NOVEMBER 1, 2017

WEB APPLICATION PENETRATION TEST

Version 1.4

TechCorp
Bill Lumbergh | Chief Information Security Officer

RHINO SECURITY LABS

ASSESSMENT INFORMATION

Penetration Tester(s)

Hector Monsegur
hector.monsegur@rhinosecuritylabs.com
(888) 944-8679

Client Contact

Bill Lumberghh
Chief Information Security Officer
TechCorp

Engagement Manager

Christopher Lakin
Chris.lakin@rhinosecuritylabs.com
(888) 944-8679

Project Number

05-17-ITL-SE

Assessment Scope Summary

Engagement Timeframe

05/02/2017 – 05/21/2017

Engagement Scope

TechCorp Corporate Website
TechCorp Administrative Portal
TechCorp Customer Portal

Revision History

Date Change	Author	Notes
05-21-2017	Hector Monsegur	First Draft
05-22-2017	Christopher Lakin	Edits
05-22-2017	Benjamin Caudill	Edits
05-23-2017	Hector Monsegur	Final Draft

TABLE OF CONTENTS

ENGAGEMENT OVERVIEW

Rhino Security Labs specializes in web application penetration techniques and practices, identifying areas for improvement. At Rhino Security Labs we specialize in manual assessments that go beyond basic automated tests to identify real attack vectors that can be used against your application.

With decades of combined experience, Rhino Security Labs is at the forefront of application security and penetration testing. With a veteran team of subject matter experts, you can be sure every resource is an authority in their field.

Service Description

Extensive Application Assessment

With their growing complexity and demand, web applications have become the target of choice for hackers. Rhino Security Labs' Application Service offerings help protect your enterprise applications web services from a range of security threats.

Application security issues are not only the most common type of vulnerability, they're also growing in complexity. While the OWASP Top 10 is used as the standard for identifying application security flaws, that's just a start - many advanced vulnerabilities are not included in that list. Automated vulnerability scanners and penetration testers focused on OWASP will fall behind new threats, leaving the application exposed to unknown risks.

At Rhino Security Labs, we go far beyond the OWASP Top 10, continually pushing the boundaries of application security and detailing the way unique architectures can be abused – and how to fix them.

Manage Application Security Risk

Webservers are no longer just for marketing. Often the interface to an entire suite of technologies, web applications can tie into critical databases, vulnerable libraries, and plugins, XML and LDAP capabilities, underlying operating systems and client web browsers. It's never "just a website" anymore, making them even more difficult to protect.

Campaign Objectives

Vulnerability Identification

Rhino Security's consultants use the results of the automated scan, paired with their expert knowledge and experience, to finally conduct a manual security analysis of the client's applications. Our assessors attempt to exploit and gain remote unauthorized access to data and systems. The detailed results of both the vulnerability scan and the manual testing are shown in this report.

KEY PERSONNEL

Passionate and forward-thinking, our consultants bring decades of combined technical experience as top-tier researchers, penetration testers, application security experts, and more. Drawing from security experience in the US military, leading technology firms, defense contractors, and Fortune 100, we pride ourselves on both depth and breadth of information security experience.

Chris Lakin - *Cybersecurity Engagement Manager*

Chris Lakin has accumulated over eight years of project management and customer engagement experience across a multitude of industries. A proponent of constant iteration and improvement, his knowledge from time spent in business and marketing adds a valuable perspective to every cybersecurity engagement. Receiving a Masters of Science in Cybersecurity Engineering from the University of Washington, Mr. Lakin excels at connecting the technical with the goals of the business.

Hector Monsegur - *Director of Assessment Services*

Hector Monsegur brings a unique perspective from decade of offensive experience and a desire to make an impact in client security. In working with the US Government, Mr. Monsegur identified key vulnerabilities - and potential attacks - against major federal infrastructure including the US military and NASA. In his role as a security ressearcher at Rhino Security Labs, he has identified countless zeroday vulnerabilities and contributed to dozens of tools and exploits. In his leadership role, his unmatched technical experience is shared to both educate other operators and guide techincal research. Mr. Monsegur is a leading speaker for security organizations and conferences around the world.

Benjamin Caudill - *CEO and Founder*

Benjamin Caudill is an adept cybersecurity professional, researcher, and entrepreneur. A veteran of the defense and aerospace industry, Mr. Caudill led investigations into advanced cyberattacks, coordinating with federal intelligence communities on complex engagements. As Founder and CEO of Rhino Security Labs, Mr. Caudill has built the boutique security firm and turned it into a major player in the penetration testing market. In addition to his executive role, Mr. Caudill oversees company research and development, ensuring the continued development of key offensive technologies.

WEB APPLICATION PENETRATION TESTING METHODOLOGY

At Rhino Security Labs, our application penetration testing targets the entire range of vulnerabilities in your web application or API. Using the same techniques as sophisticated real-world attackers, we providing unique visibility into security risks automated tools often miss. To ensure high quality, repeatable engagements, our penetration testing methodology follows these steps:

1

Reconnaissance

This process begins with detailed scanning and research into the architecture and environment, with the performance of automated testing for known vulnerabilities. Manual exploitation of vulnerabilities follows, for the purpose of detecting security weaknesses in the application.

As with malicious hackers, each penetration test begins with information gathering. Collecting, parsing, and correlation information on the target is key to identifying vulnerabilities.

2

Vulnerability Detection

Once the target has been fully enumerated, Rhino Security Labs uses both vulnerability scanning tools and manual analysis to identify security flaws. With decades of experience and custom-built tools, our security engineers find weaknesses most automated scanners miss.

3

Attack and Post-Exploitation

At this stage of the assessment, our consultants review all previous data to identify and safely exploit identified application vulnerabilities. Once sensitive access has been obtained, the focus turns to escalation and movement to identify technical risk and total business impact.

During each phase of the compromise, we keep client stakeholders informed of testing progress, ensuring asset safety and stability.

4

Assessment Reporting

Once the engagement is complete, Rhino Security Labs delivers a detailed analysis and threat report, including remediation steps. Our consultants set an industry standard for clear and concise reports, prioritizing the highest risk vulnerabilities first. The assessment includes the following:

- Executive Summary
- Strategic Strengths and Weaknesses
- Identified Vulnerabilities and Risk Ratings
- Detailed Risk Remediation Steps
- Assets and Data Compromised During Assessment

5

Remediation (Optional)

As an optional addition to the standard assessment, Rhino Security Labs provides remediation retesting for all vulnerabilities listed in the report. At the conclusion of the remediation testing and request of the client, Rhino Security Labs will update the report with a new risk level determination and mark which vulnerabilities in the report were in fact remediated to warrant a new risk level.

EXECUTIVE SUMMARY OF FINDINGS

Rhino Security Labs conducted a Web Application Penetration Test for TechCorp. This test was performed to assess TechCorp's defensive posture and provide security assistance through proactively identifying vulnerabilities, validating their severity, and providing remediation steps.

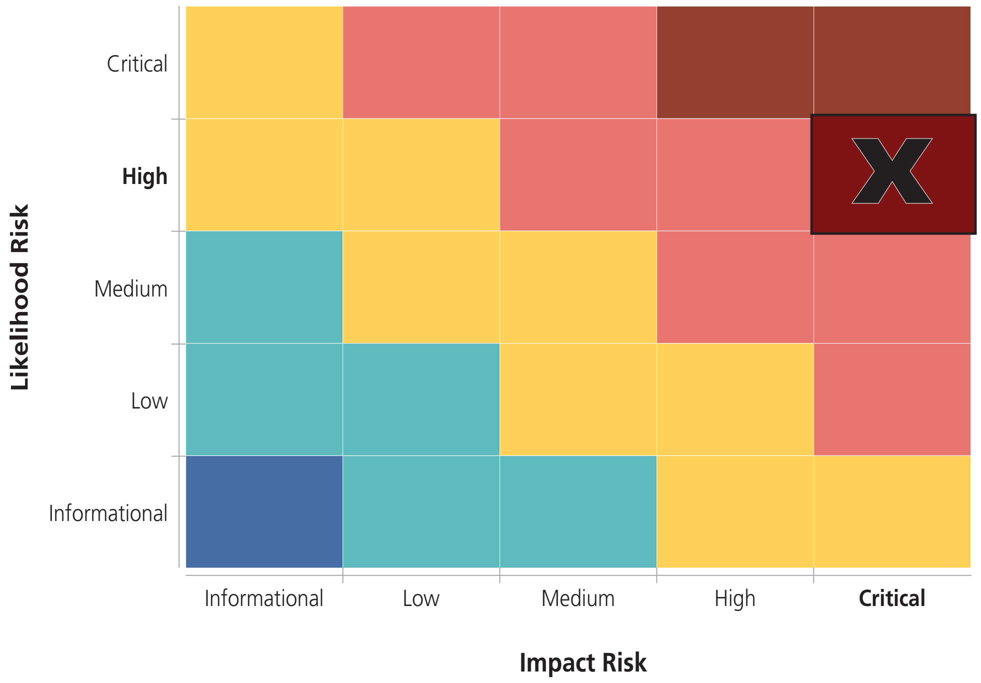
Rhino Security Labs reviewed the security of TechCorp's infrastructure and had determined a Critical risk of compromise from external attackers, as shown by the presence of the vulnerabilities detailed in this report.

The detailed findings and remediation recommendations for these assessments may be found later in the report.

WEB APPLICATION RISK RATING

Rhino Security Labs calculates Web Application risk based on Exploitation Likelihood (Ease of exploit) and Potential Impact (Technical Controls).

Overall Risk Rating: CRITICAL



Summary of Strengths

Rhino Security Labs acknowledged a number of security and technical controls that blocked attempts to carry out malicious actions. Understanding the strengths of the current environment can reinforce security best practices and provide strategy and direction toward a robust defensive posture. The following traits were identified as strengths in TechCorp's environment.

- Client implements proper Two-Factor Authentication for login forms.
- Web application is using basic encryption to help protect user data storage and in transit.

Summary of Weaknesses

Rhino Security Labs discovered and investigated many vulnerabilities during its assessments of TechCorp. We have categorized these vulnerabilities into general weaknesses across the current environment, and provide direction toward remediation for a more secure enterprise

- Poor sanitation of data from majority of user fields in application.
- Input fields are not protected against a variety of DDoS and brute forcing attacks.
- Vulnerable to cross site scripting attacks.
- Session tokens and cookies are poorly configured and managed insecurely.

Strategic Recommendations

Not all security weaknesses are technical in nature, nor can they all be remediated by security personnel. Companies often focus on the root security issues and resolve them at their core. These strategic steps are changes to the operational policy of the organization. Rhino Security Labs recommends the following strategic steps for improving the company's security.

1. Ensure proper development and repository practices by in-house and 3rd party developers
2. Remove all sensitive pages and directories from publicly accessible servers; Require authentication on such pages as necessary.
3. Remove legacy servers, subdomains, pages, and other web resources when no longer in use
4. Sanitize all user inputs before processing on webserver or other internal resources
5. Enhance security defenses with additional detection and response capabilities, such as a SIEM

SCOPING AND RULES OF ENGAGEMENT

While real attackers have no limits on web application penetration engagements, we do not engage in penetration testing activities that threaten our ethics and personal privacy.

Constraints

In addition, the following limitations were put into place:

- Vulnerabilities which would cause outages or interrupt the client's environment were noted but not validated.
- Penetration testing was limited to the agreed upon engagement period, scope, and other additional boundaries set in the contract and service agreement.

Scope of Service

The predetermined scope for Rhino Security Labs to carry out the Web Application Penetration Test was:

TechCorp Main Corporate Website

Domain Name

www.TechCorp.co

Description

The main corporate website is mostly used for marketing and business development purposes. Potential leads and clients can use the corporate website to learn more about TechCorp and our service offerings. Leads can also submit a request form to receive a pricing quote for our services.

Sensitive Assets and Processes

The most sensitive area is the request a quote form. This webform requires the user to upload their own files which are stored directly on our servers. If an attacker was able to upload a malicious file, that could potentially affect our operations and marketing strategy.

TechCorp Administrative Website Portal

Domain Name

admin.TechCorp.co

Description

The TechCorp Administrative Website Portal is restricted to only the highest level of privleged users. This administrative portal is where adjustments to TPS reports and confidential client information is stored. Administrators are constantly in this portal and it is vital to business operations.

Sensitive Assets and Processes

Within the portal, our entire client database is available along with sensitive information about individuals that work within the organization. If an attacker gained access to this system, it could shut down operations at TechCorp for good. Therefore, special attention should be given to the Admin Portal.

TechCorp Customer Website Portal

Domain Name

app.TechCorp.co

Description

The Customer Website Portal is mainly used by our clients to download their latest reports. The website portal is also used by our clients to process payments for TechCorp services.

Sensitive Assets and Processes

The reports are sensitive as they contain private information about each client. Ensuring clients are not able to access other customer data is very important. Also, since TechCorp process payments through this portal, we are regulated by PCI compliance.

SUMMARY VULNERABILITY OVERVIEW

Rhino Security Labs performed a Web Application Penetration Test for TechCorp on 07-01-2017 - 07-31-2017. This assessment utilized both commercial and proprietary tools for the initial mapping and reconnaissance of the site, as well as custom tools and scripts for unique vulnerabilities.

During the manual analysis, assessors attempted to leverage discovered vulnerabilities and test for key security flaws, including those listed in the OWASP Top 10 Vulnerabilities list. The following vulnerabilities were determined to be of highest risk, based on several factors including asset criticality, threat likelihood, and vulnerability severity.

Vulnerability Risk Definition and Criteria

The risk ratings assigned to each vulnerability are determined by averaging several aspects of the exploit and the environment, including reputation, difficulty, and criticality.

CRITICAL	Critical vulnerabilities pose a very high threat to a company's data, and should be fixed on a top-priority basis. They can allow a hacker to completely compromise the environment or cause other serious impacts to the security of the application
HIGH	High severity vulnerabilities should be considered a top priority regarding mitigation. These are the most severe issues and generally, cause an immediate security concern to the enterprise
MEDIUM	Medium severity vulnerabilities are a lower priority, but should still be remediated promptly. These are moderate exploits that have less of an impact on the environment.
LOW	Low severity vulnerabilities are real but trivially impactful to the environment. These should only be remediated after the HIGH and MEDIUM vulnerabilities are resolved.
INFORMATIONAL	Informational vulnerabilities have no impact as such to the environment by themselves. However, they might provide an attacker with information to exploit other vulnerabilities.

VULNERABILITIES BY RISK RATING

The following vulnerabilities were found within each risk level. It is important to know that total vulnerabilities is not a factor in determining risk level. Risk level is depends upon the severity of the vulnerabilities found.

4	7	4	1	4
Critical	**High**	**Medium**	**Low**	**Informational**

Total Vulnerabilities: 20

Vulnerability ID - Name	**Risk Level**
C1 - Cloudflare WAF Bypass Vulnerability **Remediation** Remove the subdomain identifying the server's direct IP address.	**CRITICAL**
C2 - SQL Injection **Remediation** Use parameterized queries (also known as prepared statements) for all database access	**CRITICAL**
C3 - Sensitive GIT Configuration File Leakage **Remediation** Remove the .git directory from being publicly accessible	**CRITICAL**
C4 - Sensitive Internal Webpages Publically Accessible **Remediation** Remove sensitive files from public servers, or authenticate users if necessary.	**CRITICAL**
H1 - Reflected Cross Site Scripting (XSS) **Remediation** Validate all user-submitted parameters to prevent injection of client-side code.	**HIGH**
H2 - API Documentation Leakage Vulnerability **Remediation** Remove sensitive files and directories from public servers.	**HIGH**

H3 - Hidden Web Functions Identified HIGH

Remediation
Remove sensitive files and directories from public servers.

H4 - No Bruteforce Protection on Authentication Forms HIGH

Remediation
Implement a captcha after five incorrect passwords

H5 - OS Command Injection HIGH

Remediation
Implement filtering to remove malicious characters from any user input fields.

H6 - Server-Side Request Forgery (SSRF) HIGH

Remediation
Validate input - use white-list to check allowed URLs

H7 - Username Enumeration through Password Reset Function HIGH

Remediation
Change password recovery messages to be generic, to not indicate validity status.

M1 - Client Cookies Detected Without HTTPOnly Flag Set MEDIUM

Remediation
Include the HTTPOnly flag by including it as an attribute in the relevant Set-cookie directive.

M2 - Content Spoofing MEDIUM

Remediation
Validate all parameters to prevent client-side injection attacks.

M3 - Cross Site Flashing (XSF) MEDIUM

Remediation
Validate all parameters to prevent XSS, XSF, and similar injection attacks for client-side code.

M4 - SSL Cookie Without Secure Flag Set MEDIUM

Remediation
Ensure all cookies issued have the secure flag set, preventing plaintext transmission (HTTP).

L1 - Cookie Scoped to Parent Domain LOW

Remediation
Remove the explicit domain attribute from your Set-cookie directive.

I1 - Cacheable HTTPS Response

INFORMATIONAL

Remediation

Configure the web server to prevent HTTPS caching.

I2 - Cross Domain Script Include

INFORMATIONAL

Remediation

Ensure no scripts included in the site are loaded from untrusted domains

I3 - HTML Does Not Specify Charset

INFORMATIONAL

Remediation

Include the recognized character set in the application.

I4 - User Agent Dependent HTTP Response

INFORMATIONAL

Remediation

With an informational rating, no changes are necessary.

VULNERABILITY FINDINGS

The vulnerabilities below were identified and verified by Rhino Security Labs during the process of this Web Application Penetration Test for TechCorp. Retesting should be planned following the remediation of these vulnerabilities.

C1 Cloudflare WAF Bypass Vulnerability

Risk Rating: CRITICAL

Exploitation Likelihood: **Critical**
Potential Impact: **Critical**

Description

Cloudflare, a popular Web Application Firewall (WAF) and DDoS protection service, was found protecting the client web application. By enumerating client subdomains, a subdomain was identified (dcconnect.techcorp.com) pointing to the direct IP address of the application. This allows an attacker to bypass the Cloudflare protections in place and attack the site directly.

Affected Domains and URLS

- client.TechCorp.com
- www.techcorp.com
- dev.techcorp.com

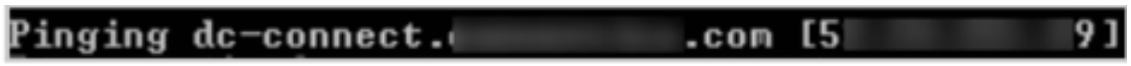
Remediation

Remove the subdomain identifying the server's direct IP address and audit other DNS records for Cloudflare-bypassing subdomains.

Testing Process

The identified issue was found by enumerating subdomains and identifying the direct IP of the protected web server.

Pictured below is a ping revealing the true IP of the affected server.



```
Pinging dc-connect.[redacted].com [5[redacted]9]
```

C2 SQL Injection

CWE-89

Risk Rating: CRITICAL

Exploitation Likelihood: **Critical**
Potential Impact: **Critical**

Description

SQL injection vulnerabilities arise when user-controllable data is incorporated into database SQL queries in an unsafe manner. An attacker can supply crafted input to break out of the data context in which their input appears and interfere with the structure of the surrounding query.

A successful SQL injection exploit can read sensitive data from the database, modify database data, execute administration operations on the database, and in some cases issue commands to the operating system

Affected Domains and URLS

client.techcorp.com

login.php

www.login.techcorp.com

login.php
forgotpassword.php

Remediation

Use parameterized queries (also known as prepared statements) for all database queries. Parameterized queries force the developer first to define all the SQL code, and then pass in each parameter to the query later.

Prepared statements ensure that an attacker is not able to change the intent of a query, even if SQL commands are inserted by an attacker.

Testing Process

This vulnerability was detected through automated scanning and confirmed by using SQLmap and other tools to exploit any potential vulnerabilities.

C3 Sensitive GIT Configuration File Leakage

Risk Rating: CRITICAL

Exploitation Likelihood: **Critical**
Potential Impact: **Critical**

Description

Git is a popular version control system designed to simplify the coordination of team-based projects. Git files, in relation to a website, can be publicly accessible if incorrectly configured. In this instance, the ".git" directory was found on the web-server which allowed public access to the configuration file. This file was found to contain cleartext credentials as well as the IP of a GIT Repo.

Affected Domains and URLS

login.techcorp.com

/.git/config

www.techcorp.com

/.git/config

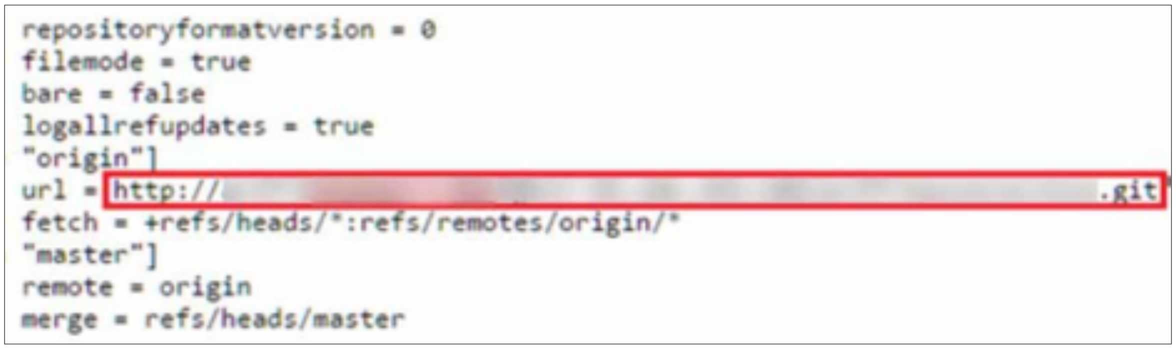
Remediation

Remove the .git directory from being publicly accessible by adding a simple .htaccess file at the root of the affected web server(s). It should contain one line, "RedirectMatch 404 /\.git" and will also hide many additional Git files that may be web-accessible.

Testing Process

This vulnerability was leaked by scanning the IP address of the web application.

Below is the result of an automated scan that provided the .git repo in question.



```
repositoryformatversion = 0
filemode = true
bare = false
logallrefupdates = true
"origin"]
url = http://[redacted].git
fetch = +refs/heads/*:refs/remotes/origin/*
"master"]
remote = origin
merge = refs/heads/master
```

C4 Sensitive Internal Webpages Publically Accessible

CWE-200

Risk Rating: CRITICAL

Exploitation Likelihood: **High**
Potential Impact: **Critical**

Description
Sensitive internal pages were identified as being publicly accessible. These files were found to contain highly sensitive information about the organization and its clients.

Affected Domains and URLS

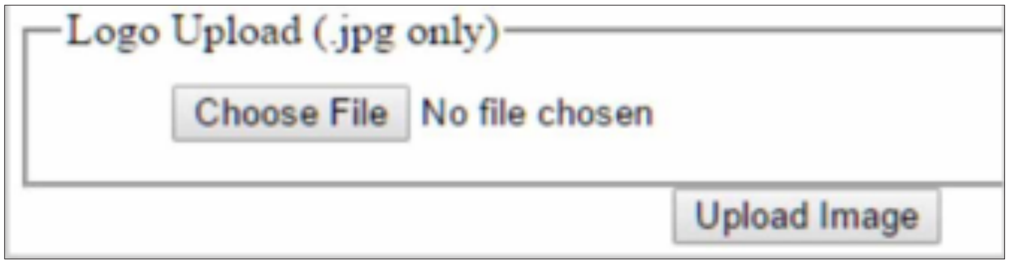
client.techcorp.com

/login

Remediation
Remove sensitive files and directories from the public web server. If necessary, restrict such areas to allowed users with authentication.

Testing Process
This vulnerability was discovered by automated scanning and manually verified by the assessor.

The following image shows where users can take advantage of arbitrary file upload functionality



H1 Reflected Cross Site Scripting (XSS)

Risk Rating: HIGH

Exploitation Likelihood: **High**
Potential Impact: **Critical**

Description

Reflected cross-site scripting vulnerabilities arise when data is copied from a request and echoed into the application's immediate response in an unsafe way. An attacker can use the vulnerability to construct a request which, if issued by another application user, will cause JavaScript code supplied by the attacker to execute within the user's browser in the context of that user's session with the application. The attacker-supplied code can perform a wide variety of actions, such as stealing the victim's session token or login credentials, performing arbitrary actions on the victim's behalf, and logging their keystrokes

Affected Domains and URLS

dev.techcorp.com

/update/crm.php
/login/php
/forgotpassword.php
/update/user.php
/update/connections.php
/update/systems.php

Remediation

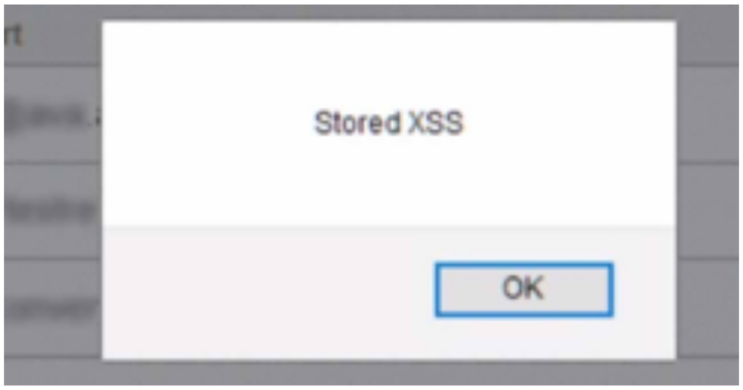
When user-controllable inputs are copied into application responses, cross-site scripting attacks can be prevented using two layers of defense:

1. Input should be validated as strictly as possible on arrival, given the kind of content which it is expected to contain. For example, personal names should consist of alphabetical and alphanumeric characters, and be relatively short; a year of birth should consist of exactly four numerals; email addresses should match a well-defined regular expression. Input which fails the validation should be rejected, not sanitized.
2. User input should be HTML-encoded at any point where it is copied into application responses. All HTML metacharacters, including < >

Testing Process

This vulnerability was detected by automated scanning and manually verified with a proof of concept exploit.

Below is a Javascript popup, verifying the reflected XSS vulnerability.



H2 API Documentation Leakage Vulnerability

Risk Rating: HIGH

Exploitation Likelihood: **Critical**
Potential Impact: **High**

Description

Documentation for the API was leaked and can be utilized to understand better – and exploit – the API.

Affected Domains and URLS

login.techcorp.com

/documentation/api/views/intro.php

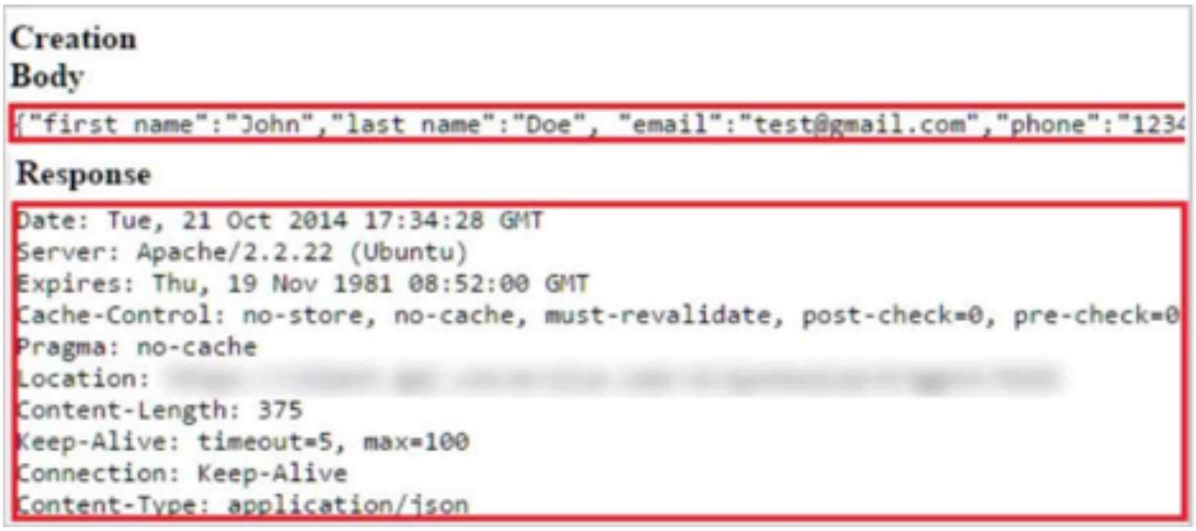
Remediation

Remove sensitive files and directories from public servers.

Testing Process

This vulnerability was discovered after downloading the Git Index file and identifying the path for API documentation.

Below is a shot from the Git index file, revealing the location of the API documentation.



Creation
Body

```
{"first name":"John","last name":"Doe", "email":"test@gmail.com","phone":"1234
```

Response

```
Date: Tue, 21 Oct 2014 17:34:28 GMT
Server: Apache/2.2.22 (Ubuntu)
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
Location: [redacted]
Content-Length: 375
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: application/json
```

H3 Hidden Web Functions Identified

Risk Rating: HIGH

Exploitation Likelihood: **High**
Potential Impact: **High**

Description

Multiple web functions were identified which could be abused. In this case, we found functionality for uploading arbitrary files to the webserver ('logo upload') and similar functionality.

Affected Domains and URLS

dev.techcorp.com

/contents/dev/logoupload.php
/contents/dev/serverupload.php
/crons/inventory_uploads.php
/crons/run_maintenance.php

Remediation

Remove sensitive files and directories from the public web server. If necessary, restrict such areas to allowed users with authentication.

Testing Process

This vulnerability was discovered after downloading the Git Index file and extracting the pathnames from it, which included the path for the hidden web function.

H4 No Bruteforce Protection on Authentication Forms

Risk Rating: HIGH

Exploitation Likelihood: **Medium**
Potential Impact: **Critical**

Description

There is no account lockout or captcha after entering a series of wrong passwords. This allows attackers to try a large number of possible passwords against a victim's account, attempting to guess the correct password. Due to the simplicity of many user's passwords, this is often successful and allows for the full takeover of a victim user's account.

Affected Domains and URLS

login.techcorp.com

/users

Remediation

This issue can be remediated by requiring user-interaction when multiple incorrect passwords are entered.

The simplest option to fix this vulnerability is to implement a captcha after a user enters a series of wrong passwords. This threshold is recommended to be between 3-10, depending on the sensitivity of the account and associated data.

If a captcha isn't an option, account lockouts are also sufficient remediation, requiring a user to verify their account through the associated email address. Applications with mobile platform support are recommended to have a slightly higher threshold to account for the small keyboards and ease of mistyping a password

Testing Process

To verify, ten wrong passwords were entered into a test account, followed by the correct password, which was accepted. There was no lockout or other security control enabled, and login was completed successfully.

H5 OS Command Injection

CWE-78

Risk Rating: HIGH

Exploitation Likelihood: **High**
Potential Impact: **Critical**

Description

OS command injection vulnerabilities allow an attacker to execute arbitrary shell commands via user input in the web application. Command injection attacks are possible when an application passes unsafe user-supplied data (forms, cookies, HTTP headers, etc.) to a system shell.

In this attack, the attacker-supplied operating system commands are usually executed with the privileges of the vulnerable application. Command injection attacks are possible largely due to insufficient input validation.

Affected Domains and URLS

dev.TechCorp.com

/database-users

Remediation

All user input should be validated as strictly as possible, and malicious characters blacklisted. Ideally, user input could be whitelisted to only known good values. If whitelisting isn't feasible, then regular testing should be done and character/phrase blacklists updated.

Testing Process

This was discovered by sending shell commands which could be identified from outside the system, such as a ping to an external, controlled server. If an ICMP request is seen from the targeted system, we can validate the vulnerability and confirm the security risk.

H6 Server-Side Request Forgery (SSRF)

Risk Rating: HIGH

Exploitation Likelihood: **Medium**
Potential Impact: **High**

Description

Server-Side Request Forgery (SSRF) is a vulnerability that appears when an attacker can create requests from the vulnerable server. Using Server-Side Request Forgery attacks, it's possible to:

- Scan and attack systems from the internal network that are not normally accessible
- Enumerate and attack services that are running on these hosts
- Exploit host-based authentication services

Affected Domains and URLS

login.techcorp.com

/
/users

Remediation

Validate input - use white-list to check allowed URLs/parameters and correctly escape user input when constructing the remote server URL

Testing Process

Once the HTTP Request was sent, a DNS lookup was made for the Host which was injected into the docstring. This confirms that a user can modify the docstring to force the server to perform certain requests or actions.

H7 Username Enumeration through Password Reset Function

Risk Rating: HIGH

Exploitation Likelihood: **Medium**
Potential Impact: **High**

Description

This vulnerability is discovered by submitting both known valid and invalid usernames/emails to the password reset functionality of the application, and comparing application responses. If the invalid user is identified through the app error message, an attacker can enumerate a list of emails used by brute-forcing possible emails and noting the differences in responses.

Affected Domains and URLS

client.techcorp.com

/home

Remediation

Change password recovery messages to be generic, to not indicate if the email was sent (confirming the user) or there was an error (invalid email).

Testing Process

This vulnerability was confirmed by providing an incorrect username and requesting for password reset. HTTP Response 500 can be reliably used to enumerate usernames and which Email IDs have accounts on the application.

M1 Client Cookies Detected Without HTTPOnly Flag Set

CWE-614

Risk Rating: MEDIUM

Exploitation Likelihood: **Medium**
Potential Impact: **High**

Description

A cookie was found without the HTTPOnly flag set, making the cookies more vulnerable to XSS attacks.

If a browser that supports HttpOnly detects a cookie containing the HttpOnly flag, and client side script code attempts to read the cookie, the browser returns an empty string as a result. This causes the attack to fail by preventing the malicious (usually XSS) code from sending the data to an attacker's website.

Affected Domains and URLS

login.techcorp.com

/

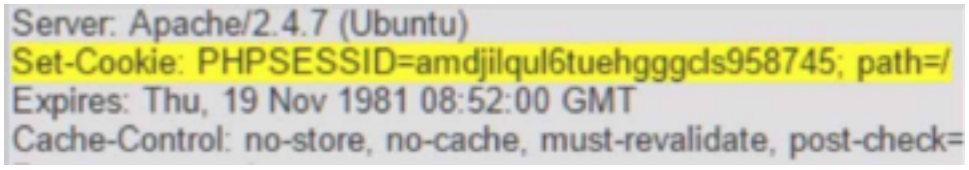
Remediation

Include the HTTPOnly flag by including it as an attribute in the relevant Set-cookie directive.

Testing Process

This vulnerability was detected by automated scanning and manually verified by the assessor. See missing HTTPOnly Flag below

Below is the session cookie being used without an HTTPOnly flag.



```
Server: Apache/2.4.7 (Ubuntu)
Set-Cookie: PHPSESSID=amdjilqul6tuehgggcls958745; path=/
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=
```

M2 Content Spoofing

Risk Rating: MEDIUM

Exploitation Likelihood: **Medium**
Potential Impact: **Medium**

Description

Content spoofing - also known as digital defacement - is an attack targeting a user using some injection vulnerability within a web application.

When an application does not properly handle user-supplied data, an attacker can supply content to a web application, typically via a parameter value, that is reflected back to the user. This presents the user with a modified page under the context of the trusted domain.

Affected Domains and URLS

techcorp.com

/home

Remediation

Follow the whitelist philosophy with data validation and only allow the user to input data of the expected type and discard data outside that range. When you have a range of data that can be entered, make sure it's properly sanitized.

Testing Process

This vulnerability was discovered by sending malicious client-side code in a parameter and confirming it's modification of the page within the browser.

M3 Cross Site Flashing (XSF)

Risk Rating: MEDIUM

Exploitation Likelihood: **High**
Potential Impact: **High**

Description

Cross-Site Scripting (XSS) attacks are client-side code injection attacks where an attacker can execute malicious code into a users browser using a vulnerable web application.

Similarly, Cross-Site Flashing (XSF) uses vulnerable ActionScript parameters to inject code into the client's browser using a malicious link.

Affected Domains and URLS

login.techcorp.com

/

Remediation

Follow the whitelist philosophy with data validation and only allow the user to input data of the expected type and discard data outside that range. When you have a range of data that can be entered, make sure it's properly sanitized.

Testing Process

This vulnerability was discovered by sending malicious client-side code in a parameter and confirming its execution within the browser by clicking the link.

M4 SSL Cookie Without Secure Flag Set

Risk Rating: MEDIUM

Exploitation Likelihood: **Medium**
Potential Impact: **High**

Description

Sensitive cookies were issued by the application which does not have the Secure flag set. The purpose of the Secure flag is to prevent cookies from being transmitted unencrypted (via HTTP) and able to intercepted by unauthorized parties.

Affected Domains and URLS

dev.techcorp.com

/portal

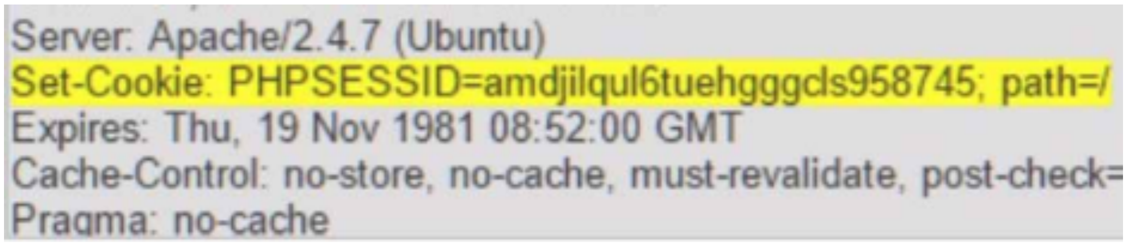
Remediation

Ensure sensitive cookies issued have the secure flag set, preventing plaintext transmission (HTTP).

Testing Process

This vulnerability was detected by automated scanning and manually verified by the assessor.

Below is a screenshot of the cookie, missing a secure flag.



```
Server: Apache/2.4.7 (Ubuntu)
Set-Cookie: PHPSESSID=amdjilqul6tuehgggcls958745; path=/
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=
Pragma: no-cache
```

L1 Cookie Scoped to Parent Domain

CWE-16

Risk Rating: LOW

Exploitation Likelihood: **Low**
Potential Impact: **High**

Description

A cookie's domain attribute determines which domains can access the cookie. If the cookie contains sensitive data (such as a session token), then this data may be accessible by less trusted or less secure applications residing on other subdomains, leading to a security compromise.

dev.techcorp.com

/database-login
/database-users

Remediation

If possible, remove the explicit domain attribute from your Set-cookie directive. This will give cookies the scope of the issuing domain and all subdomains. If you'd like the cookie available to the parent domain, thoroughly consider the security implications of having a larger cookie scope.

Testing Process

This was detected by automated scanning and confirmed manually by the assessor.

I1 Cacheable HTTPS Response

CWE-444

Risk Rating: INFORMATIONAL

Exploitation Likelihood: **Informational**
Potential Impact: **Informational**

Description

HTTPS responses (which are encrypted) were allowed to be cached and stored locally, allowing for the recovery of sensitive information on client systems.

dev.techcorp.com

/

Remediation

The application should return caching directives instructing browsers not to store local copies of any sensitive data. Often, this can be achieved by configuring the web server to prevent caching for relevant paths within the web root.

Ideally, the web server should return the following HTTP headers in all responses containing sensitive content: Cache-control: no-store Pragma: no-cache.

Testing Process

This vulnerability was detected by automated scanning and confirmed manually by the assessor.

I2 Cross Domain Script Include

Risk Rating: INFORMATIONAL

Exploitation Likelihood: **Informational**
Potential Impact: **Medium**

Description

When scripts are loaded from another domain, they have the permissions and access of the current domain, such as access to cookies. If malicious scripts are loaded from an untrusted domain, they can compromise the security of both the site and users.

login.techcorp.com

/users

Remediation

Ensure no scripts included in the site are loaded from untrusted domains.

Testing Process

This vulnerability was detected by automated scanning and manually verified by the assessor.

13 HTML Does Not Specify Charset

CWE-116

Risk Rating: INFORMATIONAL

Exploitation Likelihood: **Informational**
Potential Impact: **Informational**

Description

If a web response states that it contains HTML content but does not specify a character set, then the browser may analyze the HTML and attempt to determine which character set it appears to be using. Even if the majority of the HTML employs a standard character set such as UTF-8, the presence of non-standard characters anywhere in the response may cause the browser to interpret the content using a different character set.

This can have unexpected results and can lead to cross-site scripting vulnerabilities in which non-standard encodings like UTF-7 can be used to bypass the application's defensive filters. In most cases, the absence of a charset directive does not constitute a security flaw, particularly if the response contains static content.

techcorp.com

/
/users
/portal

Remediation

To reduce attack surface in the web application, the application should include a directive specifying a standard recognized character set, such as charset=ISO-8859-1.

Testing Process

This vulnerability was detected through automated scanning and manually verified by the assessor.

I4 User Agent Dependent HTTP Response

CWE-113

Risk Rating: INFORMATIONAL

Exploitation Likelihood: **Informational**
Potential Impact: **Informational**

Description

The application's responses appear to depend systematically on the value of the User-Agent header in requests – often a result of customizing pages for mobile users or architecture types. This behavior does not itself constitute a security vulnerability, but may point towards additional attack surface within the application that may contain vulnerabilities.

login.techcorp.com

/portal

Remediation

As user-agent dependent responses are not necessarily security vulnerabilities (and are often necessary for functionality), no explicit changes are necessary.

Testing Process

This issue was detected by automated scanning and manually confirmed by the assessor.

APPENDIX A: TOOLS & SCRIPTS

The software and tools used for security analysis are constantly evolving and changing. To stay at the forefront of industry trends, Rhino Security Labs regularly updates and integrates new tools into its social engineering methodology. Below is the toolset our consultants use during a social engineering assessment.

Enterprise and Open Source Tools

Nmap

Nmap is a powerful network security scanning application that uses carefully crafted packets to probe target networks and discover exposed open ports, services, and other host details, such as operating system type.

Nessus

Nessus is a proprietary vulnerability scanner that specializes in delivering comprehensive mappings of target system vulnerabilities, including web and network vulnerabilities, misconfigurations, weak passwords and even compliance problems, such as with HIPAA and PCI.

Metasploit Pro

An open-core commercial Metasploit edition for penetration testers. Metasploit Pro includes modules for tracking emails that have been opened, if and when their links have been opened, if their attachments were downloaded and ran, if a user submitted their credentials to a malicious domain and more.

Burpsuite Professional

Burpsuite is security platform created specifically for the purposes of intensive web application testing. Its capabilities cover the entire vulnerability assessment process, from mapping and analysis of an application to the exploitation identified vulnerabilities.

IBM Appscan

IBM Appscan is an industry-leading tool designed for in-depth security testing of web and mobile software applications. It allows for both dynamic and static analysis of application code across the entire stack, from data-layer procedures up to front-end scripts.

Rhino Security Labs Proprietary Tools and Scripts

Additional Custom Scripts and Application

In addition to the above tools, Rhino Security Labs also makes use of its own proprietary tools and scripts to quickly adapt to new and unique environments.

APPENDIX B: LIST OF CHANGES MADE TO TECHCORP SYSTEMS

The following changes were made to the environment in scope. These do not necessarily represent a significant impact to the environment, but are included for the full accounting of modifications by the penetration testing team at Rhino Security Labs.

Affected Area	No changes were made to the environment in scope, such as creating new user accounts or uploading files to the target system. This is provided as the full accounting of modifications by the penetration testing team at Rhino Security Labs.

APPENDIX C: OWASP TOP 10 APPLICATION SECURITY RISKS

The Open Web Application Security Project surveys security professionals across the globe collecting information on the most frequently found security vulnerabilities within web applications. The results are compiled, categorized, and ranked to determine the top ten. Rhino Security Labs uses both automated and manual processes to identify if these vulnerabilities are present within your application. The appendix is to help explain how some of your vulnerabiltiies might be categorized.

A1-Injection	Injection flaws, such as SQL, OS, XXE, and LDAP injection occur when untrusted data is sent to an interpreter as part of a command or query. The attacker's hostile data can trick the interpreter into executing unintended commands or accessing data without proper authorization.
A2-Broken Authentication and Session Management	Application functions related to authentication and session management are often implemented incorrectly, allowing attackers to compromise passwords, keys, or session tokens, or to exploit other implementation flaws to assume other users' identities (temporarily or permanently)
A3-Cross-Site Scripting (XSS)	XSS flaws occur whenever an application includes untrusted data in a new web page without proper validation or escaping, or updates an existing web page with user supplied data using a browser API that can create JavaScript. XSS allows attackers to execute scripts in the victim's browser which can hijack user sessions, deface web sites, or redirect the user to malicious sites.
A4-Broken Access Control	Restrictions on what authenticated users are allowed to do are not properly enforced. Attackers can exploit these flaws to access unauthorized functionality and/or data, such as access other users' accounts, view sensitive files, modify other users' data, change access rights, etc.
A5-Security Misconfiguration	Good security requires having a secure configuration defined and deployed for the application, frameworks, application server, web server, database server, platform, etc. Secure settings should be defined, implemented, and maintained, as defaults are often insecure. Additionally, software should be kept up to date.
A6-Sensitive Data Exposure	Many web applications and APIs do not properly protect sensitive data, such as financial, healthcare, and PII. Attackers may steal or modify such weakly protected data to conduct credit card fraud, identity theft, or other crimes. Sensitive data deserves extra protection such as encryption at rest or in transit, as well as special precautions when exchanged with the browser.

A7-Insufficient Attack Protection

The majority of applications and APIs lack the basic ability to detect, prevent, and respond to both manual and automated attacks. Attack protection goes far beyond basic input validation and involves automatically detecting, logging, responding, and even blocking exploit attempts. Application owners also need to be able to deploy patches quickly to protect against attacks.

A8-Cross-Site Request Forgery (CSRF)

A CSRF attack forces a logged-on victim's browser to send a forged HTTP request, including the victim's session cookie and any other automatically included authentication information, to a vulnerable web application. Such an attack allows the attacker to force a victim's browser to generate requests the vulnerable application thinks are legitimate requests from the victim.

A9-Using Components with Known Vulnerabilities

Components, such as libraries, frameworks, and other software modules, run with the same privileges as the application. If a vulnerable component is exploited, such an attack can facilitate serious data loss or server takeover. Applications and APIs using components with known vulnerabilities may undermine application defenses and enable various attacks and impacts.

A10-Underprotected APIs

Modern applications often involve rich client applications and APIs, such as JavaScript in the browser and mobile apps, that connect to an API of some kind (SOAP/XML, REST/JSON, RPC, GWT, etc.). These APIs are often unprotected and contain numerous vulnerabilities.

888.944.8679

info@rhinosecuritylabs.com

1200 East Pike Street Suite 510 | Seattle, WA 98122

RHINO
SECURITY LABS