Ratings

The table below gives a key to the icons and symbols 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.

Symbol	Risk Rating	CVSSv2 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.
	GOOD	N/A	Good security practices were being followed or an audit item was found to be present and correct.

3.5 Findings Overview

All the issues identified during the assessment are listed below with a risk rating for each issue.

Ref	Finding	Risk
SR-101-001	**SQL Injection**	Critical
SR-101-002	**Uploaded File Types Were Not Restricted**	High
SR-101-003	**Reflected Cross-Site Scripting**	Medium
SR-101-007	**Login to Administrative Interface Exposed**	Medium
SR-101-004	**ASP.NET Header Disclosure**	Low
SR-101-005	**HTTP Header Discloses Internal IP Address**	Low
SR-101-009	**WSDL Descriptions Could Be Obtained By Unauthenticated Users**	Low
SR-101-010	**Invalid SSL Certificate**	Low
SR-101-012	**Hosts Respond to ICMP**	Low
SR-101-013	**Directory Listings Enabled**	Low
SR-101-015	**No Support for TLS Versions Above 1.0**	Low
SR-101-014	**BEAST SSL / TLS Weaknesses**	Info

4 Technical Details

SR-101-001	SQL Injection		
Risk Rating	Critical	**Status**	Open

Description:
A critical SQL injection vulnerability was found on one PHP page, allowing full access to the site's back-end database and extraction of sensitive customer details. This vulnerability occurs when user-supplied input is used in the dynamic construction of a SQL query, without sufficient input validation being performed. This is usually a very serious vulnerability, as it effectively allows a remote attacker to execute (often arbitrary) SQL commands on the underlying database server with the privileges of the web application's database access, leaving the database open to execution of stored procedures, privilege escalation, and information retrieval. It is important to note that this issue could be exploited by unauthenticated attackers.

SQL injection vulnerabilities of this nature are frequently the cause of high-profile website breaches, so it is strongly recommended that this issue is investigated further.

The vulnerability was identified in at least one of the PHP pages (/scabc/vulnerable_page.php, job parameter).

The results of an arbitrary query could be seen in the page by using the UNION operator to inject an additional SELECT query into the job parameter of the URL. In the screenshot below, the query injected is select user():

```
http://www.nccgroup-client.co.uk/scabc/vulnerable_page.php?job=171%20union%20select%201,2,3,user%28%29,
➜  5,6,7,8,9,10,11,12,13,14,15,16%20order%20by%201%20asc--&postcode=M1%207EF&r=1145767722
```

Data could be extracted efficiently using the MySQL LIMIT and OFFSET query options, obtaining one result row per HTTP request. For example, a query such as that shown below could be used to obtain the name of the first table in the contact database (schema):

```
http://www.nccgroup-client.co.uk/scabc/jo3.php?job=
➜  171union select 1,2,3,table_name,5,6,7,8,9,10,11,12,13,14,15,16 from information_schema.tables wher
➜  e table_schema = 0x6f6e65636f6e74616374  order by 1 asc limit 1 offset 1--
➜  &postcode=M1 7EF&r=1145767722
```

The `Offset` parameter would then be incremented to get the next row. The current database was identified as contact, with `contact_agent` and `contact_test` also being present.

A list of tables for the current database was extracted as follows:

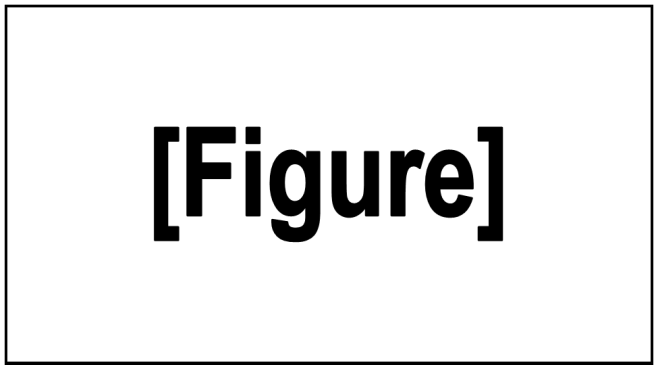

Figure 1: Extracted information

All customer data (such as addresses and security questions and answers) could be eventually be extracted from the database using this technique.

Recommendation:
The breach was possible due to vulnerabilities in the PHP source code which queried the back-end database. While only a single parameter in one PHP page was identified as vulnerable, it is possible that further such vulnerabilities exist which were not discovered within the time constraints of the current black box test. The source code of all pages should be reviewed for code patterns similar to those found in the vulnerable page, and all pages should be recoded to use prepared statements for database access. [1,2,3]

The same principle of using strict input validation and prepared statements applies equally to new development in other web frameworks such as ASP or ASP.NET. Banning particular SQL keywords in input as a fix for SQL injection is discouraged as this is often ineffective and can be evaded.

Further measures should be taken as part of a defence-in-depth strategy. Passwords should be stored as salted hashes and sensitive details (such as security questions/answers) should be stored encrypted. Users should be prevented from setting very weak passwords and be advised to not use the same passwords they use elsewhere.

In general, dynamic SQL should not be used within the application. Environments such as J2EE, ASP.NET, PHP, and Perl support the use of parameterised queries or prepared statements to ensure that the structure of the SQL statement is defined prior to entering user input.

If it is necessary to use dynamic SQL, user input should be validated and sanitised first. Numeric input should be passed through a numeric check, and string input should be fixed to escape the single quote (') character.

Affects:

IP Address	DNS Name
192.001.1.111	www.clientwebsite.com

[1] **OWASP Guidance** https://www.owasp.org/index.php/SQL_Injection https://www.owasp.org/index.php/SQL_Injection_Prevention_Cheat_Sheet https://www.owasp.org/index.php/Query_Parameterization_Cheat_Sheet
[2] **OWASP Top 10 –Injection** https://owasp.org/www-project-top-ten/2017/A1_2017-Injection
[3] **CWE-089: Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')** https://cwe.mitre.org/data/definitions/89.html

SR-101-002	Uploaded File Types Were Not Restricted		
Risk Rating	High	**Status**	Open

Description:

The application allowed any type of file to be uploaded, without performing anti-virus scanning or other content checking against the uploaded file. This meant that attackers could target other users of the application by uploading malware to the server. They might also have been able target the server itself, by uploading a web shell that would provide the attacker with the capability to run commands on the server and access the file system with the full privileges of the web server process.

It was also possible for an attacker to upload a malicious executable to the server. As a proof of concept, a web shell was uploaded to the server. Weaknesses in the manner in which these files were stored meant it was then possible to execute the web shell under the context of the user logged into the application which allowed some limited command execution. More generally, files could also be downloaded and so might be run by other regular users which could, for instance, facilitate the spread of malware.

It was also possible to upload a Flash file with an allowed extension such as .jpg; such a file could be used in a cross-site data hijacking attack.

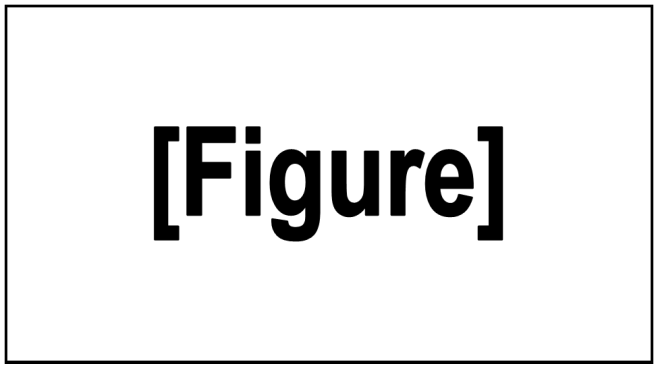

Figure 2: Flash file with amended file extension accepted for upload

It was also found that uploaded files were robustly scanned for malware. As a proof of concept, the EICAR file was uploaded. This is a benign file used to test the response of anti-virus software. The application allowed the upload of this file without any apparent server-side restriction, scanning, or removal of the file.

Recommendation:

The file types allowed to be uploaded should be restricted to only those that are necessary for business functionality.[4,5,6,7,8]

The application should perform filtering and content checking on any files which are uploaded to the server. Files should be thoroughly scanned and validated before being made available to other users. If in doubt, the file should be discarded.

[4] **OWASP Guidance** https://www.owasp.org/index.php/Unrestricted_File_Upload
[5] **EICAR Test File** http://www.eicar.org/86-0-Intended-use.html
[6] **CWE-434: Unrestricted Upload of File with Dangerous Type** https://cwe.mitre.org/data/definitions/434.html
[7] **SANS –8 Basic Rules to Implement Secure File Uploads** http://software-security.sans.org/blog/2009/12/28/8-basic-rules-to-implement-secure-file-uploads/
[8] **The Pitfalls of Allowing File Uploads on Your Website** http://blog.detectify.com/post/86298380233/the-pitfalls-of-allowing-file-uploads-on-your-website

An anti-virus solution should be used to check the uploaded file; any files flagged as potentially malicious should be discarded.

Note that the effectiveness of content-checking controls should not be wholly relied upon, and that they will only provide benefit if the anti-virus and malware signatures are regularly updated.

Some consideration should be given to storing uploaded files in a database, rather than on the file system. This would significantly reduce the risk associated with the file upload facility, but it is recognised that this might require extensive changes to the current application design.

Write permission should be removed from files and folders other than the upload folders, if these are accessible to the application. In IIS7 or higher, it is a good practice to disable or remove the dynamic extensions from the upload folders by using the "Handler Mappings" section.

Adding a "Content-Disposition: Attachment" header to static files such as PDF and document files will secure the website against Flash-based cross-site data hijacking. This can be done by using "HTTP Response Headers" in IIS for the upload folders.

Affects:

IP Address	DNS Name
192.001.1.111	www.clientwebsite.com

SR-101-003	Reflected Cross-Site Scripting		
Risk Rating	Medium	**Status**	Open

Description:

The application 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 the case of a GET request, this means that a URL which appears to be associated with the site (and therefore trustworthy to regular users) could contain malicious code that would be executed by the user's browser within the context of the application when the link is visited. In the case of a POST request, a victim user would have to first be coerced to an otherwise unrelated site which then launches attack using a form.

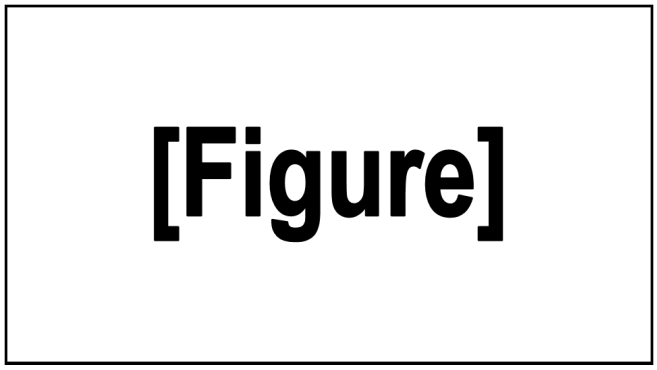

Figure 3: XSS payload (in this case, a simple message) being executed under user's context

```
Evidence of affected URLs, parameters, requests and responses has been redacted.
```

Reflected cross-site scripting vulnerabilities are extremely common in web applications but can have a serious impact. They 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.

Recommendation:

Reliable avoidance of cross-site scripting vulnerabilities should consist of two stages - input validation and output encoding.[9,10,11]

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 number field, for example, the whitelisted 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

[9] **OWASP XSS References** https://www.owasp.org/index.php/Cross-site_Scripting_(XSS) https://www.owasp.org/index.php/XSS_(Cross_Site_Scripting)_Prevention_Cheat_Sheet

[10] **OWASP Top 10 –Cross-Site Scripting** https://owasp.org/www-project-top-ten/2017/A7*2017-Cross-Site_Scripting*(XSS)

[11] **CWE-079: Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')** https://cwe.mitre.org/data/definitions/79.html

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.

The pages listed should be modified to handle malicious data properly in the associated fields.

Affects:

IP Address	DNS Name	Page	Parameters
192.001.1.111	www.clientwebsite.com	vulnerablepage.html	aed222, aed223

SR-101-007	Login to Administrative Interface Exposed		
Risk Rating	Medium	**Status**	Open

Description:

A number of login pages for accessing administrative interfaces were identified during testing of the external infrastructure. This is against security best practice which recommends that login pages to access such interfaces should be only available to trusted IP addresses.

The following login pages were identified:

```
Nccgroup-client.co.uk/login
```

Recommendation:

Implement access control lists which only allow access from trusted IP addresses.

Affects:

IP Address	DNS Name
192.001.1.111	www.clientwebsite.com

SR-101-004	ASP.NET Header Disclosure		
Risk Rating	Low	**Status**	Open

Description:
Various headers produced by the application provide information about the software installed on the web server. An attacker may use this information to gain a greater understanding of the underlying technologies involved and tailor further attacks to these specific products. It is therefore good practice to exclude information such as this from HTTP responses.

An example HTTP response from the server is shown below:

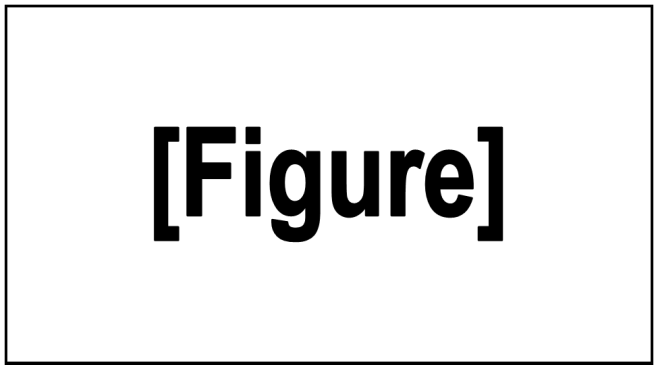

Figure 4: HTTP response

The highlighted headers revealed the exact version of ASP.NET which is in use.

Recommendation:
The web server should be reconfigured so that software version information is not included in HTTP responses.[12, 13, 14, 15, 16]

- For IIS 7.5 and later, the URL Rewrite HTTP module available from Microsoft can be configured to remove the Server header from IIS responses.
- The X-Powered-By header can be removed via the "Custom HTTP Headers" section of IIS Manager for the relevant site.
- The X-AspNet-Version header can be removed by adding the following node to the

```
<httpRuntime enableVersionHeader="false" />
```

- The X-AspNetMvc-Version header can be removed by adding the following line of code to the ApplicationStart method of the application's Global.asax.cs file:

```
MvcHandler.DisableMvcResponseHeader = true;
```

[12] CWE-200: Information Exposure: https://cwe.mitre.org/data/definitions/200.html
[13] MSDN - Remove Unwanted HTTP Response Headers: http://blogs.msdn.com/b/varunm/archive/2013/04/23/remove-unwanted-http-response-headers.aspx
[14] Removing Unnecessary HTTP Headers in IIS and ASP.NET: http://www.4guysfromrolla.com/articles/120209-1.aspx
[15] OWASP Examples: https://www.owasp.org/index.php/Testing_for_Web_Application_Fingerprint_(OWASP-IG-004)
[16] Change or modify a Response Header value using URL Rewrite: http://blogs.msdn.com/b/benjaminperkins/archive/2012/11/02/change-or-modify-a-response-header-value-using-url-rewrite.aspx

Affects:

IP Address	DNS Name
192.001.1.111	www.clientwebsite.com

SR-101-005	HTTP Header Discloses Internal IP Address		
Risk Rating	Low	**Status**	Open

Description:
It was possible to determine an internal IP address by sending a crafted request to the application. An attacker may use this information to gain a greater understanding of the internal network and tailor further attacks.

The HTTP request and response shown below illustrate this issue.

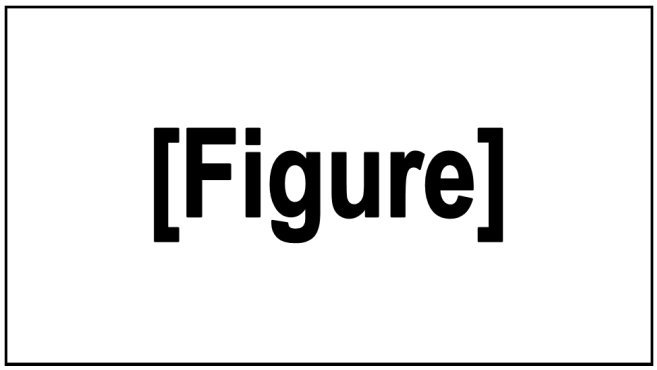

Figure 5: HTTP response

Recommendation:
On IIS version 7 and above, this issue can be addressed by setting the IIS alternateHostname property (as described at the www.iis.net link below). This allows the specified hostname to be used in place of the internal IP address in redirection responses. Testing should be performed to ensure that this does not adversely affect any legitimate redirection functionality in the application.[17, 18, 19]

Affects:

IP Address	DNS Name
192.001.1.111	www.clientwebsite.com

[17] CWE-200: Information Exposure: https://cwe.mitre.org/data/definitions/200.html
[18] Server Runtime <serverRuntime>: http://www.iis.net/configreference/system.webserver/serverruntime
[19] Microsoft Support –FIX: http://support.microsoft.com/kb/834141

SR-101-009	WSDL Descriptions Could Be Obtained By Unauthenticated Users		
Risk Rating	Low	**Status**	Open

Description:

It was possible to retrieve Web Services Description Language (WSDL) from web service endpoints as an anonymous user. Whilst this functionality could be of use to a legitimate developer, it would also help an attacker to determine the methods exposed by a service and how to create a well-formed request.

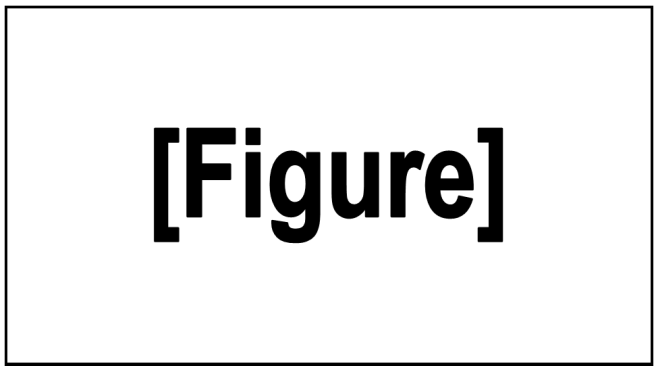

Figure 6: WSDL file

Recommendation:

For WCF services, the displaying of exception details is largely controlled with the "*serviceMetadata*" property, which is set in the web.config file. To prevent the download of WSDL descriptions, the configuration directive should be set as shown:[20,21,22]

```
<serviceDebug includeExceptionDetailInFaults="false"/>
```

Alternatively, the URLScan tool, the IIS Request Filtering feature, or a web application firewall could be used to block incoming requests for WSDL descriptions.

Affects:

IP Address	DNS Name
192.001.1.111	www.clientwebsite.com

[20] <serviceMetadata> behaviour: http://msdn.microsoft.com/en-us/library/ms731317(v=vs.110).aspx
[21] IIS Request Filtering: http://www.iis.net/configreference/system.webserver/security/requestfiltering
[22] Microsoft's Free Security Tools –URLScan Security Tool: http://blogs.technet.com/b/security/archive/2013/01/22/microsoft-s-free-security-tools-urlscan-security-tool.aspx

SR-101-010	Invalid SSL Certificate		
Risk Rating	Low	**Status**	Open

Description:

The SSL certificate installed on the service was invalid, and would not be trusted by connecting applications. This would make a man-in-the-middle attack easier to perform.

The certificate was invalid because:

- It was self-signed
  - The endpoint cannot be validated because the certificate has not been issued by a recognised certificate authority

A screenshot showing the certificate and highlighting the relevant detail can be seen in Supplemental Data - SSL Certificate Details on page 26.

An end user attempting to contact the secure service would likely be presented with a certificate warning, which to most users would be indistinguishable from the warning associated with a fraudulent certificate. The chance of a successful man-in-the-middle attack against the remote host is therefore increased. Such an attack would allow the attacker to view and edit data in transit without the knowledge of the end user.

In addition, a certificate warning being issued by the browser could reduce user confidence in the site's security.

Recommendation:

A new SSL certificate should be purchased or generated to replace the existing one. The certificate should be signed by an established certificate authority and cover all the hosts on which it will be installed (preferably using Subject Alternative Names where multiple hostnames are required).[23, 24, 25, 26]

Affects:

IP Address	DNS Name
192.001.1.111	www.clientwebsite.com

[23] Crying Wolf: An Empirical Study of SSL Warning Effectiveness: http://lorrie.cranor.org/pubs/sslwarnings.pdf

[24] Benign security warnings have trained users to ignore them: http://arstechnica.com/security/2009/07/benign-security-warnings-have-trained-users-to-ignore-them/

[25] Alice in Warningland: A Large-Scale Field Study of Browser Security Warning Effectiveness: https://www.usenix.org/conference/usenixsecurity13/technical-sessions/presentation/akhawe

[26] NCC Group whitepaper on the configuration of SSL/TLS services: https://www.nccgroup.trust/en/learning-and-research-centre/white-papers/how-organisations-can-properly-configure-ssl-services-to-ensure-the-integrity-and-confidentiality-of-data-in-transit/

SR-101-012	Hosts Respond to ICMP		
Risk Rating	Low	**Status**	Open

Description:

Hosts were found to respond to ICMP messages. Further information has been included in Supplemental Data - ICMP Details on page 27.

Recommendation:

All inbound and outbound ICMP messages should be restricted at border routers and firewall devices; if they are specifically required for legitimate business reasons then access control lists for these services should be applied.

Note that completely disabling ICMP may restrict PMTU discovery and might affect other applications on the network. The ICMP requirements for the network should be considered before implementing any changes to filtering controls.

Affects:

IP Address	DNS Name
192.001.1.111	www.clientwebsite.com

SR-101-013	Directory Listings Enabled		
Risk Rating	Low	**Status**	Open

Description:

Directory listings were enabled for the site. This allows an attacker to browse directories for which there is no index page (e.g. index.html, index.php, or default.aspx) present.

This can be used to gain knowledge of the file and folder structure, and makes it easy to look for information that may be useful when carrying out a targeted attack, such as the specific frameworks or libraries in use. It may also lead to the discovery of sensitive files which, without directory listings, would be much harder to find by brute-force.

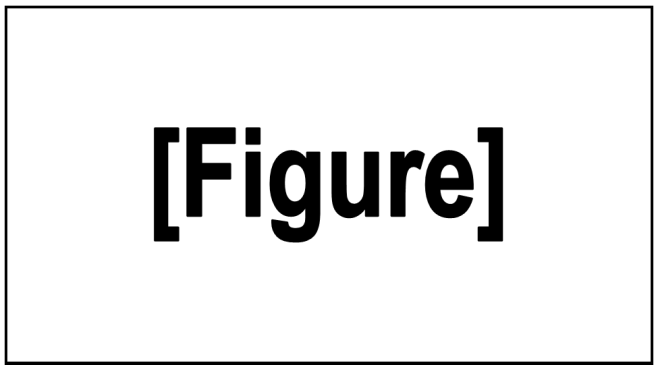

Figure 7: Screenshot of file structure

Recommendation:

Directory listings should be disabled.[27,28]

For IIS, this can be achieved in IIS Manager via the Directory Browsing section - the "Disable" option should be selected. Alternatively, the change can be carried out by direct modification of the relevant web.config file, in the system.webServer/directoryBrowse attribute.[29]

Affects:

IP Address	DNS Name
192.001.1.111	www.clientwebsite.com

[27] CWE-548: Information Exposure Through Directory Listing: http://cwe.mitre.org/data/definitions/548.html
[28] Enable or Disable Directory Browsing in IIS 7: http://technet.microsoft.com/en-us/library/cc731109(WS.10).aspx http://technet.microsoft.com/en-us/library/cc731109(WS.10).aspx
[29] Directory Browse setting: http://www.iis.net/configreference/system.webserver/directorybrowse

SR-101-015	No Support for TLS Versions Above 1.0		
Risk Rating	Low	**Status**	Open

Description:

The affected server did not support TLS above version 1.0. Since 2008, TLS has been at version 1.2, with version 1.3 being defined in August 2018. Versions 1.2 and 1.3 of TLS are more resistant to known attacks, and TLSv1.2 supports more modern cipher suites that are widely acknowledged to offer the best cryptography available for securing Internet connections. TLSv1.3 further improves on security by removing unsafe or unused features, eliminating unnecessary handshake steps and forcing the use of newer encryption methods. t should be noted that use of TLS 1.0 and 1.1 is now flagged as insecure (such as by browser warnings) by the major browsers, and in January 2021 the NSA issued guidance urging system administrators to only provide support for TLS versions 1.2 or 1.3.[30]

TLS 1.0 was the highest protocol version supported, as shown in Supplemental Data - SSLScan Output on page 25.

Recent versions of all web browsers support 1.2 and 1.3, and therefore supporting these protocols server-side offers better security for those clients using modern browsers. TLSv1.2 also supports a class of cipher suites that offer Authenticated Encryption with Associated Data (AEAD). These cipher suites include an authenticated integrity check within the encryption operation, and are resistant to more attacks than their older counterparts. Google's Chrome browser states that the security of connections to websites that do not use AEAD ciphers is "obsolete". Currently, this is not presented as an error to the user, merely as information for anyone viewing the page's security report (in the browser's developer console). However, as Google continues to push for the rapid adoption of more robust Internet cryptography, this message may become more prominent.

TLSv1.3 was approved for use in August 2018 and goes to further enhance security. The security benefits of TLSv1.3 include the removal of unsafe or unused features, elimination of unnecessary handshake steps and the forced use of newer encryption methods. As well as preventing encryption downgrade attacks, the streamlined approach to session initiation will offer performance gains over previous TLS implementations.

Supporting TLS 1.0 as a security control is non-compliant with the Payment Card Industry (PCI) Data Security Standard (DSS). While these regulations may not be directly relevant, it should be expected that this directive will become security best practice and may also be reflected in more obvious ways, such as web browser warnings.

Refer also to the finding SR-101-014 on page 24.

Recommendation:

Add support for TLS v1.2 and v1.3.[31,32,33,34,35,36,37]

Support for TLS version 1.0 should be disabled unless there is a specific business requirement to allow users of older, less secure browsers to continue to connect. Consider also disabling support for version 1.1, which will become unsupported by major browsers in the near future.

All affected hosts should be configured to prefer the latest cipher suites that they offer, such as AES-GCM (TLS 1.2 and 1.3 only).

The reference from Mozilla below provides recommendations on cipher suite ordering based on the profile of connecting clients.

[30] Eliminating Obsolete Transport Layer Security (TLS) Protocol Configurations - https://media.defense.gov/2021/Jan/05/2002560140/-1/-1/0/ELIMINATING_OBSOLETE_TLS_UOO197443-20.PDF
[31] Mozilla Server Side TLS: https://wiki.mozilla.org/Security/Server_Side_TLS#Recommended_configurations
[32] Obsolete Cipher Suites: https://www.chromium.org/Home/chromium-security/education/tls#TOC-Obsolete-Cipher-Suites
[33] OWASP Transport Layer Protection Cheat Sheet: https://cheatsheetseries.owasp.org/cheatsheets/Transport_Layer_Protection_Cheat_Sheet.html
[34] NCC Group Whitepaper on the Configuration of SSL/TLS Services: www.nccgroup.com
[35] PCI Security Standards Council: https://www.pcisecuritystandards.org/documents/Migrating-from-SSL-Early-TLS-Info-Supp-v1_1.pdf
[36] Saying Goodbye to SSL/Early TLS: blog.pcisecuritystandards.org
[37] Overview of TLS v1.3: https://owasp.org/www-pdf-archive/OWASPLondon20180125_TLSv1.3_Andy_Brodie.pdf

Affects:

IP Address	DNS Name
192.001.1.111	www.clientwebsite.com

SR-101-014	BEAST SSL / TLS Weaknesses		
Risk Rating	Informational	**Status**	Open

Description:

A vulnerability that could allow information disclosure exists in SSL and version 1.0 of TLS. The weakness is caused by an improper choice of initialisation vector (IV) used by block ciphers operating in cipher block chaining (CBC) mode. The exploit that takes advantage of the vulnerability is known as "browser exploit against SSL/TLS" (BEAST). BEAST allows attackers to compromise the confidentiality of connections to reveal short sections of plaintext, with session cookies being the most likely target. The potential for the BEAST vulnerability has been reported here because the servers were seen to support block ciphers in CBC mode operating under vulnerable SSL/TLS protocol versions.

The BEAST attack itself is a client-side attack and requires the attacker to not only be in a position to inspect the encrypted traffic but also to initiate crafted requests made from the victim's browser. In addition to this requirement, the major browser vendors have implemented client-side fixes for the IV flaw (Apple being the last to do so in October 2013) and thus users with up-to-date browsers should not be affected. For these reasons the issue has been rated as informational.

Recommendation:

No server-side remedial action can fully eliminate the conditions necessary for a successful BEAST attack. While disabling SSL is recommended, removing support for TLS version 1.0 could prevent some users from accessing the service.[38,39,40]

While prioritising the use of cipher suites based on the RC4 stream cipher would mitigate BEAST, RC4 has been shown to suffer from cryptographic flaws that mean this action is not a recommended solution.

TLS versions 1.1 and 1.2 are not susceptible to the weak IV design. However, while supporting them is recommended, that does not specifically resolve the BEAST attack because the man-in-the-middle attacker can launch what is known as a "protocol downgrade attack". In this attack the man-in-the-middle interferes with the TLS connection to try to force browsers to use lower TLS versions that are vulnerable to BEAST. This downgrade attack can be mitigated by supporting the TLS_FALLBACK_SCSV mechanism, but this must also be supported by the user's browser for the mitigation to work.

Affects:

IP Address	DNS Name
192.001.1.111	www.clientwebsite.com

[38] Original BEAST Attack: http://vnhacker.blogspot.co.uk/2011/09/beast.html
[39] NCC Group Whitepaper on the Configuration of SSL/TLS Services: https://www.nccgroup.trust/en/learning-and-research-centre/white-papers/how-organisations-can-properly-configure-ssl-services-to-ensure-the-integrity-and-confidentiality-of-data-in-transit/
[40] SSL Labs: https://community.qualys.com/blogs/securitylabs/2013/09/10/is-beast-still-a-threat https://community.qualys.com/blogs/securitylabs/2013/10/31/apple-enabled-beast-mitigations-in-os-x-109-mavericks

5 Supplemental Data - SSLScan Output

```
Redacted.
```

6 Supplemental Data - SSL Certificate Details

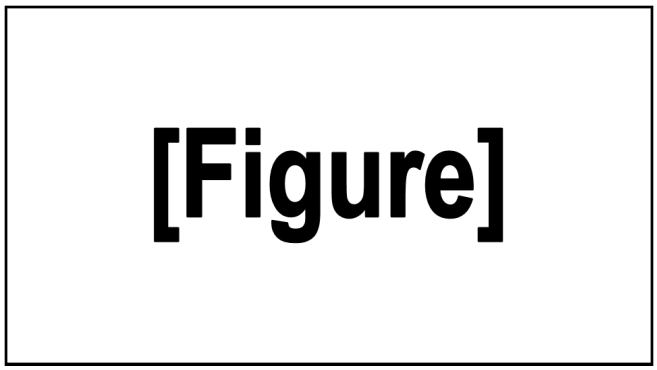

Figure 8: SSL Certificate with invalid fields highlighted

7 Supplemental Data - ICMP Details

Redacted.

8 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 Client Company Ltd (ClientName).

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

8.1 Document Data

Data Classification	Client Confidential
Client Name	ClientName
Project Reference	99999
Proposal Reference	P99999-Sample
Document Title	Web Application and External Infrastructure Security Assessment Sample Report
Author	NCC Consultant

8.2 Document History

Issue No.	Issue Date	Issued by	Change Description
0.1	2018-01-15	NCC Consultant	Draft for NCC Group internal review only
0.2	2018-01-16	A Reviewer	Revised QA
1.0	2018-01-16	NCC Consultant	Released to client
2.0	2018-02-14	NCC Consultant	Cover page address updated
3.0	2021-03-23	NCC Consultant	Minor content and format changes

8.3 Document Distribution List

Name	Role
A. Sponsor	Project Sponsor, ClientName
NCC Consultant	Senior Security Consultant, NCC Group
A. Manager	Account Manager, NCC Group

9 Assessment Team

The following members of staff were assigned to this assessment:

Name	Job Title	Comments
NCC Consultant	Senior Security Consultant	CREST Registered Tester (CRT), CHECK Team Member (CTM)