



**REDSIEGE**  
Information Security

# Final Report

EXTERNAL NETWORK PENETRATION TEST

NAKATOMI TRADING CORP

JULY 15, 1988

SIMPLE

NAVIGATION

We use links in the document so you can quickly navigate through the document and find the information you want

TABLE OF CONTENTS

EXECUTIVE SUMMARY ..... 3

FINDINGS SUMMARY ..... 4

FINDINGS CLASSIFICATIONS ..... 5

FINDINGS..... 6

    CRITICAL RISK FINDINGS..... 6

        Finding-01 Lack of Multi-Factor Authentication ..... 6

        Finding-02 Password Reuse..... 7

    HIGH RISK FINDINGS..... 9

        Finding-03 Default Credentials..... 9

    MEDIUM RISK FINDINGS..... 10

        Finding-04 Website Missing HSTS Header..... 10

    LOW RISK FINDINGS..... 12

        Finding-05 Missing Web Server Content-Security-Policy Header..... 12

METHODOLOGY ..... 14

APPENDIX..... 15

    FINDING CATEGORIES ..... 15

    TABLE OF FIGURES ..... 16

## THE SUMMARY FOR THE BUSY EXECUTIVE

*Red Siege focuses recommendations to highlight the strategic actions that should be taken by management to have the greatest impact on security.*

*We also provide a high-level overview of the issues to quickly give the leadership the information they need.*

## EXECUTIVE SUMMARY

Red Siege experts evaluated the security of Nakatomi Trading Corp's (NTC) perimeter network during the course of a three-week period in July 1988. The goal of the assessment was to identify security vulnerabilities in NTC's internet facing systems and services. All issues identified by Red Siege have been manually verified and exploited (where applicable) to demonstrate the underlying risk to NTC, its employees and clients.

### FINDINGS OVERVIEW

Findings grouped by risk severity:

❗ Critical Risk issues	2
🔴 High Risk issues	4
🟡 Medium Risk issues	6
🟢 Low Risk issues	8
ℹ Informational issues	2



### KEY FINDINGS

Red Siege identified one critical vulnerability related to the authentication to NTC's VPN which allows an attacker to access internal systems. Additionally, Red Siege identified two high severity vulnerabilities that have the potential to impact visitors to NTC's website which could impact NTC's brand and reputation.

- NTC's VPN does not use two-factor authentication and multiple users were found using weak passwords. An attacker could easily guess these passwords and access the internal network, compromising the confidentiality of NTC's data and possibly leading to data loss.

- Red Siege found significant shortcomings in defenses and secure coding related to a common web related attack known as cross-site scripting (XSS). This type of attack allows targets visitors to the site, which could expose personally identifying information, credentials, or even compromise the victim's computer.

### STRATEGIC RECOMMENDATIONS

To increase the security posture of NTC, Red Siege recommends the follow strategic actions be taken:

- **IMPLEMENT TWO-FACTOR AUTHENTICATION ON PUBLIC FACING SYSTEMS.** Internet facing systems are regularly being probed and attacked. Extra care needs to be taken on these systems to prevent unauthorized access.
- **STRENGTHEN PASSWORD REQUIREMENTS.** NTC should use technical means to ban known bad/weak passwords and train users on safe password practices.
- **REQUIRE DEFENSIVE CODING TRAINING FOR DEVELOPERS.** Developers are the first line of defense when it comes to custom web applications. Developers should be made aware of common mistakes that lead to vulnerabilities and learn ways to prevent these issues before the code is run on production systems.
- **IMPLEMENT DATA ALLOWLISTING.** Data sent from a user to the webserver should always be treated as potentially malicious. Developers should identify the data expected by the application and disallow characters that are invalid.

Red Siege would like to thank the Nakatomi Trading Corp for the opportunity to work on this project. Should you have any questions regarding these findings or the contents of this report, please feel free to contact us.

## FINDINGS

### OVERVIEW AND QUICK

#### NAVIGATION

We give you a summary of the findings with links to the in-depth discussion. This allows you to quickly get to the information you need!

## FINDINGS SUMMARY

### FINDING-01 LACK OF MULTI-FACTOR AUTHENTICATION

 Critical Risk Authentication

### FINDING-02 PASSWORD REUSE

 Critical Risk Passwords

### FINDING-03 DEFAULT CREDENTIALS

 High Risk Configuration Management

### FINDING-04 WEBSITE MISSING HSTS HEADER

 Medium Risk Configuration Management

### FINDING-05 MISSING WEB SERVER CONTENT-SECURITY-POLICY HEADER

 Low Risk Configuration Management

ICONOGRAPHY  
FOR FASTER  
READING

*Easy to read icons  
mean you can  
quickly get the  
information you  
need. Icons are  
useful when  
printed in black in  
white and for  
those with color  
vision deficiency.*

## FINDINGS CLASSIFICATIONS

Each vulnerability or risk identified has been labeled as a Finding and categorized as a Critical Risk, High Risk, Medium Risk, Low Risk, or Informational, which are defined as:

### **CRITICAL RISK ISSUES**

These vulnerabilities should be addressed promptly as they may pose an immediate danger to the security of the networks, systems, or data.

Exploitation does not require advanced tools or techniques or special knowledge of the target.

### **HIGH RISK ISSUES**

These vulnerabilities should be addressed promptly as they may pose a significant danger to the security of the networks, systems, or data.

The issue is commonly more difficult to exploit but could allow for elevated permissions, loss of data, or a system downtime.

### **MEDIUM RISK ISSUES**

These vulnerabilities should be addressed in a timely manner.

Exploitation is often difficult and requires social engineering, existing access, or special circumstances.

### **LOW RISK ISSUES**

The vulnerabilities should be noted and addressed at a later date.

These issues offer very little opportunity or information to an attacker and may not pose an actual threat.

### **INFORMATIONAL ISSUES**

These issues are for informational purposes only and likely do not represent an actual threat.

## ACTIONABLE FINDINGS

Our findings include a "Validation" section will tell you how you can verify an issue is fixed. This lets your people validate fixes by themselves. Always make sure your vendor offers this!

ALWAYS  
CUSTOM  
RESULTS.  
ALWAYS!

YOU'LL NEVER SEE  
COPY/PASTE FROM A  
SCANNER...EVER!

## FINDINGS

### CRITICAL RISK FINDINGS

#### Finding-01 LACK OF MULTI-FACTOR AUTHENTICATION

##### ! Critical Risk Authentication

#### Observation

The Web VPN server, allows authentication without use of a second factor. Red Siege was able to guess valid credentials through a password spray attack and subsequently access the internal network.

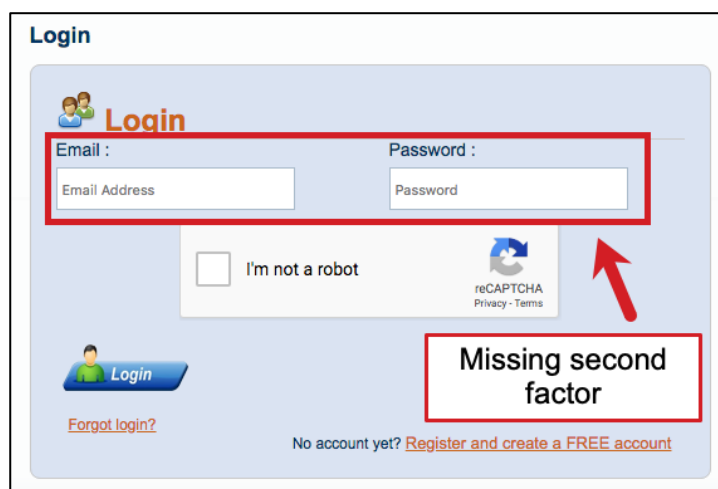


Figure 1. VPN Login Form Lacking 2nd Factor

#### Affected System

10.1.2.3 – <https://vpn.ntc.nope/vpn>

#### Description

The use of two-factor authentication is an important security control to prevent attackers from accessing systems via compromised or weak passwords. Externally facing systems, especially those with access to sensitive data such as VPN or email, should be protected with two-factor authentication. Sensitive internal systems should be protected with two-factor as well. Two-factor authentication limits the effectiveness of two common, effective password attack techniques, password spraying and credential stuffing.

With password spraying, attackers will attempt to login to a range of target accounts using a few common insecure passwords. With credential stuffing, an attacker will use credentials compromised on 3rd party websites in an attempt to access data on other sites on the internet. In both cases, two-factor authentication limits the effectiveness of the breached reused credentials as the attacker would also need to guess the two-factor token or perform a targeted social engineering attack.

A strong password can limit the effectiveness of password attacks; however, ensuring that all users select good passwords is quite difficult. In addition, if the strong password is selected but reused and then compromised elsewhere on the internet, the 2nd factor will limit the usefulness of the credential.

## Recommendations

Implement two-factor authentication on key external systems.

## References

[CIS: Two-Factor Authentication](#)

[UK National Cyber Security Centre: Defending against password spraying attacks](#)

[OWASP: Credential Stuffing](#)

[NIST Special Publication 800-63: Authenticator and Verifier Requirements](#)

## Validation

Access the system(s) in question and verify that the user is prompted for an additional authenticator besides the password.

## Finding-02 PASSWORD REUSE



### Critical Risk Passwords

## Observation

If users of different access or privilege levels have the same password, an attacker who compromises the password of a lower-privilege user could reuse that password to access other higher privilege accounts with the same password. Red Siege identified users having the same password. It appears the password may be a default password set when provisioning new users.

Password set A: 16 accounts with the same leetspeak<sup>1</sup> version of the word "password":

- |             |             |                     |
|-------------|-------------|---------------------|
| • apowell   | • jnelson   | • mfarrell          |
| • arodgers  | • karl      | • nakatomi-svc-acct |
| • hgmcclane | • lbarnes   | • tgabriel          |
| • hgruber   | • marketing | • wstuart           |
| • jmcclane  | • mbowman   | • ykomarov          |

Password set B: 3 accounts with the same password based on the word "password" followed by a number:

- |           |             |         |
|-----------|-------------|---------|
| • bbulaga | • cmatthews | • rcobb |
|-----------|-------------|---------|

---

<sup>1</sup> <https://en.wikipedia.org/wiki/Leet>

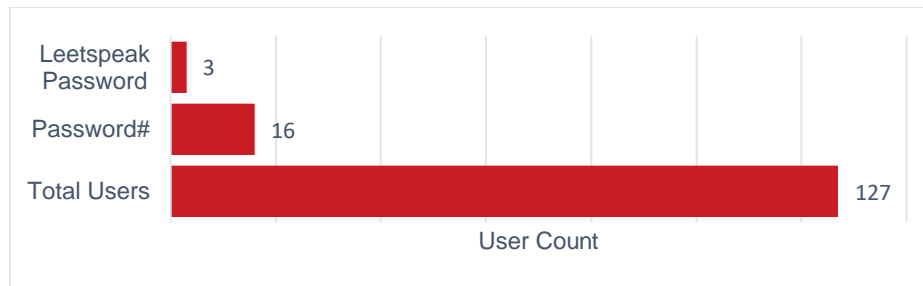


Figure 2. Number of Users with Bad Passwords

Password reuse, or sharing of passwords between accounts, is problematic from a security perspective. After identifying the password for an account, attackers typically attempt to determine if the password is shared with any other accounts. If the password is shared with a higher-privilege account, the attacker will be able to elevate privileges within the environment

## Recommendations

Client Name Short should change the passwords of the affected accounts immediately, creating a unique and strong password for each account.

Client Name Short should consider deploying a password manager that allows users to "check out" a password for a temporary password and changes the account password immediately after the password is "checked in" or the approved time limit expires.

Consider implementing multifactor authentication for administrative accounts.

Provide user awareness training stressing the risks of password reuse and the importance of using unique passwords for each account.

## References

[CIS: Critical Control 5 - Controlled Use of Administrative Privileges](#)

[NIST 800-63: Memorized Secret Verifiers](#)

[NIST 800-63: Digital Identity Guidelines FAQ](#)

## Validation

NTC uses an LDAP directory for user authentication. NTC can extract the password hashes from the directory service using the command below and crack them with a tool like John the Ripper<sup>2</sup> or Hashcat<sup>3</sup>.

```
ldapsearch -h 10.29.1.14 -W -o ldif-wrap=no -LLL -b dc=nakatomi,dc=nope -D
uid=nakatomi-svc-acct,cn=tech,ou=tech,ou=VPN-access,dc=nakatomi,dc=org uid=* uid
userPassword > user-password.txt
```

Figure 3. Extract Username and Password from Directory Service

<sup>2</sup> John is an open source password cracking utility available at <http://www.openwall.com/john/>

<sup>3</sup> Hashcat is an open source password cracking utility available at <https://hashcat.net/hashcat/>



Then, convert the hashes to a format crackable by Hashcat or John the Ripper, and use either tool to identify bad/weak passwords.

```
import base64

f = open('user-password.txt','r')
s = f.readlines()
f.close()

for l in s:
    if l[:4] == 'uid:':
        out = l[5:-1]
        good = False
    elif l[:13] == 'userPassword:':
        out += ':' + str(base64.b64decode(l[15:]).decode('utf-8'))
    print(out)
```

Figure 4. Python Script to Convert Hashes to Crackable Format

## HIGH RISK FINDINGS

### Finding-03 DEFAULT CREDENTIALS

#### High Risk Configuration Management

#### Observation

NTC uses default credentials on APC PDU systems.

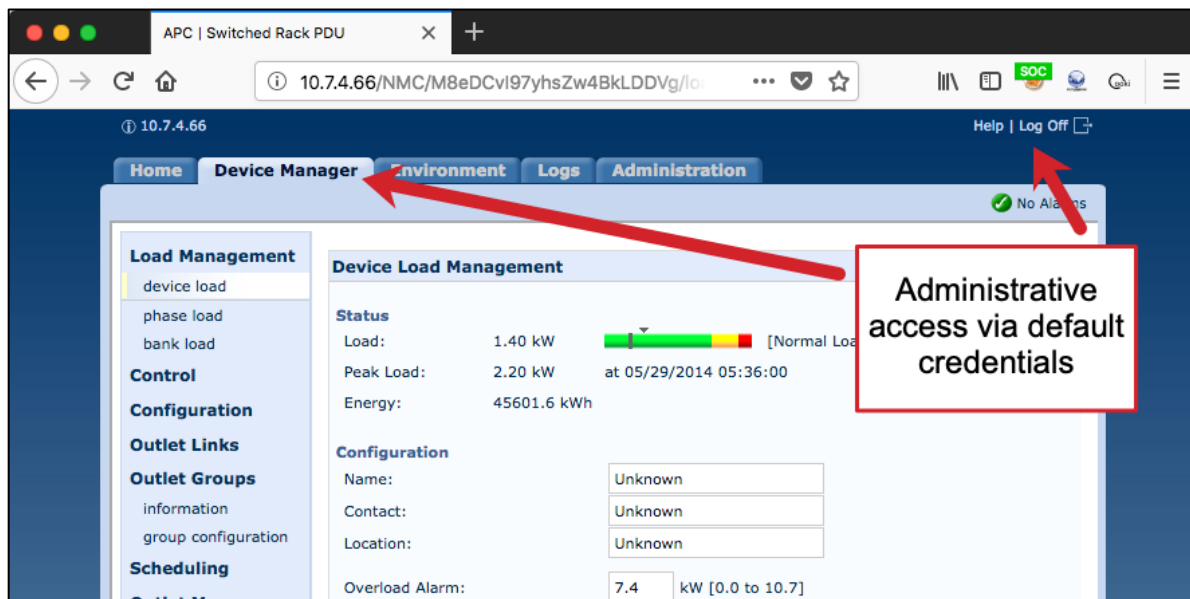


Figure 5. Access to APC PDU via Default Credentials

## Affected System

10.1.2.50

10.7.4.66

## Description

Many devices, including routers, networking equipment, IOT devices, access points, and printers have default usernames and passwords that allow administrative access. These credentials are publicly published and easily searchable. Default credentials provide an attacker with an access, often privileged access, to the systems which could compromise their integrity and the confidentiality of the data on these systems.

## Recommendations

Review the system hardening guidelines and ensure the guidelines include steps for identifying and removing default credentials

Change the default passwords and, if possible, usernames.

When creating new passwords, generate secure, long passwords that are resistant to password guessing and cracking attacks.

Use two-factor authentication for all administrative accounts where possible.

## References

[List of default credentials](#)

[CIS: Critical Control 5 - Controlled Use of Administrative Privileges](#)

[CIS: Two-Factor Authentication](#)

## Validation

Refer to the OWASP Guide, section [OTG-AUTHN-002](#) to identify default credentials. In addition, refer to the documentation for the systems in question to find default credentials.

## MEDIUM RISK FINDINGS

Finding-04 WEBSITE MISSING HSTS HEADER



**Medium Risk Configuration Management**

## Observation

Red Siege determined the application web servers in the assessment scope did not implement the HTTP **Strict-Transport-Security**<sup>4</sup> header, which helps defend against HTTPS downgrade and machine-

---

<sup>4</sup> [https://en.wikipedia.org/wiki/HTTP\\_Strict\\_Transport\\_Security](https://en.wikipedia.org/wiki/HTTP_Strict_Transport_Security)

in-the-middle attacks. Figure 6 illustrates the lack of the `Strict-Transport-Security` response header in a server response.



```
root@U:/opt/client/jasons# curl -I https://[redacted]
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Length: 4957
Content-Type: text/html; charset=utf-8
Expires: -1
Server: Microsoft-IIS/8.5
X-AspNet-Version: 4.0.30319
X-UA-Compatible: IE=8
Date: Tue, 28 Aug 2018 21:13:14 GMT
Set-Cookie: ASP.NET_SessionId=2u22orhshcog2omxebzeykvg; path=/; secure; HttpOnly; HttpOnly;Secure
```

Strict-Transport-Security header not present

Figure 6. HSTS Header not Present

## Affected Systems

10.1.2.3 – <https://www.ntc.nope>

## Description

The HTTP `Strict-Transport-Security` header prevents the accidental exposure of potentially sensitive application information over unencrypted channels. The header instructs web browsers to only interact with the web server using HTTPS. In the event of a downgrade attack<sup>5</sup> or a server misconfiguration, the web browser will refuse to access the web server over unencrypted HTTP channels.

## Recommendations

Client Name Short should configure application web servers to include the `Strict-Transport-Security` header in all server responses as follows.

```
Strict-Transport-Security: max-age=31536000;
```

## References

[Mozilla Developer Network: Strict-Transport-Security](#)

[OWASP: HTTP Strict Transport Security Cheat Sheet](#)

## Validation

The presence of the `Strict-Transport-Security` header can be validated using the PowerShell console.

```
Invoke-WebRequest -Uri https://example.tld | Select-Object -ExpandProperty Headers
```

The presence of the `Strict-Transport-Security` header can be validated using curl on Linux systems.

<sup>5</sup> [https://en.wikipedia.org/wiki/Downgrade\\_attack](https://en.wikipedia.org/wiki/Downgrade_attack)

```
curl -skI https://example.tld | grep -i strict-transport-security
```

When HSTS is enabled, you should see output similar to that shown Figure 7.

```
$ curl -skI https://www.ntc.nope
HTTP/1.1 200 OK
Date: Mon, 15 July 1985 15:39:45 GMT
Strict-Transport-Security: max-age=63072000; includeSubdomains;
Last-Modified: Wed, 28 Mar 2018 22:17:10 GMT
Accept-Ranges: bytes
Content-Length: 14968
Vary: Accept-Encoding
Content-Type: text/html
```

Figure 7. Retrieving Web Server Headers via Curl

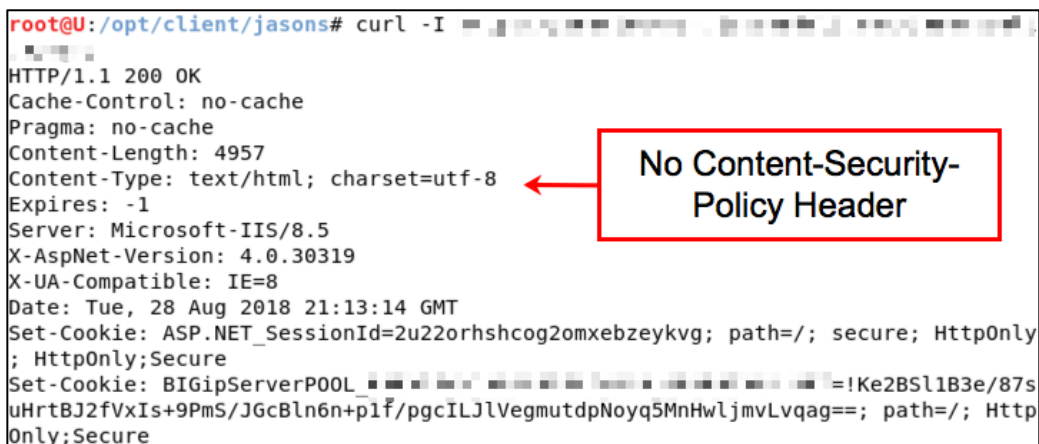
## LOW RISK FINDINGS

### Finding-05 MISSING WEB SERVER CONTENT-SECURITY-POLICY HEADER ^

 Low Risk Configuration Management

#### Observation

Red Siege identified web servers which did not include the **Content-Security-Policy** header in server responses. Red Siege sampled responses from all in-scope web applications and did not find the **Content-Security-Policy** header present in any server responses. An example server response can be seen in Figure 8



```
root@U:/opt/client/jasons# curl -I
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Length: 4957
Content-Type: text/html; charset=utf-8
Expires: -1
Server: Microsoft-IIS/8.5
X-AspNet-Version: 4.0.30319
X-UA-Compatible: IE=8
Date: Tue, 28 Aug 2018 21:13:14 GMT
Set-Cookie: ASP.NET_SessionId=2u22orhshcog2omxebzeykvg; path=/; secure; HttpOnly; HttpOnly; Secure
Set-Cookie: BIGipServerP00L...=!Ke2BSl1B3e/87s
uHrtBJ2fVxIs+9PmS/JGcBlN6n+p1f/pgcILJlVegmutdpNoyq5MnHwljmvLvqag==; path=/; HttpOnly; Secure
```

No Content-Security-Policy Header

Figure 8. Content-Security-Policy Header Missing

#### Affected Systems

10.1.2.3 - https://www.ntc.nope

## Description

The **Content-Security-Policy** header (CSP) allows site administrators to control what resources – i.e., scripts, images, frames, etc. – can be loaded in a given website or web page. A properly-configured CSP implementation can mitigate the impact of a cross-site scripting (XSS) attack by preventing the execution of arbitrary scripting code injected into server responses.

## Recommendations

NTC should apply the Mozilla Web Security Guidelines<sup>6</sup> advice for a default CSP policy as follows:

```
default-src 'none'; img-src 'self'; script-src 'self'; style-src 'self'
```

CSP configurations may interfere with the normal operation of a web site. As such, Mozilla recommends first deploying the **Content-Security-Policy-Report-Only** header with the directives noted above. This allows the CSP policy to be tuned without interrupting the normal operation of the web site. Once issues have been identified and resolved, the **Content-Security-Policy-Report-Only** header can be replaced with the **Content-Security-Policy** header to begin enforcing CSP configurations.

## References

[Mozilla Developer Network: Content-Security-Policy](#)

[OWASP: Secure Headers Project](#)

<https://securityheaders.com>

## Validation

The presence of the **Content-Security-Policy** header can be validated using the PowerShell console.

```
Invoke-WebRequest -Uri https://example.tld | Select-Object -Expand Headers
```

The presence of the **Content-Security-Policy** header can be validated using curl on Linux systems.

```
curl -skI https://example.tld | grep -i content-security-policy
```

The highlighted portion below in Figure 9 should be present with the feature is enabled.

```
$ curl -skI https://www.ntc.nope
HTTP/1.1 200 OK
Date: Fri, 08 Jun 2018 15:39:45 GMT
Server: Apache
Content-Security-Policy: frame-ancestors 'self';
Last-Modified: Wed, 28 Mar 2018 22:17:10 GMT
Accept-Ranges: bytes
...output truncated for brevity...
```

Figure 9. Verifying Content-Security-Policy with Curl

---

<sup>6</sup> <https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Content-Security-Policy>

## METHODOLOGY

This portion intentionally left blank. This section includes examples of the techniques used to assess the target systems.

## APPENDIX

### FINDING CATEGORIES

Vulnerability categories and the related weaknesses are listed below:

ARCHITECTURE – Related to system or network design

AUTHENTICATION – User authentication and access rights

CONFIGURATION MANAGEMENT – Related to system configuration and hardening

CRYPTOGRAPHY – Implementation and use of encryption and hashing

DATA VALIDATION – Input validation and data handling

DATA EXPOSURE – Unintended or excessive exposure of data

PASSWORD MANAGEMENT – Password storage and complexity requirements

PATCH MANAGEMENT – Patch and vulnerability management of systems

PERMISSIONS AND ACCESS CONTROL – Management of permissions, privileges, and features related to access control

## TABLE OF FIGURES

Figure 1. VPN Login Form Lacking 2nd Factor.....	6
Figure 2. Number of Users with Bad Passwords .....	8
Figure 3. Extract Username and Password from Directory Service .....	8
Figure 4. Python Script to Convert Hashes to Crackable Format .....	9
Figure 5. Access to APC PDU via Default Credentials .....	9
Figure 6. HSTS Header not Present .....	11
Figure 7. Retrieving Web Server Headers via Curl.....	12
Figure 8. Content-Security-Policy Header Missing .....	12
Figure 9. Verifying Content-Security-Policy with Curl.....	13



Prepared by Red Siege, LLC. Portions of this document, and the templates used in its production are the property of Red Siege, LLC. and cannot be used or copied without permission.

While precautions have been taken in the preparation of this document, Red Siege, LLC., the publisher, and the author(s) assume no responsibility for errors, omissions, or for damages resulting from the use of the information contained herein. Use of Red Siege, LLC and its services does not guarantee the security of any system, or that computer intrusions will not occur.