February 2022

NISM Unit 11

Group 3 Executive Summary

Jonathan, Ying, Haseeb

Table of Contents

1. Summary of the work carried out

Group 3 assigned a website (customersrus.co.uk), a customer relationship management system by SugarCRM Community Edition, the free version of SugarCRM products. No prior information about the system infrastructure, application source code and an outside scan black-box approach was needed.

1.1 Tools and tests used

We used Kali Linux with the pre-installed packages of penetration testing applications. OWASP penetration test methodology gathered information about the target, evaluated configuration and deployment management by considering the server security, and tested web application vulnerabilities that would offer us technical reports.

We evaluated various tools to carry out the work, namely NMAP, Burp Suite, Metasploit, Nessus Essentials, SQLMap, Wapiti, Nikto, OWASP ZAP, Dimitry, Knockpy, Hydra, and Skipfish. We prioritised these tools considering manual or automated testing.

1. OWASP ZAP

   OWASP ZAP was used to discover directory and file paths and execute active scans (Dahle, 2020) to scan web application security. The automation permitted ease of setup and used a proxy feature to manipulate the traffic past the WAF. The spider feature allows mapping web application files and directories paths (Joao, 2021). In identifying STRIDE all aspects, which was imperative to evaluate our hypothesis. ZAP provided a comprehensive technical report of 195 vulnerabilities, including high-risk SQL injection, medium risk buffer overflows and low-risk absence of Anti-CSRF tokens.

2. Nmap

   NMAP allowed the capability to discover IP addresses and open ports. Targeted information gathering occurred in the first phase of the penetration test process (Kaur, 2017). The command-line tool manually allows tests using a script engine to enable users to write and automate scripts to practically detect and avoid being discovered by firewalls and IDS (Joao, 2021). Considering STRIDE, we identified Spoofing and Information Disclosure where authentication could be compromised, leading to the breach of sensitive data on the Sugar CRM site.

3. Burp Suite

   Burp suite features a proxy, sequencer, intruder, decoder, comparer, and repeater, to analyse the server's response, encoded hashes, compare data, intercept and modify traffic, validate tokens, and customised automated attack (Joao, 2021). Using STRIDE, we found cleartext submission of password and JavaScript dependency vulnerability findings could identify spoofing, tampering, repudiation and denial of service due to poor patch management.

4. Nikto

   Karangle et al. (2019) highlighted that Nikto could find information about installed software and web server in surface testing. Nikto tested server misconfiguration, outdated software, and insecure files. Using a manual approach was essential for identifying tampering in STRIDE analysis.

5. Hydra

   Hydra was used to perform a brute force attack, with its capability to brute-force SSH and crack passwords by trying login with various passwords on our login page. Kakarla et al. (2018) mentioned that successful attempts could take minutes. Regarding STRIDE, Hydra supports finding out the Spoofing threat as if we could access the website while not authorised.

1.2 Summarised results

The initial Part 1 Design Document methodology gathered information about the target website; we used Nikto, Burp Suite, and NMAP to discover IP address, web server, operation system, ports, and protection, as shown below.

IP Address	• 66.66.247.187
Operating System	• Linux Kernel 2.6 (RedHat Enterprise Linuz 7)
Web Server	• Apache
Website	• customersrus.co.uk
Web App Version	• SugarCRM Community Edition 6.5.25
Open Ports	• 21, 53, 80, 110, 143, 443, 465, 587, 993, 995, 2525, 3306, 5432
Filtered Ports	• 25, 135, 139, 445, 3372, 3389
Security Protection	• Imunify360 (Cloud Linux)

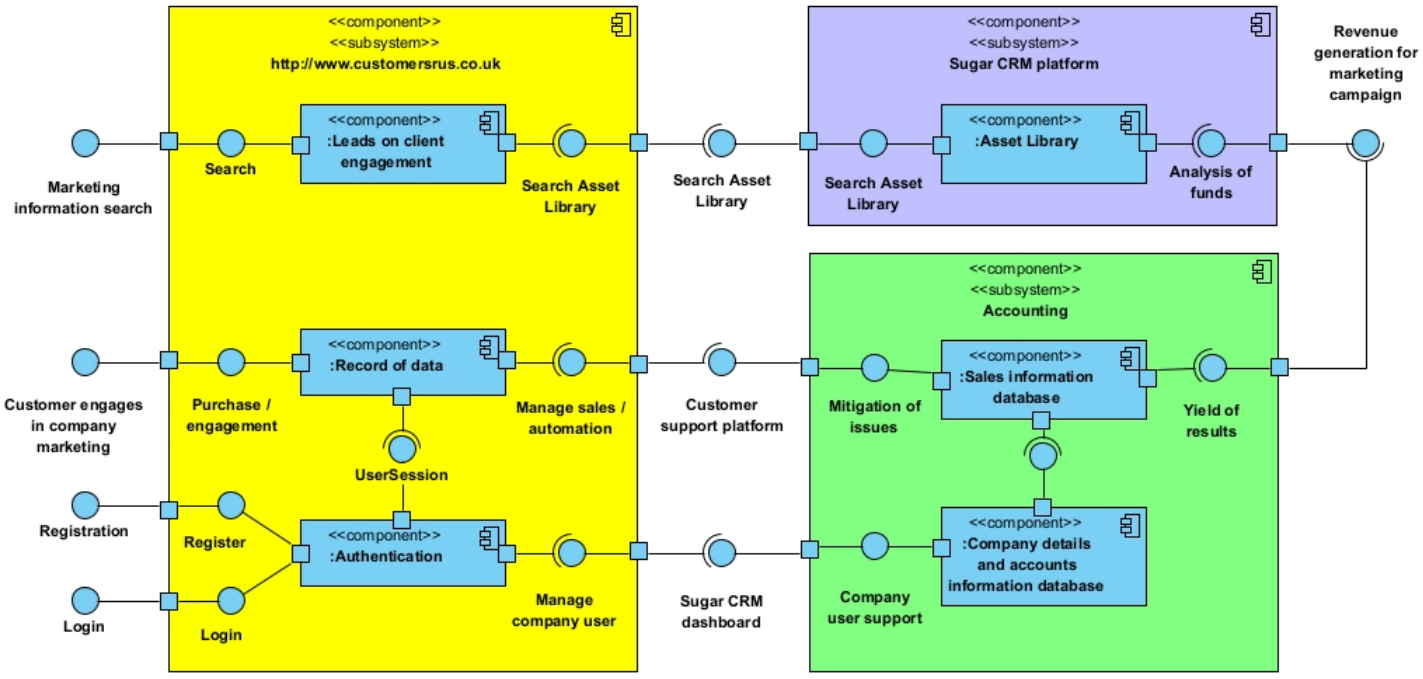
1.3 Summary of detected vulnerabilities

Tool	Detected vulnerability
OWASP ZAP	• SQL Injection • Buffer Overflow • Vulnerable JavaScript library • Absence of Anti-CSRF Tokens • Cookie No HttpOnly Flag • Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s) • SMTP Service Cleartext Login Permitted
Nmap	• Acquired all open ports
Burp Suite	• Cleartext submission of password • Vulnerable JavaScript dependency
Nikto	• Clickjacking X-Frame-Options • XSS-Protection header • X-Contenet-Type-Options header • Unsterilised PHP code execution

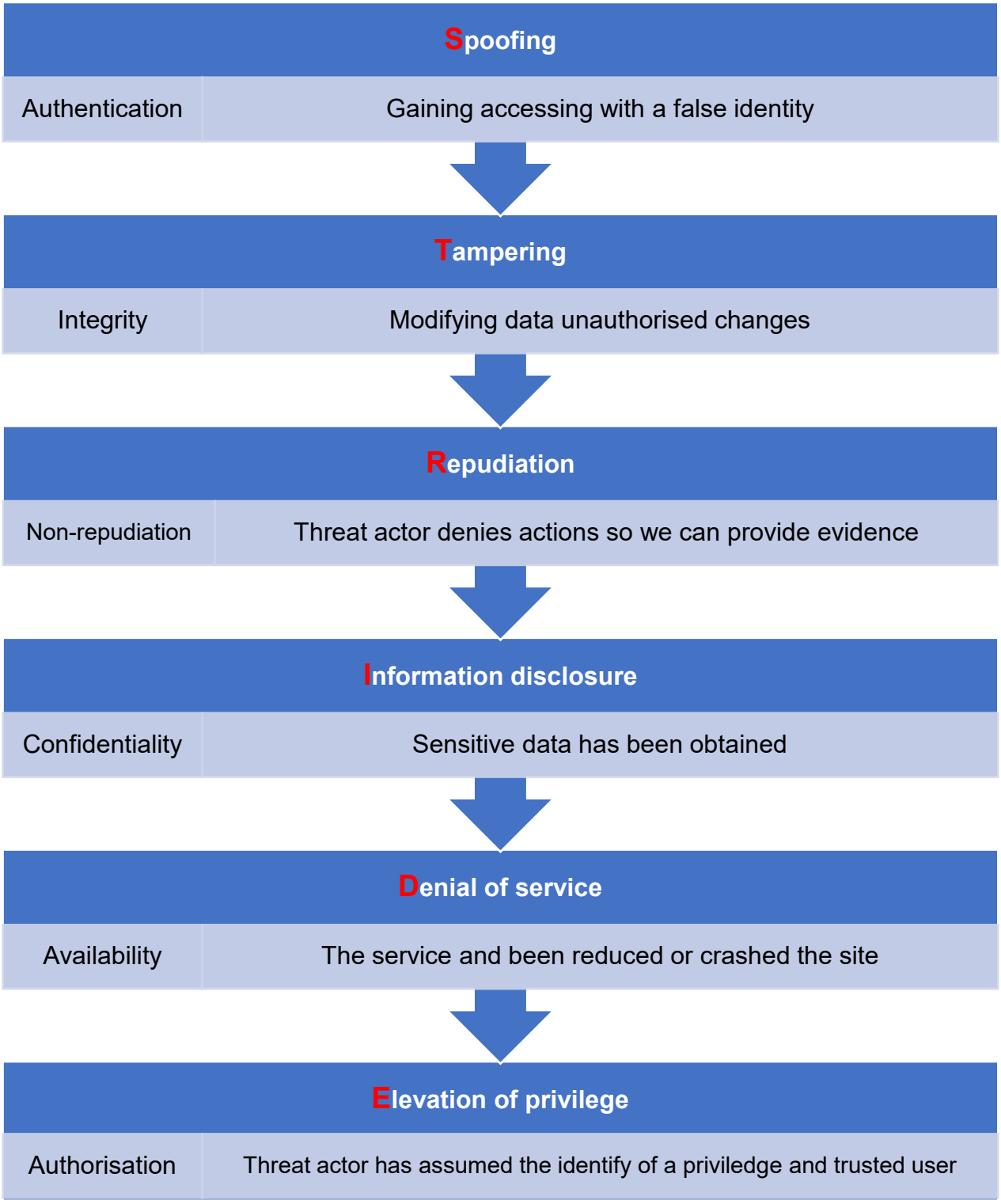
2. Summary findings (non-technical)

2.1 Examining the use case scenario of the Sugar CRM platform and user access.

Use case UML diagram: Evaluation of the target (Sugar CRM)

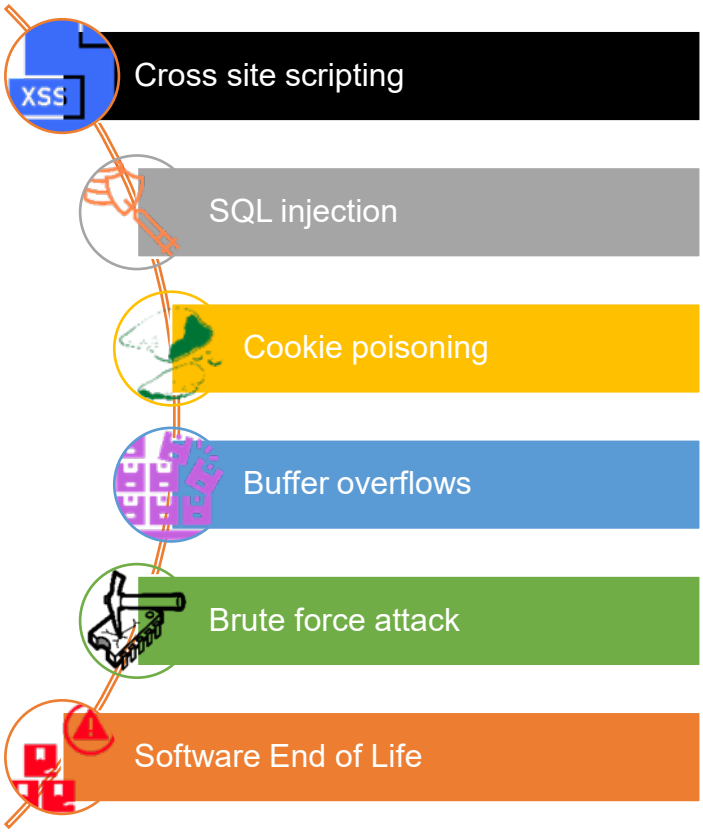


2.2 Using the STRIDE method to identify threats.



We offered a STRIDE threat analysis and identified the significant vulnerabilities specified below.

2.3 Scanning results of vulnerabilities



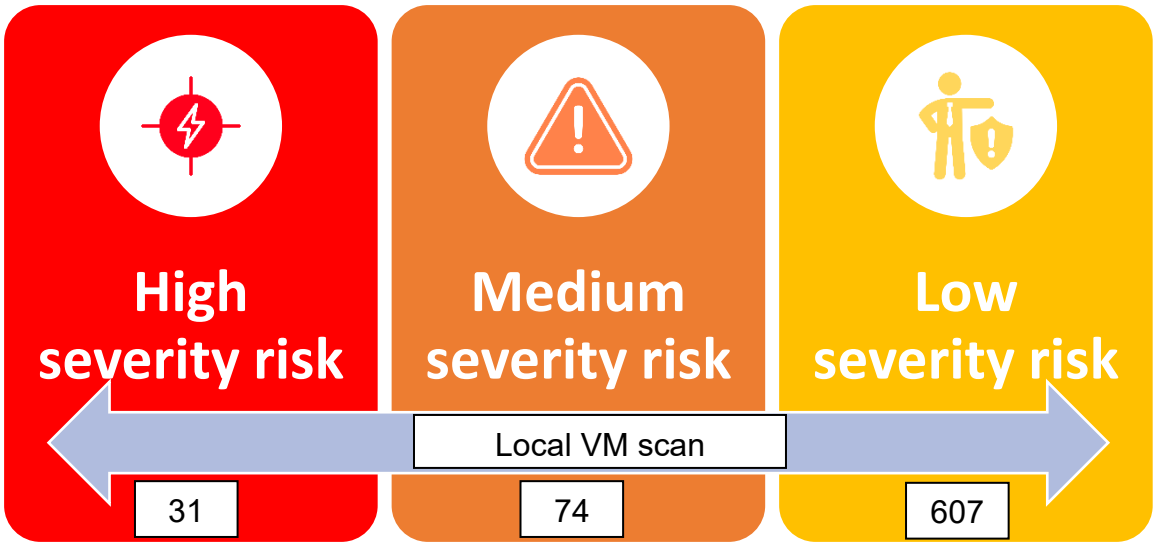
2.4 Impact to business assets with STRIDE and threat findings

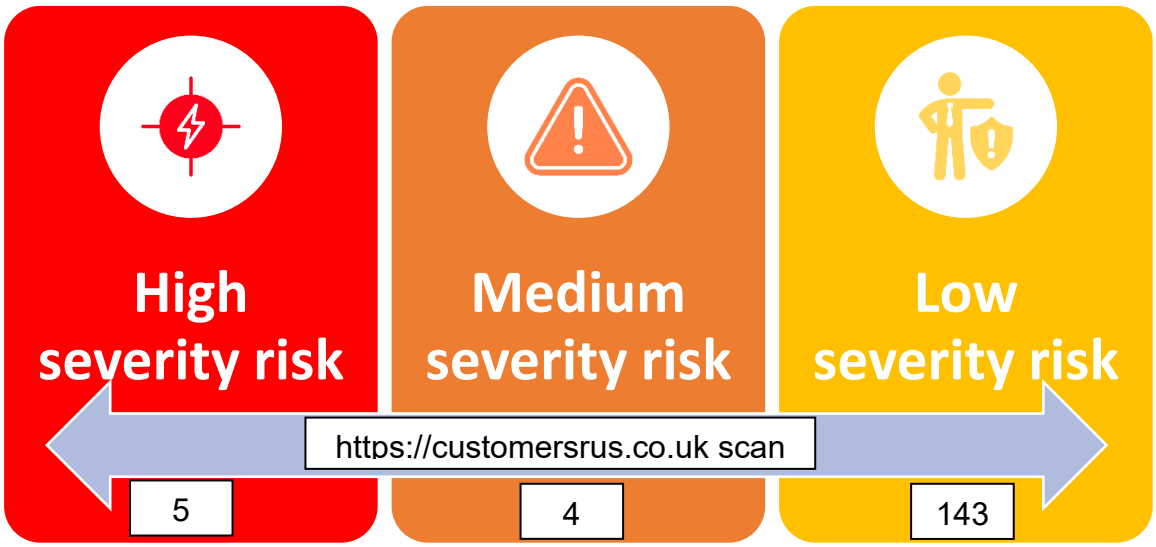
Communication	• Impersonating - SRIE • Tampering - T
Accounts / Database	• Unauthorised accesss - TIE
Credentials	• Impersonating - SRIE • Making changes - T • Unauthorised access - TIE
Logging	• Modifying events - TR • Unauthorised access - SIE
Configurations	• Tampering - STI • Unauthorised access - IE
Certificates & Keys	• Unauthorised changes - STIE

2.5 Risk Assessment of vulnerabilities found

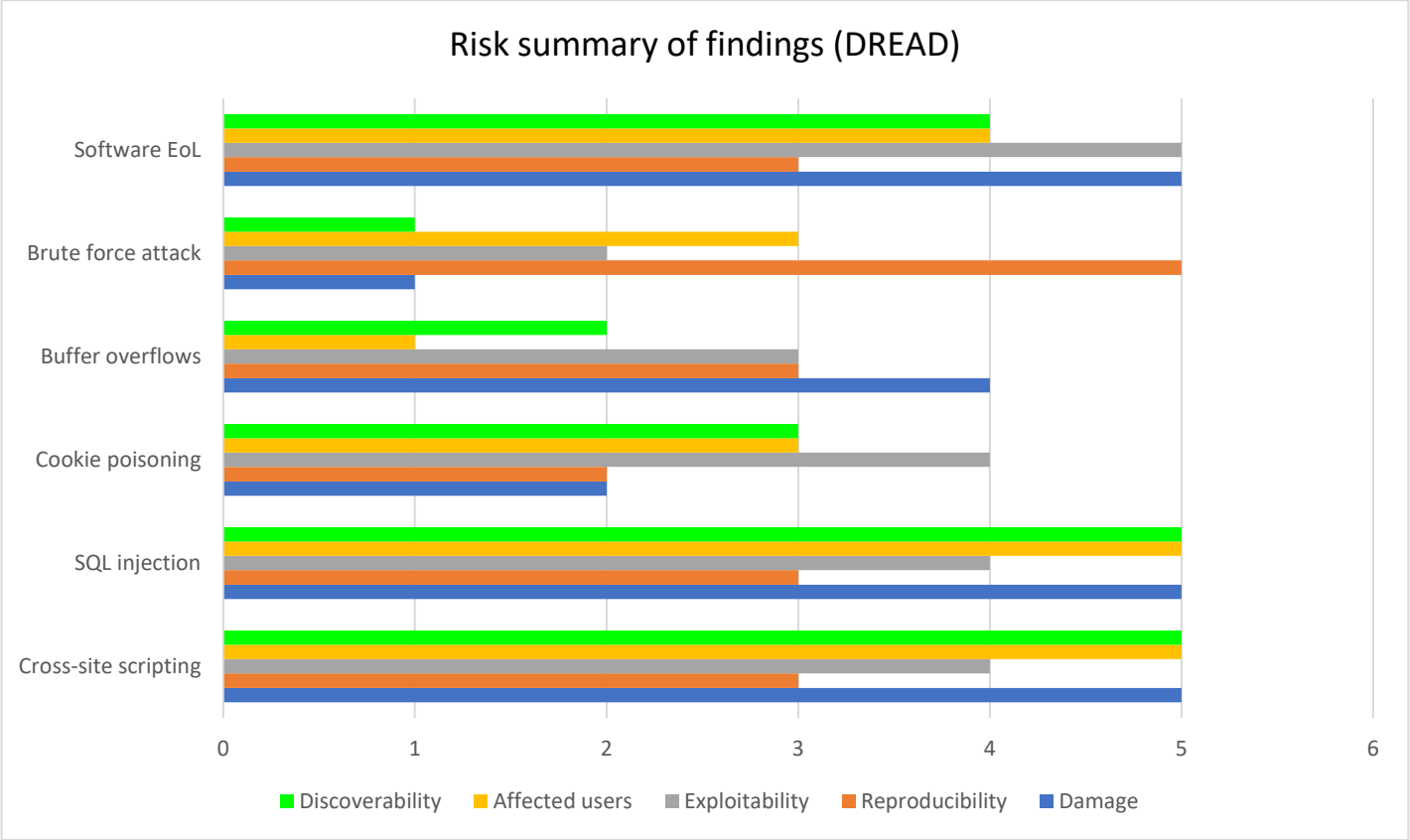
Below are categorised risks that could have an impact on business operations. High severity requires immediate attention and is potentially relatively easy for attackers to exploit and provide control of systems.

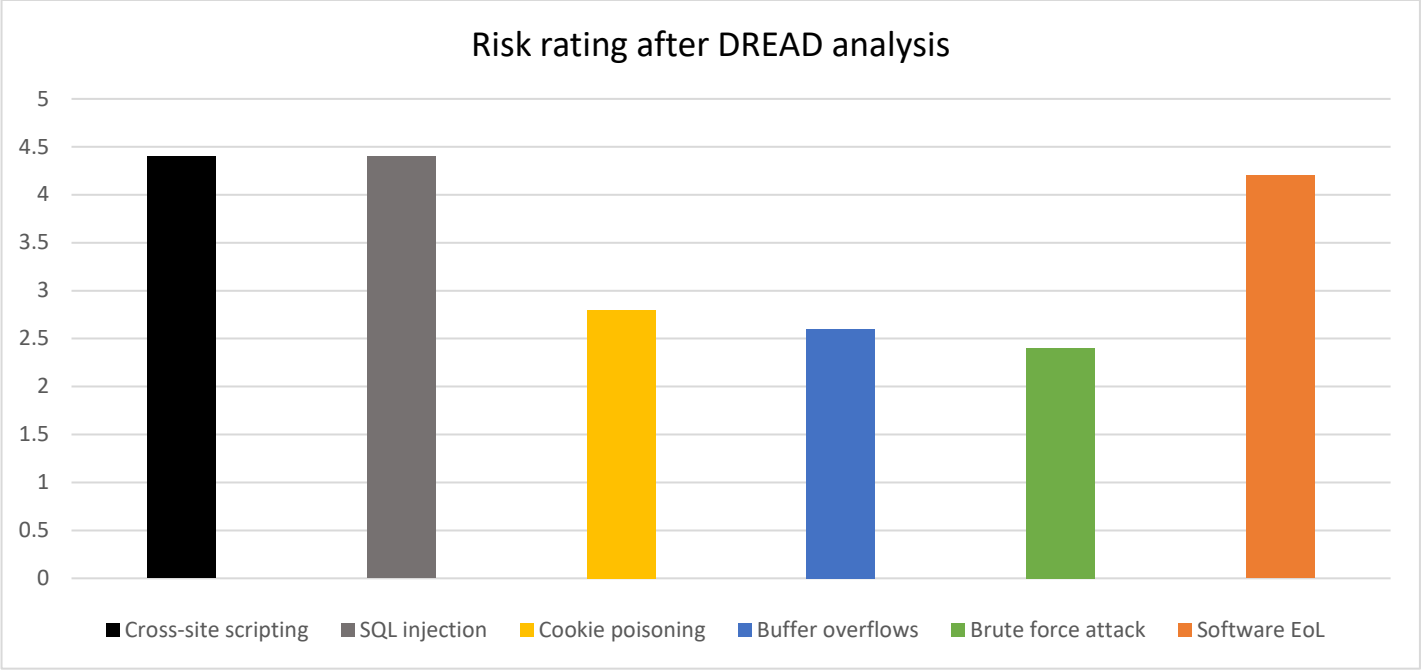
We set up the SugarCRM 6.5 version in a Virtual Machine to offer a comparison to see any significant discoveries. We conclude significantly more vulnerabilities were found in the localhost; however, both scans confirmed our assumptions of predicted vulnerabilities.





2.6 Tabulation of findings by severity using DREAD analysis



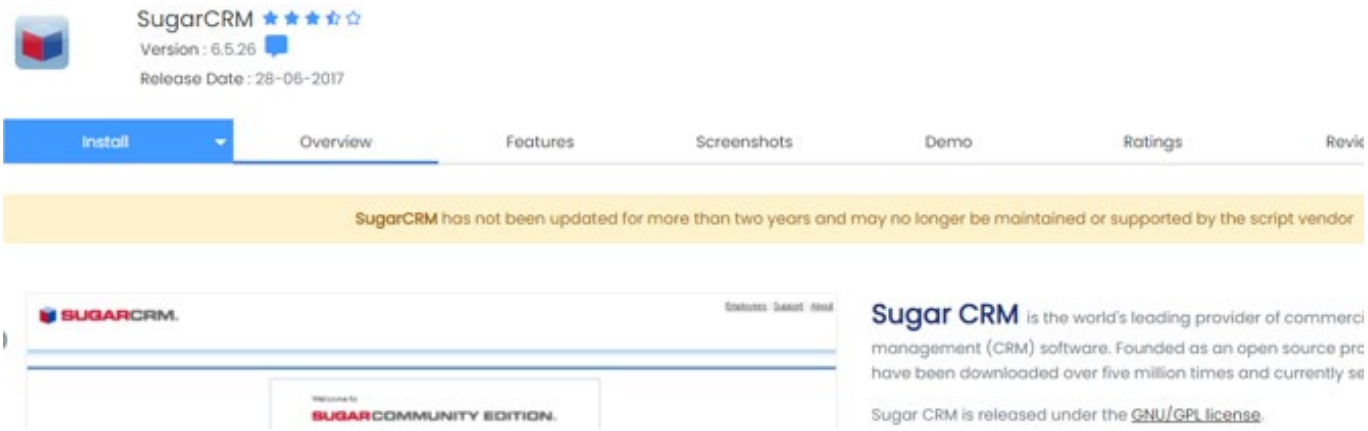


The data provided reconfirmed our assumptions, and we maintained our initial analysis when presented with the findings.

3. Discussion of vulnerabilities

The potential vulnerabilities detected on RedHat Enterprise Linux 7, Exim smtpd 4.94.2, MySQL 5.5.5-10.3.23-MariaDB-cll-lve (Appendix 6.2. Nmap) provided by the hosting company are out of control. This discussion will focus on the target WebApp "SugarCRM".

The "About" page of SugarCRM Community Edition has a login requirement using Softaculous CMS software. We assumed that the target site used the latest version 6.5.26 (Softaculous, N.D.).



Further omissions were discovered that DNS record does not contain Anti Spam/ Phishing policy, DOS/ DDOS attacks are excluded in the test due to the 'house rules'. The website was hosted by a genuine company using WAF (Appendix 6.5) and used four subdomains in two different locations; this guaranteed protection and availability of service. The tools did not detect unrestricted access and weak authentication vulnerabilities. We could not access the login to the website by using the default user with a common password list to bypass the login and crack the website credentials.

Five expected vulnerabilities found in SugarCRM Community Edition 6.5.26 (Vulmon Search, N.D.):

- CVE-2018-6308
- CVE-2017-14508
- CVE-2017-14509
- CVE-2017-14510
- CVE-2018-17784

3.1. SQL Injection

CVE-2017-14508 to 10

Common Vulnerability Scoring System (CVSS) Score: 6.5 - 4.3

Description:

Identified a remote file inclusion in the Connectors module that allows authorised users to use module=CallRest&url= query string to access system files remotely. Proper input validation was implemented as a solution. (Robin, 2017). Several sections in SugarCRM's Documents and Emails module have been discovered as vulnerable to SQL injection by an authorised user. If successfully exploited, a remote attacker might tamper with data in the database and gain elevated privileges over the affected application.

Proper SQL escaping has been added; No official solution to address this vulnerability for version 6.5.26 or below. (SugarCRM, 2017)

Detection:

The vulnerable function call is found in the source code of SugarCE as below.

Path: SugarCE-Full-6.5.26/modules/Connectors/controller.php

```
228     function action_CallRest() {
229         $this->view = 'ajax';
230
231         if(false === ($result=@file_get_contents($_REQUEST['url']))) {
232             echo '';
233         } else if(!empty($_REQUEST['xml'])){
234             $values = array();
235             $p = xml_parser_create();
236             xml_parse_into_struct($p, $result, $values);
237             xml_parser_free($p);
238             $json = getJSONobj();
239             echo $json->encode($values);
240         } else {
241             echo $result;
242         }
243     }
```

ZAP scan results were successfully manipulated five hits in the target site using the boolean conditions [Log In AND 1=1 --] and [Log In AND 1=2 --]. (Appendix 6.3. ZAP)

CVE-2018-6308

CVSS Score: 7.5

Description:

The DefenseCode ThunderScan application discovered multiple SQL injection vulnerabilities in SugarCRM Community Edition (DefenseCode, 2018). A remote attacker tamper with data in the database and elevated privileged access.

Due to End of Life (EoL) software, there is no official solution for version 6.5.26 or below. (Sugar Club, 2018)

Detection:

The vulnerable function call is found in the open-source file of SugarCE as below (Matthew, 2018).

Path: SugarCE-Full-6.5.26\modules\Campaigns\Tracker.php

```
59  if(empty($_REQUEST['track'])) {
60      $track = "";
61  } else {
62      $track = $_REQUEST['track'];
63  }
64  if(!empty($_REQUEST['identifier'])) {
65      $keys=log_campaign_activity($_REQUEST['identifier'],'link',true,$track);
66
67  }else{
68      //if this has no identifier, then this is a web/banner campaign
69      //pass in with id set to string 'BANNER'
70      $keys=log_campaign_activity('BANNER','link',true,$track);
71
72  }
```

CVE-2017-14508 to 10 required an authorised user for the successful exploitation.

CVE-2018-6308 required to call specific functions and steps for the successful exploitation, which penetration testing tool could not scan by default.

3.2. Cross Site Scripting (XSS)

CVE-2018-17784

Risk Score: 4.3

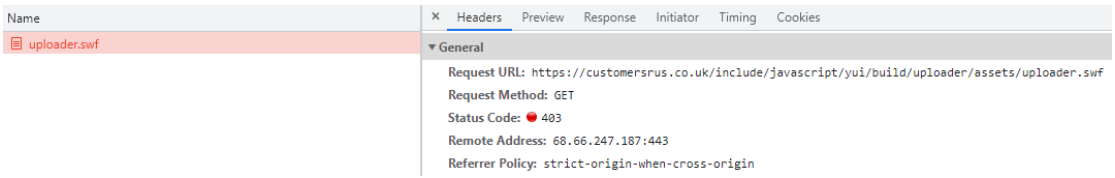
Description:

Multiple vulnerabilities in embedded YUI and FlashCanvas potentially allow an unauthenticated, remote attacker to launch an XSS attack on an affected machine (Exploit Database, 2018).

Detection:

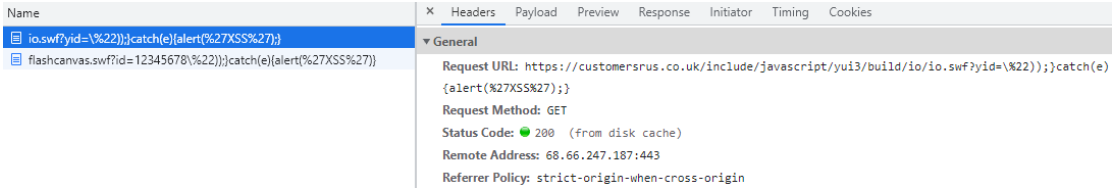
Except for the uploader.swf file, the vulnerable .swf file can be accessed on the target site below.

https://customersrus.co.uk/include/javascript/yui/build/uploader/assets/uploader.swf



https://customersrus.co.uk/include/javascript/yui3/build/io/io.swf

https://customersrus.co.uk/include/SugarCharts/Jit/FlashCanvas/flashcanvas.swf



ZAP scan results identified library jquery; version 1.7.1 is vulnerable.

Misconfigurations were found on Anti-CSRF tokens and HttpOnly Flag, which could mitigate XSS exploits. (Appendix 6.3. ZAP)

Vulnerable JS Library (2)

▶ GET https://customersrus.co.uk/cache/include/javascript/sugar_grp1_jquery.js?v=aGs7eX8Z4L90E3w2V1m7Qw

▶ GET https://customersrus.co.uk/cache/include/javascript/sugar_grp1_yui.js?v=aGs7eX8Z4L90E3w2V1m7Qw

Absence of Anti-CSRF Tokens (18)

Cookie No HttpOnly Flag (18)

Burp Suite scan results provided similar findings as ZAP. (Appendix 6.1. Burp Suite)

2. Vulnerable JavaScript dependency

2.1. http://customersrus.co.uk/cache/include/javascript/sugar_grp1_jquery.js
2.2. http://customersrus.co.uk/cache/include/javascript/sugar_grp1_yui.js

4. Cookie without HttpOnly flag set

4.1. http://customersrus.co.uk/
4.2. http://customersrus.co.uk/index.php

3.3. Buffer Overflow

Description:

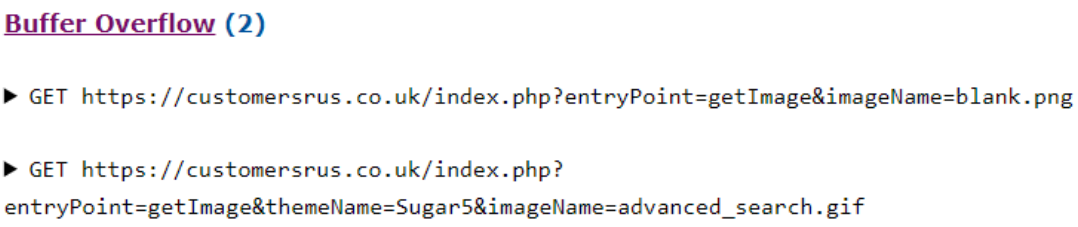
Buffer overflow errors overwrite process memory fragments that should never have been changed purposely or unexpectedly. (Common Weakness Enumeration, 2021).

Detection:

The vulnerable files can be accessed in the target site with the below links.

https://customersrus.co.uk/index.php?entryPoint=getImage&imageName=blank.png https://customersrus.co.uk/index.php?entryPoint=getImage&themeName=Sugar5&imageName=advanced_search.gif

ZAP scan results detected potential buffer overflow with overflow attack; the script closed the connection and threw a 500 Internal Server Error. (Appendix 6.3. ZAP)



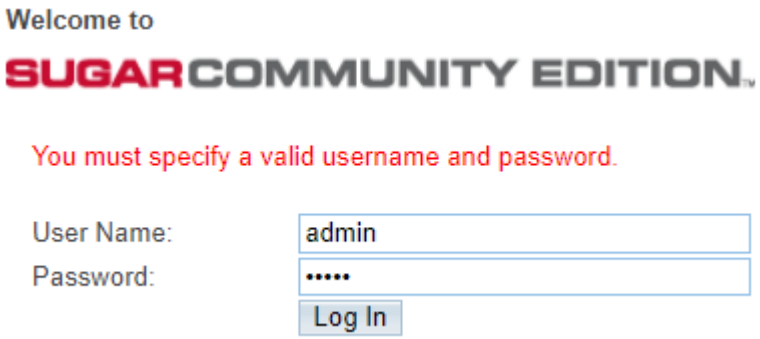
3.4. Account Security

Brute Force Attack

Description:

To attempt trial-and-error, guessing login details, encryption keys, or discovering hidden web pages using all conceivable combinations.

The login module is not built-in with an account that will lockout for unsuccessful login attempts in 6.5; it could install a third party's module for IP address blocking (Lion Solution, N.D.) or upgrade to a newer SugarCRM version (SugarCRM, 2020).



Detection:

Used Hydra to perform brute force attack on the testing server to proof of concept (POC) and verify command able to validate the correct password.

```
┌──(user㉿kali)-[~]
└─$ sudo hydra -l admin -p wrongpassword localhost https-post-form "/SugarCE/index.php:module=Users&action=Authenticate&return_module=Users&return_action=Login&cant_login=&login_module=&login_action=&login_record=&login_token=&login_oauth_token=&login_mobile=&user_name=^USER^&user_password=^PASS^&Login=Log+In:module=Users&action=Login"
Hydra v9.1 (c) 2020 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-02-06 01:04:22
[DATA] max 1 task per 1 server, overall 1 task, 1 login try (l:1/p:1), ~1 try per task
[DATA] attacking http-post-forms://localhost:443/SugarCE/index.php:module=Users&action=Authenticate&return_module=Users&return_action=Login&cant_login=&login_module=&login_action=&login_record=&login_token=&login_oauth_token=&login_mobile=&user_name=^USER^&user_password=^PASS^&Login=Log+In:module=Users&action=Login
1 of 1 target completed, 0 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-02-06 01:04:27

┌──(user㉿kali)-[~]
└─$ sudo hydra -l admin -p kali localhost https-post-form "/SugarCE/index.php:module=Users&action=Authenticate&return_module=Users&return_action=Login&cant_login=&login_module=&login_action=&login_record=&login_token=&login_oauth_token=&login_mobile=&user_name=^USER^&user_password=^PASS^&Login=Log+In:module=Users&action=Login"
Hydra v9.1 (c) 2020 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-02-06 01:04:50
[DATA] max 1 task per 1 server, overall 1 task, 1 login try (l:1/p:1), ~1 try per task
[DATA] attacking http-post-forms://localhost:443/SugarCE/index.php:module=Users&action=Authenticate&return_module=Users&return_action=Login&cant_login=&login_module=&login_action=&login_record=&login_token=&login_oauth_token=&login_mobile=&user_name=^USER^&user_password=^PASS^&Login=Log+In:module=Users&action=Login
[443][http-post-form] host: localhost   login: admin   password: kali
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-02-06 01:04:55
```

Perform brute force attack on target site with 3862 tries without blocked.

```
┌──(user㉿kali)-[~]
└─$ sudo hydra -l admin -P pwdlist.txt customersrus.co.uk https-post-form "/index.php:module=Users&action=Authenticate&return_module=Us
ers&return_action=Login&cant_login=&login_module=&login_action=&login_record=&login_token=&login_oauth_token=&login_mobile=&user_name=^
USER^&user_password=^PASS^&Login=Log+In:module=Users&action=Login" -F -S -v
Hydra v9.1 (c) 2020 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal
purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-02-06 01:08:04
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a previous session found, to prevent overw
riting, ./hydra.restore
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344400 login tries (l:1/p:14344400), ~896525 tries per task
[DATA] attacking http-post-forms://customersrus.co.uk:443/index.php:module=Users&action=Authenticate&return_module=Users&return_action=
Login&cant_login=&login_module=&login_action=&login_record=&login_token=&login_oauth_token=&login_mobile=&user_name=^USER^&user_passwor
d=^PASS^&Login=Log+In:module=Users&action=Login
[VERBOSE] Resolving addresses ... [VERBOSE] resolving done
[STATUS] 559.00 tries/min, 559 tries in 00:01h, 14343841 to do in 427:40h, 16 active
[STATUS] 553.00 tries/min, 1659 tries in 00:03h, 14342741 to do in 432:17h, 16 active
[STATUS] 551.71 tries/min, 3862 tries in 00:07h, 14340538 to do in 433:13h, 16 active
```

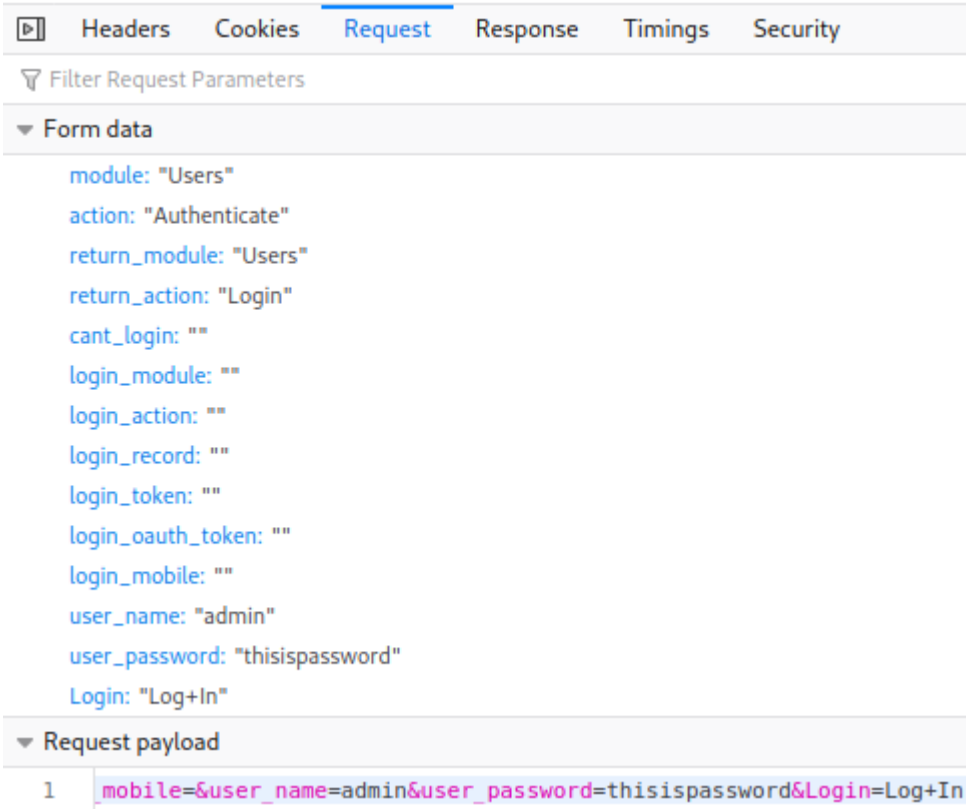
Cleartext submission

Description:

The site sends sensitive or critical information in cleartext through a communication connection that unauthorised parties can intercept. It could be configurated with HTTP method restrictions besides HTTPS redirection (SugarCRM, 2021).

Detection:

The POST request of password is in cleartext.



The HTTP could be accessed by the browser "Edge".

Burp Suite scan results identified the cleartext password (Appendix 6.1. Burp Suite).

1. Cleartext submission of password

4. Conclusions and recommendations

SugarCRM should implement the ITIL Service Value System (Nyhuis, 2020) to provide the best service management practices and deliver business value. The business impact should balance security measures against profit, providing a good service for clients minimising downtime, whether planned or unplanned. To prioritise business needs, we have allocated recommendations identifying short term recommendations (high priority) to long term recommendations (low priority).

4.1 Short term

1. Use up-to-date software: Prevent using EOL software (Sugar Club, 2018). Consider migrating up-to-date CRM software and patching web and application servers regularly to mitigate the impact of vulnerabilities.
2. Data encryption: Ensure that personal data and transactions are encrypted before transferring between the customer device and e-commerce website; it will help against a Man-In-The-Middle attack (Lokhande & Meshram, 2013). Force to use HTTPS with TLS encryption and restrict HTTP method, prevent cleartext submission for sensitive or critical information.
3. Account security: Limit unsuccessful login attempts by locking the account and blocking the IP address. Securing the server and admin panels by complex passwords and changing them frequently, using multi-factor authentication, and restricting access (inVerita, 2022).
4. Validate inputs: Implement the filter of a special character to verify that only properly constructed data enters the process, which could help to reduce the impact of code injection. (OWASP, N.D.)
5. SQL best practices: To prevent SQL injection attacks, Surkay (2021) suggested stopping dynamic SQL, storing sensitive credentials encrypted, limiting database permissions and privileges on the web application, and avoiding showing database errors directly to the user.

4.2 Medium term

1. Cloud service would prevent malicious code injection and DoS attacks. The beneficial inclusion of a Web Application Firewall (WAF) detected in our findings will help filter, monitor and block potentially malicious traffic and prevent unauthorised data loss.

   A Security Information and Event Management (SIEM) solution would assist in collecting logging data to detect threats and alerts. Outsourcing a managed security services provider (MSSP) can be tailored to budgets and specific needs, making them valued purchases as high costs can occur for insourcing a security operations centre (SOC).
2. CAPTCHAs would prevent dictionary attacks from automatic scripts. Low cost, however, can interrupt the experience and negatively impact their website usage for genuine users.

   HttpOnly attribute cookies and Anti-CSRF Tokens would mitigate Cross Site Scripting (XSS) attacks as special characters should be sanitised and disable external processing entities (Vamsi & Jain, 2021).
3. Spam/phishing policies to help staff and users through applying Domain-based Authentication, Reporting and Conformance (DMARC). Email threats are reduced by Send Policy Framework (SPF) in email headers for authentication. This can be challenging to deploy and time-consuming to implement but incurs little to no cost.

4.3 Long term

1. Staff and clients should be aware of the security policy and the potential threats; staff should not disclose their login information and password or share their credentials between them (Bader, 2021).
2. Lokhande & Meshram (2013) highlighted that developers and security professionals could fix the website by using a regular automated web vulnerability scanner to check the website to find out software vulnerabilities.
3. OWASP (2021) suggested remediation by separating data from commands using safe API and positive server-side validation; In addition, an intrusion detection system (IDS) could be incorporated for suspicious traffic on the client side.
4. Implementation of GDPR, such as getting valid consent of all customers and providing the opportunity to opt-out (Hosford, 2017).
5. Data protection must also be considered as hired vendors may handle sensitive data of SugarCRM and clients. Protecting data, customer privacy, and access rights need carefully selected processes and procedures.

5. References

Antaryami, A. (2021) Comparative analysis of Parrot, Kali Linux and Network Security Toolkit (NST). Available from: https://doi.org/10.7939/r3-pcre-7v35 [Accessed 2 February 2022].

Ashraf, M., Zahra, A. & Asif, M. et al. (2022) "Ethical Hacking Methodologies: A Comparative Analysis.", *2021 Mohammad Ali Jinnah University International Conference on Computing (MAJICC)*. IEEE. Available from: https://ieeexplore.ieee.org/abstract/document/9526243 [Accessed 11 February 2022].

Boehmer, W. (2009). *Cost-benefit trade-off analysis of an ISMS based on ISO 27001.* [Accessed 9 February 2022].

Chalvatzis, I., Karras, D. & Papademetriou, R. (2019) "Evaluation of Security Vulnerability Scanners for Small and Medium Enterprises Business Networks Resilience towards Risk Assessment." In 2019 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA). 2019. IEEE. pp. 52-58. Available from: https://ieeexplore.ieee.org/document/8873438 [Accessed 5 February 2022].

Common Weakness Enumeration. (2021) CWE-120: Buffer Copy without Checking Size of Input ('Classic Buffer Overflow'). *CWE List*. Available from: https://cwe.mitre.org/data/definitions/120.html [Accessed 9 February 2022].

Dahle, T. (2020) Large Scale Vulnerability Scanning. Master Degree, UNIVERSITY OF OSLO.

DefenseCode. (2018) SugarCRM Community Edition Multiple SQL Injection Vulnerabilities. *DefenseCode ThunderScan SAST Advisory.* Available from: https://www.defensecode.com/advisories/DC-2018-01-011_SugarCRM_Community_Edition_Advisory.pdf [Accessed 9 February 2022].

Exploit Database. (2018) SugarCRM 6.5.26 - Cross-Site Scripting. *Exploits*. Available from: https://www.exploit-db.com/exploits/45594 [Accessed 9 February 2022].

Firch, J. (2021) How To Prevent A Buffer Overflow Attack. Available from: https://purplesec.us/prevent-buffer-overflow-attack/ [Accessed 11 February 2022].

Hosford, P. (2017) Argos and Virgin Media among companies prosecuted for wrongly sending marketing texts, calls and emails. Available from: https://www.thejournal.ie/argos-marketing-emails-virgin-media-3499614-Jul2017/ [Accessed 23 January 2022].

inVerita. (2022) Top 10 E-commerce Security Threats and their Solutions. *Blog*. Available from: https://inveritasoft.com/blog/top-ecommerce-security-threats-and-their-solutions [Accessed 13 February 2021].

João, H. (2021) Web Application Penetration Test: Proposal for a generic Web Application Testing Methodology. Master Degree, ISCTE – IUL.

Kakarla, T., Mairaj, A. & Javaid, A. (2018) "A Real-World Password Cracking Demonstration Using Open Source Tools for Instructional Use.", *2018 IEEE International Conference on*

Electro/Information Technology (EIT). IEEE. Available from: https://ieeexplore.ieee.org/document/8500257 [Accessed 12 February 2022].

Kali Tools (2021). Available from: https://tools.kali.org/information-gathering/nikto [Accessed 5 February 2022].

Kali Tools (2021). Available from: https://www.kali.org/tools/hydra [Accessed 2 February 2022].

Karangle, N., Mishra, A. & Khan, D. (2019) "Comparison of Nikto and Uniscan for measuring URL vulnerability." In 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT). 2019. IEEE. Available from: https://ieeexplore.ieee.org/document/8944463 [Accessed 5 February 2022].

Kaur, G. & Kaur, N. (2017) Penetration Testing – Reconnaissance with NMAP Tool. International Journal of Advanced Research in Computer Science, 8 (3): 844-846. Available from: http://www.ijarcs.info/index.php/Ijarcs/article/view/3111 [Accessed 2 February 2022].

Lion Solution. (N.D.) CRM Defender. *Sugar Outfitters*. Available from: https://www.sugaroutfitters.com/addons/crm_defender [Accessed 9 February 2022].

Lokhande, P. & Meshram, B. (2013) E-Commerce Applications: Vulnerabilities, Attacks and Countermeasures. *International Journal of Advanced Research in Computer Engineering & Technology*. Available from: https://www.researchgate.net/publication/235697382_E-Commerce_Applications_Vulnerabilities_Attacks_and_Countermeasures [Accessed 9 February 2022].

Matthew, P. (2018) SugarCRM-Community-Edition. *Github*. Available from: https://github.com/matthewpoer/SugarCRM-Community-Edition/tree/master/OldFiles [Accessed 9 February 2022].

Nyhuis, M. (2020). What Is the ITIL Framework? Available: https://www.diligent.com/insights/compliance/what-is-the-itil-framework/ [Accessed 11 February 2022].

OWASP (2021) A03 Injection - OWASP Top 10:2021. Available from: https://owasp.org/Top10/A03_2021-Injection/ [Accessed 10 February 2022].

OWASP. (N.D.) Input Validation Cheat Sheet. *OWASP Cheat Sheet Series*. Available from: https://cheatsheetseries.owasp.org/cheatsheets/Input_Validation_Cheat_Sheet.html [Accessed 13 February 2021].

Raj, S. & Walia, N. (2020) "A Study on Metasploit Framework: A Pen-Testing Tool." In 2020 International Conference on Computational Performance Evaluation (ComPE). 2020. IEEE. Available from: https://ieeexplore.ieee.org/abstract/document/9200028 [Accessed 3 February 2022].

Robin, P. (2017) SugarCRM's Security Diet - Multiple Vulnerabilities. *Security*. Available from: https://blog.sonarsource.com/sugarcrm-security-diet-multiple-vulnerabilities/?redirect=rips [Accessed 9 February 2022].

Softaculous. (N.D.) SugarCRM. *ERP App*. Available from: https://www.softaculous.com/apps/erp/SugarCRM [Accessed 9 February 2022].

Sugar Club. (2018) Sugar Community Edition open source project ends. *Sugar News*. Available from: https://sugarclub.sugarcrm.com/engage/b/sugar-news/posts/sugar-community-edition-open-source-project-ends [Accessed 9 February 2022]

SugarCRM. (2017) sugarcrm-sa-2017-006. *Security*. Available from: https://support.sugarcrm.com/Resources/Security/sugarcrm-sa-2017-006/ [Accessed 9 February 2022].

SugarCRM. (2020) Modifying the Unsuccessful Login Lockout Period. *User Login Management.* Available from: https://support.sugarcrm.com/Knowledge_Base/Password_Management/Modifying_the_Unsuccessful_Login_Lockout_Period/ [Accessed 9 February 2022].

SugarCRM (2021) Web Server Configuration. *Sugar Developer Guide 11.0*. Available from: https://support.sugarcrm.com/Documentation/Sugar_Developer/Sugar_Developer_Guide_11.0/Security/Web_Server_Configuration/ [Accessed 13 February 2021].

Surkay, B. (2021) What is SQL Injection and How to Prevent It?. *Vulnerabilities*. Available from: https://www.pcidssguide.com/what-is-sql-injection-how-to-prevent-it/ [Accessed 13 February 2021].

Vamsi, P. & Jain, A. (2021) Practical Security Testing Of Electronic Commerce Web Applications. *International Journal of Advanced Networking and Applications*, 13 (01): 4861-4873. Available from: https://oaji.net/articles/2021/2698-1630317891.pdf [Accessed 10 February 2022].

Vulmon Search. (N.D.) SugarCRM 6.5.26 vulnerabilities and exploits. *Vulmon Search*. Available from: https://vulmon.com/searchpage?q=sugarcrm+6.5.26&sortby=byrelevance [Accessed 9 February 2022].

6. Appendices

6.1. Burp Suite

burp_report.html

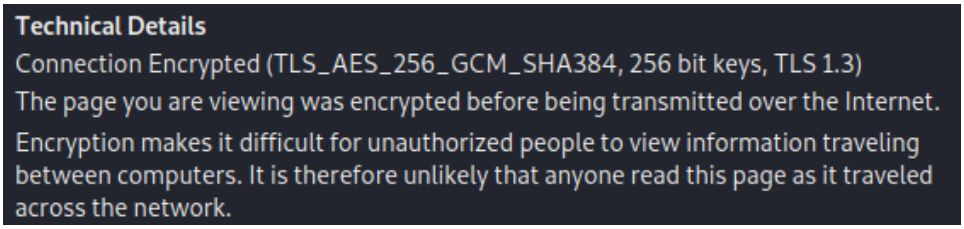
6.2. Nmap

nmap_vulners.log

6.3. ZAP

ZAP-Report-custo
mersrus.co.uk_slim

6.4. TLS



6.5. WAF

```
┌──$ nikto -host customersrus.co.uk
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          68.66.247.187
+ Target Hostname:    customersrus.co.uk
+ Target Port:        80
+ Start Time:         2022-01-29 09:54:06 (GMT-5)
---------------------------------------------------------------------------
+ Server: Apache
+ Server banner has changed from 'Apache' to 'imunify360-webshield/1.18' which may suggest a WAF, load balancer or proxy is in place
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type


+ ERROR: Error limit (20) reached for host, giving up. Last error: opening stream: can't connect (timeout): Operation now in progress
+ Scan terminated:  17 error(s) and 3 item(s) reported on remote host
+ End Time:           2022-01-29 10:03:41 (GMT-5) (575 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```