Harshdeep Shrivastava – 20BCY10068
Sainyam Agrawal – 20BCY10151
Aswin Shailajan – 20BCE10209

Security Awareness Training

1. Introduction

In today's rapidly evolving digital landscape, organizations face a multitude of security threats that can compromise their sensitive information and undermine their operations. Cyberattacks, data breaches, and other malicious activities have become increasingly sophisticated, targeting not only technological vulnerabilities but also exploiting human error and ignorance.

The purpose of this report is to present the findings of the foot-printing, reconnaissance and cyber-attack activities conducted on the domain "testfire.net". The assessment aimed to gather information about the target domain's infrastructure, identify potential vulnerabilities, and provide recommendations for improving its security posture.

The report begins by discussing the significance of security awareness training in today's threat landscape, highlighting the changing nature of cyber threats and the critical role employees play in safeguarding organizational assets.

Furthermore, the report addresses all the findings of the footprinting, reconnaissance and various attacks done by numerous tools and methods. It gives a detailed description about the vulnerability, how it can be found, how it can be exploited, it’s Common Vulnerability Exposure(CVE) number , its severity score based on Common Vulnerability Scoring System(CVSS), and its impacts. The report also provides with the output results of various penetration testing and vulnerability testing tools.

Finally, the report addresses the challenges organizations may encounter when implementing security awareness training initiatives, such as resistance to change, limited resources, and the need for ongoing reinforcement. It offers insights into overcoming these challenges and provides practical recommendations for creating a sustainable and effective security awareness training program.

2. Information Gathering for Security Awareness Training

Before developing and delivering security awareness training programs, it is essential to gather relevant information to understand the various aspects of potential threats and safe practices. This section focuses on different methods of information gathering that can provide valuable insights for designing effective security awareness training programs. The following smaller titles explore specific areas of information gathering in the context of security awareness training.

- **Email Footprint Analysis:**
  Email Footprint Analysis refers to the process of gathering and analyzing information related to an organization's email communication to identify potential vulnerabilities and risks. The analysis begins by examining the email address format, which can provide clues about naming conventions used by organizations.

- **DNS Information Gathering:**
  DNS (Domain Name System) Information Gathering involves collecting data related to an organization's DNS infrastructure to assess potential security risks and vulnerabilities by examining DNS information, one can gain insights into the infrastructure, services, and potential vulnerabilities associated with a domain.

- **WHOIS Information Gathering:**
  WHOIS Information Gathering involves extracting and analyzing publicly available domain registration information to gain insights into the ownership and administration of domains.

  WHOIS records contain details such as domain registrants, contact information, registration dates, and domain expiration dates. By examining WHOIS information, organizations can identify potential risks associated with domains and educate employees on the importance of domain management and protection. This information can be useful for various purposes, such as identifying domain owners, investigating potential misuse, or performing cybersecurity research.

- **Information Gathering for Social Engineering Attacks:**
  Social Engineering attacks exploit human psychology and behavior to manipulate individuals into divulging sensitive information or performing unauthorized actions. Gathering information related to social engineering techniques can help organizations educate their employees about the tactics employed by attackers.

Social engineering attacks often leverage personal or publicly available information to build rapport, establish trust, or deceive individuals. By understanding these techniques and the potential sources of information, employees can become more vigilant and adept at recognizing and mitigating social engineering attempts.

- **Information Gathering for Physical Security Assessments:**

Physical Security Assessments involve evaluating an organization's physical infrastructure, access controls, and security measures to identify vulnerabilities and potential entry points for unauthorized individuals.

Physical security assessments often involve gathering information about an organization's premises, security policies, surveillance systems, and employee access controls. By raising awareness about physical security risks and best practices, organizations can enhance employee vigilance and contribute to a more secure work environment.

3. Vulnerability Identification

Vulnerability identification is a critical component of security awareness training programs, as it enables organizations to understand and address potential weaknesses in their systems and applications. This section focuses on the process of identifying vulnerabilities and provides essential information, including vulnerability names, Common Weakness Enumeration (CWE) codes, and corresponding Open Web Application Security Project (OWASP) categories and descriptions. The following smaller titles cover different aspects of vulnerability identification within the context of security awareness training.

These are some vulnerabilities that were found during static code analysis. To enable these vulnerabilities, place the highlighted text in AltoroJ's WEB-INF folder and enable those properties you would like to enable by uncommenting them

Behavioural Vulnerabilities

- enableAdminFunctions property enables administrative functions in AltoroJ (adding users, changing passwords, etc). Enabling this function and then running AppScan on AltoroJ will likely trash the database and make AltoroJ unusable. You will need to either manually delete the database every time or enable database.reinitializeOnStart and restart Tomcat

  #enableAdminFunctions=true

- specialLink property changes certain links in AltoroJ (e.g. "Search News Articles" in the authenticated user navigation panel) to the link specified in this property. This allows for demonstrating AppScan's malware detection capabilities. WARNING: It is STRONGLY RECOMMENDED that you USE ONLY SITES THAT DO NOT SERVE ACTUAL MALWARE (use other types of undesired sites)

  #specialLink=http://www.warez.com

- advancedStaticPageProcessing property enables advanced file/page lookup using Bash or Command Prompt. This exposes AltoroJ to remote command execution and system path traversal WARNING: USE THIS SETTING AT YOUR OWN RISK. USE IT ON PROTECTED SYSTEMS OR IN A VM ONLY ACCESSIBLE FROM THE HOST

  #advancedStaticPageProcessing=true

- enableFeedbackRetention property turns on AltoroJ functionality to store feedback information in its database. This allows for demonstration of persistent XSS attacks. WARNING: Enabling this function and then running AppScan on AltoroJ will likely trash the database and make AltoroJ unusable. You will need to either manually delete the database every time or enable database.reinitializeOnStart and restart Tomcat

  #enableFeedbackRetention=true

Database Vulnerabilities

- database.alternateDataSource property allows AltoroJ to connect to an external database (e.g. DB2). This property is the name of the data source configured in tomcat's context.xml which contains details for this database connection. This will also require adding in the appropriate JDBC driver JAR. See the following URL for more details: https://tomcat.apache.org/tomcat-7.0-doc/jndi-datasource-examples-howto.html
  IMPORTANT: If you use this property, you must also set database.reinitializeOnStart to true the first time you run AltoroJ and log into the application. This is required to initialize AltoroJ data. You can disable database.reinitializeOnStart afterwards. Failure to use database.reinitializeOnStart=true at least once will cause AltorJ to fail unless you manually create DB contents

  #database.alternateDataSource=jdbc/AltoroDB

- database.reinitializeOnStart setting forces AltoroJ to reinitialize its database every time Tomcat is restarted instead of only initializing it if AltoroJ database does not exist. You MUST use this setting at least once if using an external DB.

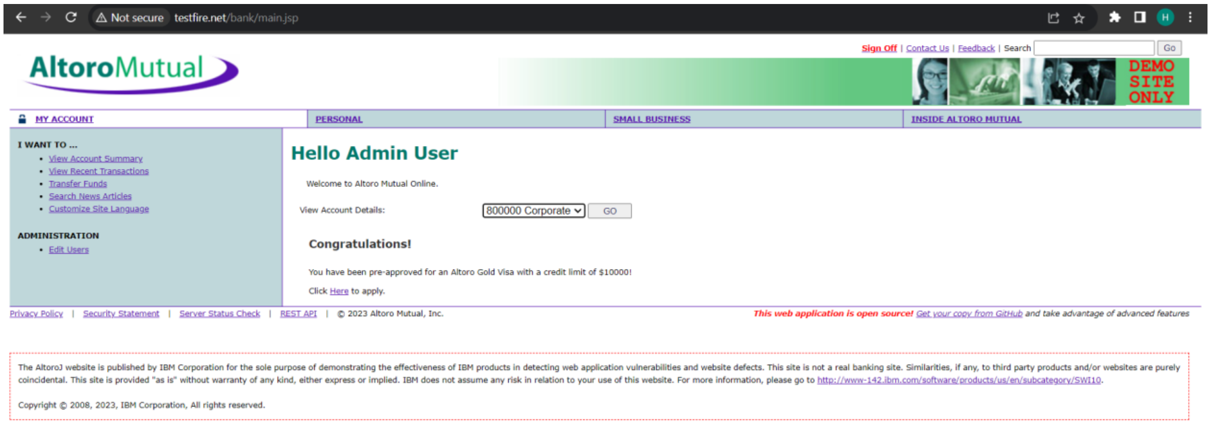
  #database.reinitializeOnStart=true

- Some of the Non-Public News Articles, which are hidden from search settings can be found and read after obtaining the source code or by even inspecting the correct element in the browser.

```
<publication>
		<id>4</id>
		<date>1/8/2004</date>
		<title>Altoro Mutual bank about to purchase 10 AppScan and 50 AppScan DE
from Watchfire inc.</title>
		<isPublic>False</isPublic>
</publication>
```

- The website is also vulnerable to XSS attacks in the feedback form fields.

- Upon further dynamic analysis, it was found that the Website is Vulnerable to SQL Injection attacks. During the login process, if some special characters are given in password, the process crashes, this denotes that the username and password are being processed in clear text and no encryption or hashing algorithms are being used. To gain access of admin level clearance use Username = admin and Password = ‘ or 1=1-- . This plays with the sql syntax used to process the password and makes the condition true by using 1=1--.



⚠ Not secure testfire.net/bank/main.jsp

Sign Off | Contact Us | Feedback | Search Go

AltoroMutual

DEMO SITE ONLY

MY ACCOUNT | PERSONAL | SMALL BUSINESS | INSIDE ALTORO MUTUAL

I WANT TO ...

- View Account Summary
- View Recent Transactions
- Transfer Funds
- Search News Articles
- Customize Site Language

ADMINISTRATION

- Edit Users

Hello Admin User

Welcome to Altoro Mutual Online.

View Account Details: 800000 Corporate GO

Congratulations!

You have been pre-approved for an Altoro Gold Visa with a credit limit of $10000!

Click Here to apply.

Privacy Policy | Security Statement | Server Status Check | REST API | © 2023 Altoro Mutual, Inc.

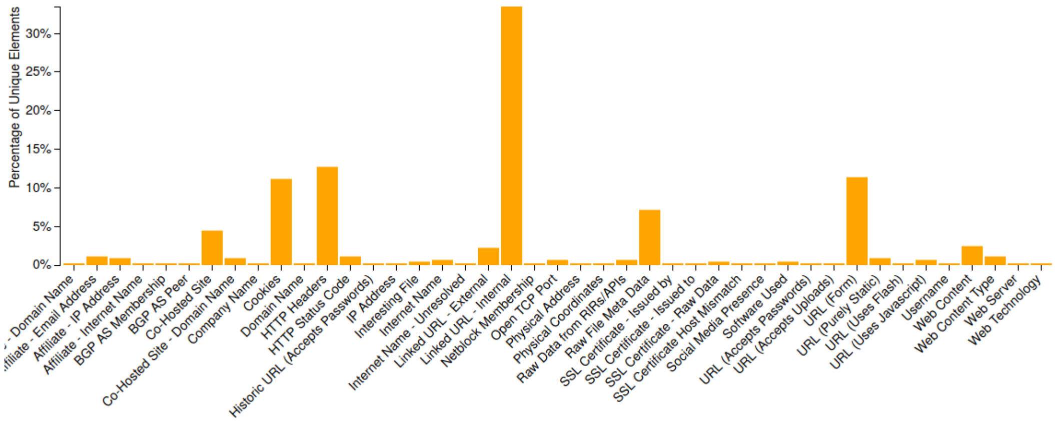
This web application is open source! Get your copy from GitHub and take advantage of advanced features

The AltoroJ website is published by IBM Corporation for the sole purpose of demonstrating the effectiveness of IBM products in detecting web application vulnerabilities and website defects. This site is not a real banking site. Similarities, if any, to third party products and/or websites are purely coincidental. This site is provided "as is" without warranty of any kind, either express or implied. IBM does not assume any risk in relation to your use of this website. For more information, please go to http://www-142.ibm.com/software/products/us/en/subcategory/SWI10.

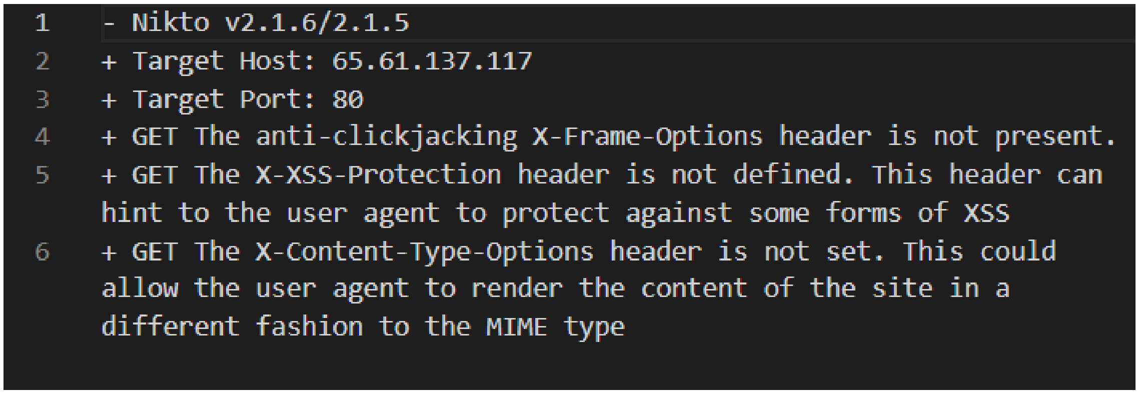
Copyright © 2008, 2023, IBM Corporation, All rights reserved.

Tool Outputs and Findings

- Reconnaissance Bar Graph:



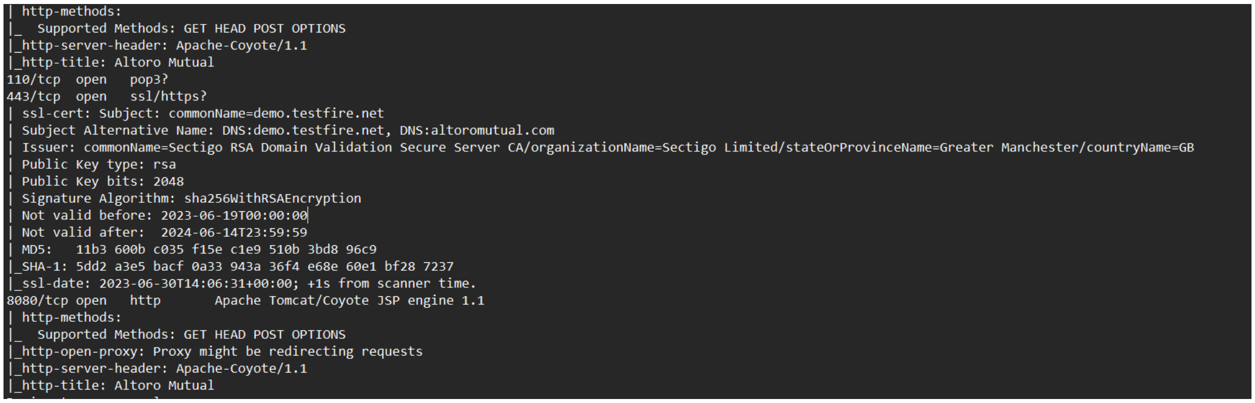
- Nikto Output on the Open Ports found



```
- Nikto v2.1.6/2.1.5
+ Target Host: 65.61.137.117
+ Target Port: 80
+ GET The anti-clickjacking X-Frame-Options header is not present.
+ GET The X-XSS-Protection header is not defined. This header can
hint to the user agent to protect against some forms of XSS
+ GET 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
```

This Indictes that the port 80 is not secured properly.

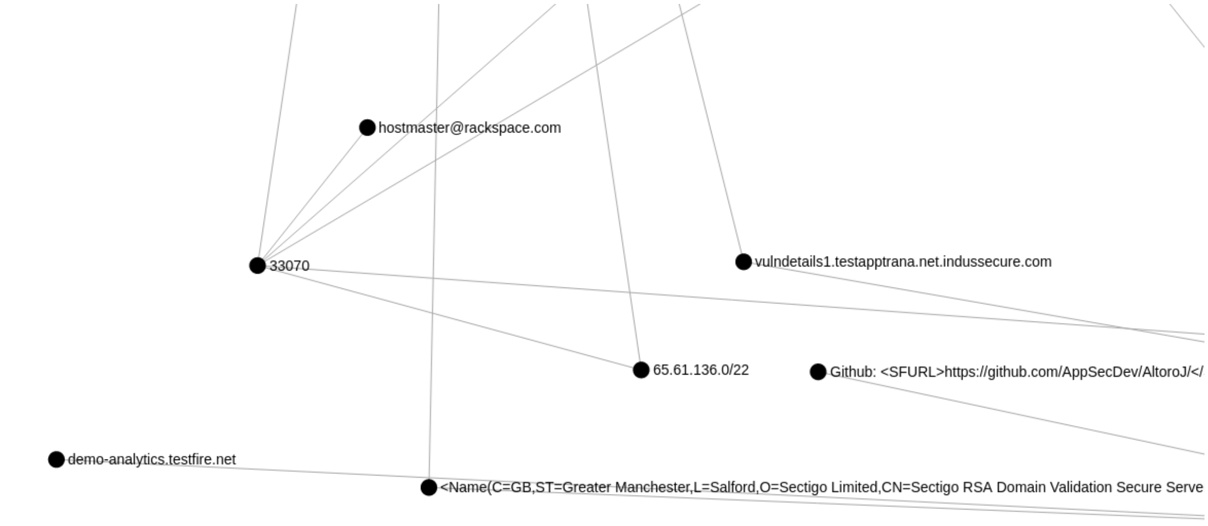
- HTTP 1.1 Certificate

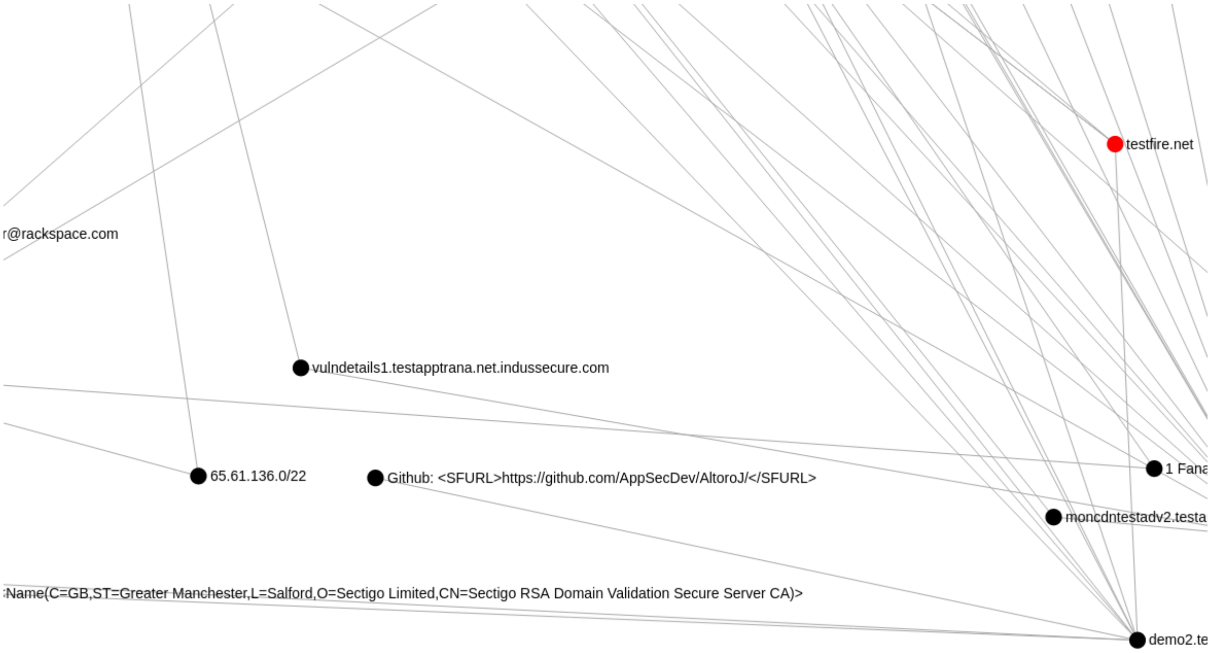


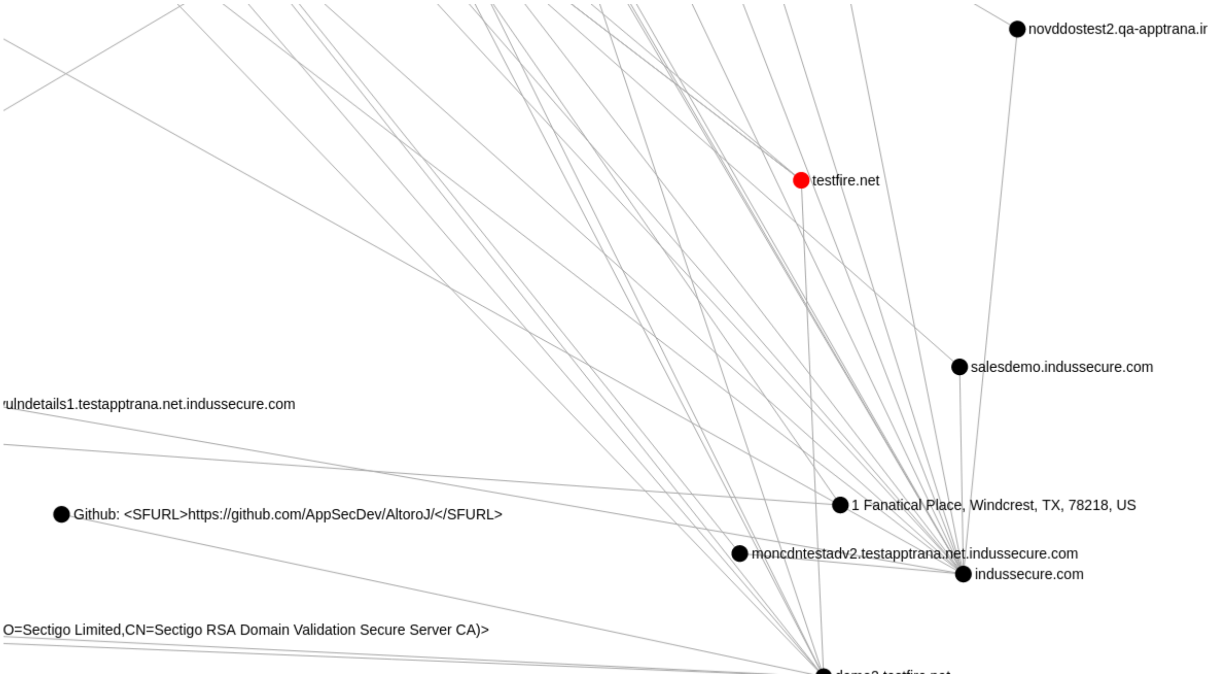
```
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-server-header: Apache-Coyote/1.1
|_http-title: Altoro Mutual
110/tcp  open   pop3?
443/tcp  open   ssl/https?
| ssl-cert: Subject: commonName=demo.testfire.net
| Subject Alternative Name: DNS:demo.testfire.net, DNS:altoromutual.com
| Issuer: commonName=Sectigo RSA Domain Validation Secure Server CA/organizationName=Sectigo Limited/stateOrProvinceName=Greater Manchester/countryName=GB
| Public Key type: rsa
| Public Key bits: 2048
| Signature Algorithm: sha256WithRSAEncryption
| Not valid before: 2023-06-19T00:00:00
| Not valid after:  2024-06-14T23:59:59
| MD5:   11b3 600b c035 f15e c1e9 510b 3bd8 96c9
|_SHA-1: 5dd2 a3e5 bacf 0a33 943a 36f4 e68e 60e1 bf28 7237
|_ssl-date: 2023-06-30T14:06:31+00:00; +1s from scanner time.
8080/tcp open   http       Apache Tomcat/Coyote JSP engine 1.1
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-open-proxy: Proxy might be redirecting requests
|_http-server-header: Apache-Coyote/1.1
|_http-title: Altoro Mutual
```

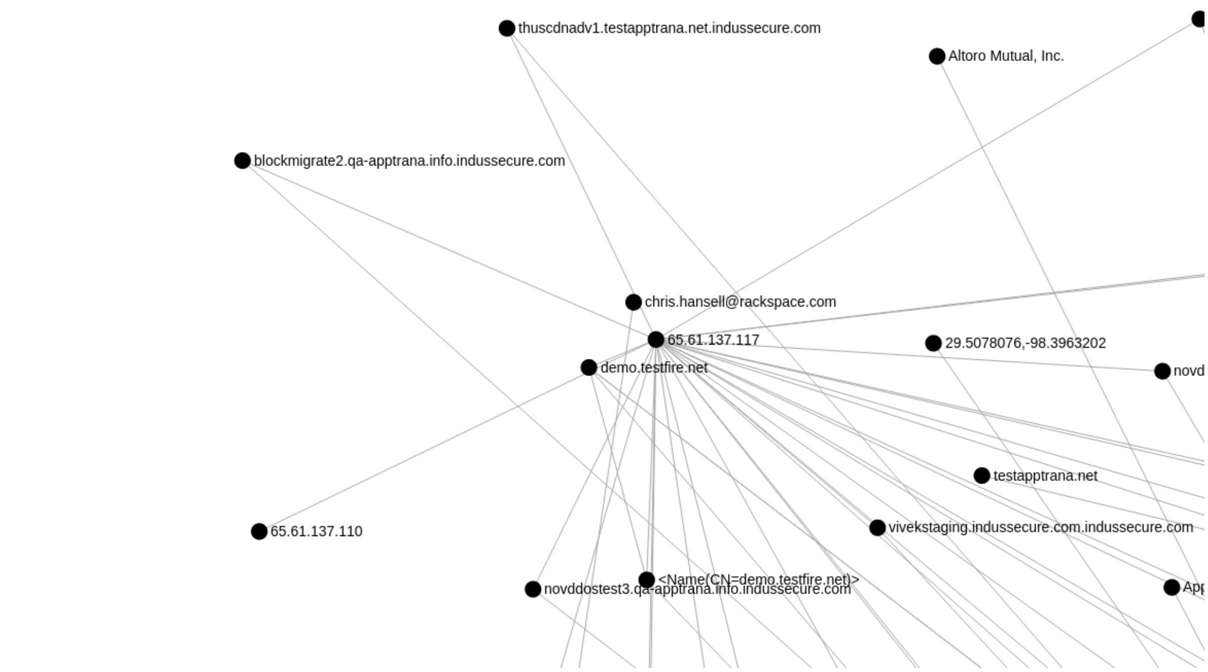
- Open TCP ports:
  - 25/tcp
  - 80/tcp
  - 110/tcp
  - 443/tcp
  - 8080/tcp

- Server IP Address: 65.61.137.117
- HTTP Version: 1.1
- Signature Algorithm: sha256WithRSAEncryption
- Public Key type: rsa
- Public Key bits: 2048
- Web Graph of all the direct and indirect relations found during web Scraping

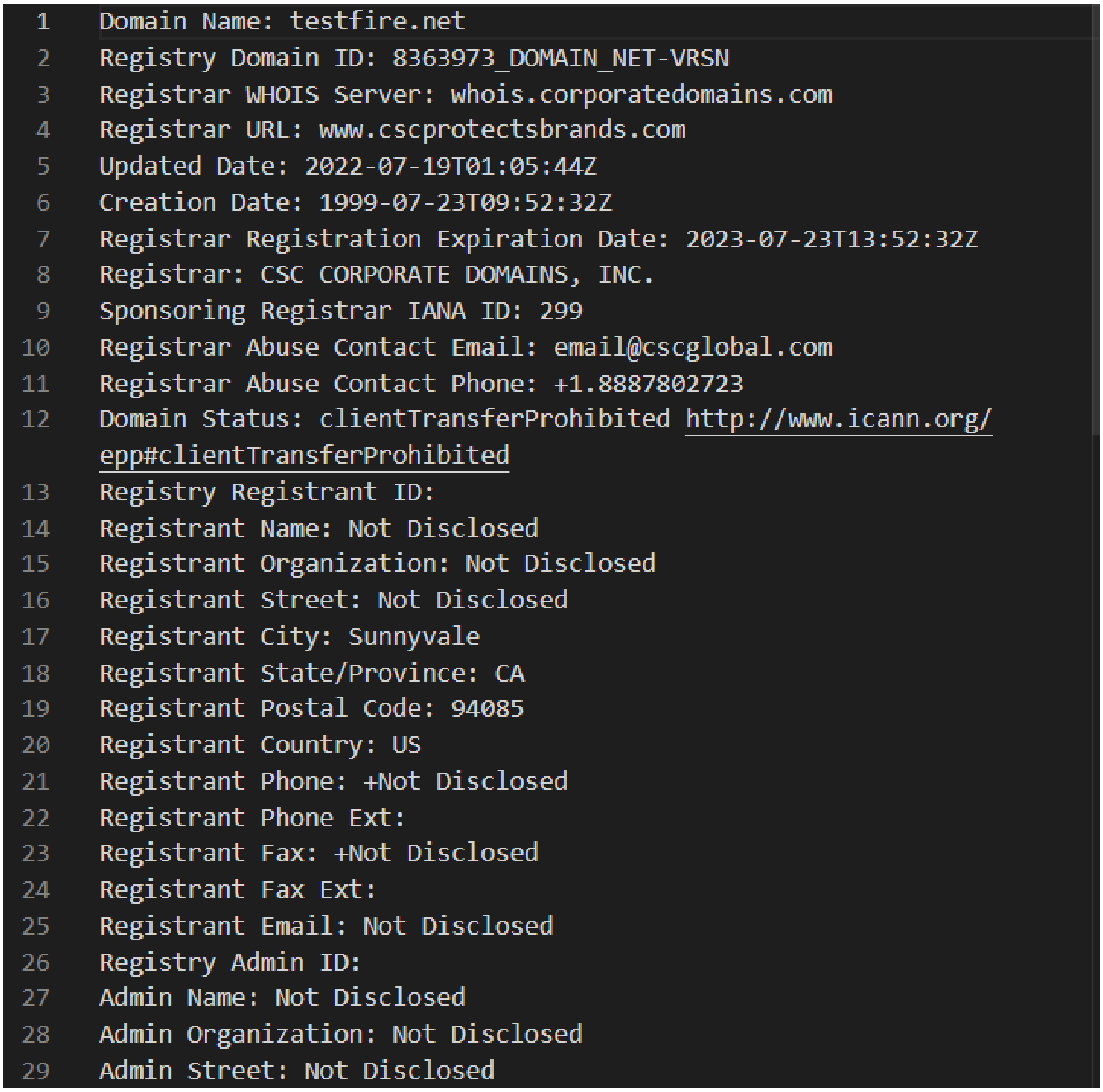

hostmaster@rackspace.com
33070
vulndetails1.testapptrana.net.indussecure.com
65.61.136.0/22
Github: <SFURL>https://github.com/AppSecDev/AltoroJ/</
demo-analytics.testfire.net
<Name(C=GB,ST=Greater Manchester,L=Salford,O=Sectigo Limited,CN=Sectigo RSA Domain Validation Secure Serve


testfire.net
r@rackspace.com
vulndetails1.testapptrana.net.indussecure.com
1 Fana
65.61.136.0/22
Github: <SFURL>https://github.com/AppSecDev/AltoroJ/</SFURL>
moncdntestadv2.testa
:Name(C=GB,ST=Greater Manchester,L=Salford,O=Sectigo Limited,CN=Sectigo RSA Domain Validation Secure Server CA)>
demo2.te


novddostest2.qa-apptrana.ir
testfire.net
salesdemo.indussecure.com
ulndetails1.testapptrana.net.indussecure.com
1 Fanatical Place, Windcrest, TX, 78218, US
Github: <SFURL>https://github.com/AppSecDev/AltoroJ/</SFURL>
moncdntestadv2.testapptrana.net.indussecure.com
indussecure.com
O=Sectigo Limited,CN=Sectigo RSA Domain Validation Secure Server CA)>
demo2.testfire.net


thuscdnadv1.testapptrana.net.indussecure.com
Altoro Mutual, Inc.
blockmigrate2.qa-apptrana.info.indussecure.com
chris.hansell@rackspace.com
65.61.137.117
29.5078076,-98.3963202
demo.testfire.net
novd
testapptrana.net
65.61.137.110
vivekstaging.indussecure.com.indussecure.com
<Name(CN=demo.testfire.net)>
novddostest3.qa-apptrana.info.indussecure.com
App

- WHOIS Output:



```
Domain Name: testfire.net
Registry Domain ID: 8363973_DOMAIN_NET-VRSN
Registrar WHOIS Server: whois.corporatedomains.com
Registrar URL: www.cscprotectsbrands.com
Updated Date: 2022-07-19T01:05:44Z
Creation Date: 1999-07-23T09:52:32Z
Registrar Registration Expiration Date: 2023-07-23T13:52:32Z
Registrar: CSC CORPORATE DOMAINS, INC.
Sponsoring Registrar IANA ID: 299
Registrar Abuse Contact Email: email@cscglobal.com
Registrar Abuse Contact Phone: +1.8887802723
Domain Status: clientTransferProhibited http://www.icann.org/epp#clientTransferProhibited
Registry Registrant ID:
Registrant Name: Not Disclosed
Registrant Organization: Not Disclosed
Registrant Street: Not Disclosed
Registrant City: Sunnyvale
Registrant State/Province: CA
Registrant Postal Code: 94085
Registrant Country: US
Registrant Phone: +Not Disclosed
Registrant Phone Ext:
Registrant Fax: +Not Disclosed
Registrant Fax Ext:
Registrant Email: Not Disclosed
Registry Admin ID:
Admin Name: Not Disclosed
Admin Organization: Not Disclosed
Admin Street: Not Disclosed
```

4. Vulnerability Path and Parameter Identification

Identifying vulnerability paths and parameters is crucial for understanding how attackers can exploit weaknesses in systems and applications. By recognizing these paths and parameters, organizations can effectively implement preventive measures and educate employees on secure coding practices. This section explores various methods, types, tools, best practices, and challenges associated with vulnerability path and parameter identification.

1. Methods for Identifying Vulnerability Paths and Parameters:

Identifying vulnerability paths and parameters requires a systematic approach to analyze systems, applications, and code to uncover potential entry points and data inputs that could be exploited by attackers.

Tools like nmap, nikto, sqlmap and spideerfoot were mainly used to find the vulnerabilities in the website.

2. Types of Vulnerability Paths and Parameters:

Vulnerability paths and parameters can manifest in different forms, depending on the nature of the application and the type of vulnerability being targeted.

Common types of vulnerability paths include input validation, authentication, authorization, session management, and data storage. Understanding these types helps organizations identify the specific areas where vulnerabilities may be present and develop appropriate countermeasures.

3. Common Tools and Techniques for Identifying Vulnerability Paths and Parameters:

Several tools and techniques are available to assist in identifying vulnerability paths and parameters within systems and applications.

Tools such as static code analyzers, vulnerability scanners, and web application security scanners can automate the identification process and provide insights into potential vulnerabilities. Additionally, manual techniques, including manual code review and penetration testing, offer a deeper understanding of application vulnerabilities.

4. Best Practices for Vulnerability Path and Parameter Identification:

Following best practices in vulnerability path and parameter identification ensures a thorough and effective analysis of systems and applications.

Best practices include conducting regular security assessments, adopting a layered defense approach, leveraging automated scanning tools, involving security experts, and fostering a culture of security awareness and collaboration within the organization.

5. Challenges and Limitations of Vulnerability Path and Parameter Identification:

Vulnerability path and parameter identification may come with challenges and limitations that organizations need to consider.

Challenges may include complex application architectures, obfuscated code, limited access to source code, false positives/negatives from automated tools, and the need for skilled personnel. Recognizing these challenges allows organizations to develop strategies to overcome them and improve the effectiveness of vulnerability identification processes.

By utilizing appropriate methods, understanding different vulnerability path and parameter types, leveraging tools and techniques, following best practices, and addressing challenges, organizations can enhance their vulnerability identification efforts. This, in turn, enables the development of targeted security awareness training programs and the implementation of necessary security controls to safeguard systems and applications from potential exploits.

5. Detailed Instruction for Vulnerability Reproduction

Providing detailed instructions for vulnerability reproduction is essential for effective communication between security researchers and developers. Clear and comprehensive instructions help ensure that vulnerabilities are accurately understood and can be properly addressed. This section focuses on the importance of detailed instructions, the components of well-written instructions, steps for reproducing vulnerabilities, best practices for writing effective instructions, tools and techniques for verifying fixes, as well as the challenges and limitations associated with vulnerability reproduction instructions.

1. Importance of Providing Detailed Instructions:

Detailed instructions play a crucial role in conveying the necessary information for security researchers and developers to understand and address vulnerabilities accurately.

Clear instructions provide a consistent and reproducible approach, allowing for thorough analysis and validation of vulnerabilities. They help minimize misinterpretation and facilitate effective collaboration between security researchers and developers, leading to timely and successful resolution of identified vulnerabilities.

2. Components of a Well-Written Vulnerability Reproduction Instruction:

Well-written vulnerability reproduction instructions should include specific components to ensure clarity and accuracy in the vulnerability identification and resolution process.

Components may include a concise vulnerability description, system configuration details, prerequisite conditions, step-by-step instructions for reproducing the vulnerability, expected and observed behaviors, sample code or data inputs, and any additional supporting information.

3. Steps for Reproducing Vulnerabilities:

Reproducing vulnerabilities involves recreating the conditions and actions that lead to the vulnerability's discovery.

Step-by-step instructions should be provided, outlining the necessary setup, test environment configuration, inputs, interactions, and expected outcomes to reproduce the vulnerability consistently. This systematic approach helps security researchers and developers validate the reported vulnerability and analyze its root cause.

4. Best Practices for Writing Effective Vulnerability Reproduction Instructions:

Writing effective vulnerability reproduction instructions requires adherence to certain best practices to ensure clarity, accuracy, and reproducibility.

Best practices may include using a standardized template, providing clear and concise language, using screenshots or illustrations to enhance understanding, including all relevant information and dependencies, and organizing the instructions logically.

5. Tools and Techniques for Verifying Vulnerability Fixes:

Verifying vulnerability fixes is an important step in the vulnerability resolution process, ensuring that the identified vulnerabilities have been properly addressed.

Tools and techniques such as code review, security testing frameworks, penetration testing, and automated vulnerability scanners can assist in verifying the effectiveness of fixes. These tools and techniques aid in detecting any residual vulnerabilities or potential regressions.

6. Challenges and Limitations of Vulnerability Reproduction Instructions:

Vulnerability reproduction instructions may face challenges and limitations that can impact their effectiveness.

Challenges may include complex system configurations, dependencies on specific environments or data, inadequate information provided by the reporter, limited access to source code, or differences in testing environments. Addressing these challenges requires clear communication and collaboration between security researchers and developers.

By providing detailed instructions, including the necessary components, following best practices, and utilizing appropriate tools and techniques, organizations can facilitate accurate vulnerability reproduction, validation of fixes, and effective resolution. This promotes a collaborative and efficient approach to vulnerability management, ultimately strengthening the overall security posture of the organization.