Limiting Attack Surface for Infrastructure Applications using Custom YAML Templates in Nuclei Automation

MSc Research Project
Cyber Security

Hardik Solanki
Student ID: x21117659

School of Computing
National College of Ireland

Supervisor: Prof. Vikas Sahni

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Hardik Solanki

Student ID: x21117659

Programme: MSc. In Cyber Security **Year:** 2022-2023

Module: MSc Research Project

Supervisor: Prof. Vikas Sahni

Submission Due Date: January 6th, 2023

Project Title: Limiting Attack Surface for Infrastructure Applications using Custom YAML Templates in Nuclei Automation

Word Count: 5483 **Page Count:** 20

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Hardik Solanki

Date: 6th, January 2023

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Limiting Attack Surface for Infrastructure Applications using Custom YAML Templates in Nuclei Automation

Hardik Solanki
x21117659

Abstract

With the growing connectivity of information systems around the world, as well as the accessibility of data resources, the Integrity and Confidentiality of Data and Services is also under threat. In order to reduce security risks, organizations use Vulnerability Assessment (VA), a technique for evaluating security threats. This condition can be achieved by automating and managing vulnerabilities with customized defined YAML templates. In this research, by developing a customized YAML template that includes all publicly available latest vulnerabilities including Zero Day attack related to Infrastructure applications and integrate into "Nuclei" Automated YAML based Scanner which is "Open source-A Community Powered Vulnerability Scanner" (Built by the Project Discovery Team). Furthermore, it has been used to run the "Nuclei" automated scanner on working infrastructure application for Vulnerability Assessment & Managing security Testing approach based on defined vulnerabilities in YAML templates which is developed in this research. Overall, this paper improves knowledge of security automation based on customized end points and assists in preventing, limiting attack surface, and monitoring Vulnerabilities Assessment (VA) without producing any false-positive results before any exploit happens in the Internal/External Infrastructure Applications.

Keywords *– Vulnerability Assessment; Vulnerability Management; Threats; Attack Surface; Security Audit; YAML Templates, Nuclei Automation Scanner; Automated Vulnerability scanning; Infrastructure Security.*

1 Introduction

In modern world, with almost everything being done online, Cyber Security has become more important than ever before (Ashwini Sheth et al., 2021). As a consequence, most of the daily activities are automated, so all that needs to is, minimize human interaction as much as possible and also, a similar statement can also be made about an organization's security posture. The majority of firms currently have teams in place that are in charge of prioritizing vulnerabilities and remediating or mitigating vulnerabilities within set deadlines. The teams in charge of remediation and mitigation are typically much bigger than the group in charge of prioritizing vulnerabilities. Because of this, it usually makes sense to begin by automating the work of the vulnerability management team while making as minimal changes as possible to the teams responsible for resolving and mitigating vulnerabilities. Specifically, will examine the advantages of Vulnerability Assessment Automation for Infrastructure Applications and their Security (Raydel Montesino & Stefan Fenz, 2011) and how this can be combined with Nuclei automation, which analyzes objects, loopholes, and vulnerabilities using YAML templates provided by users. By using YAML templates to guide requests to a large number of targets without producing false positives, Nuclei is an open-source, Golang-based tool designed to scan large numbers of targets quickly. There are a number of other protocols that Nuclei supports, including HTTP, TCP, DNS, HTTP, SSL, File, Whois, Websockets, Headless, Mobile, and Cloud Environment. Because of their robust and adaptable template, nuclei can be used as models for a variety of security checks. As nuclei scanner actually scans with user define templates, so in this paper will develop customized nuclei templates based on YAML Language that scans & monitor all the hosts that covers the top most recent attacks, vulnerabilities, CVE-ids/CWE, end-points, exposed panels, takeovers, misconfigurations, technologies, default-logins, etc. over the internet on a daily basis. Using nuclei automation, in this project, it can be ensured that every specified target and that all of these attack surface levels for Infrastructure application were removed automatically prior to the attack. In this manner, the vulnerability assessment, penetration testing, application security, and attack surface engagements were made more effective (Nuclei - Automated Vulnerability Scanner Tool | All About the Testing, 2022).

Figure 1: Nuclei Automation

1.1 Motivation:

A variety of threats can be found whenever using applications, databases, infrastructure, or products in daily life. Moreover, new vulnerabilities are discovered every day, such as Injection attacks, Cross-Site Scripting attacks, File Inclusion attack, Mobile based application attacks, Cloud based attacks and other sensitive files disclosure, including password files, configuration files, license files, server files, and more. This makes identifying vulnerable endpoints across all the hosts in the IT infrastructure is more difficult. Essentially, Nuclei Automation Scanner is capable of efficiently executing every infrastructure application while modifying the testing approach with a specific set of checks or any publicly vulnerable endpoint which user will specified in YAML templates and it is easy to incorporate nuclei in any procedure which involves continuous scanning, monitoring and managing the vulnerability surface (Project Discovery & nuclei engine, 2022). This way, adding automation to organization’s vulnerability management program will provide organization with multiple benefits like increased accuracy when prioritizing vulnerabilities, reducing the chance of human errors, increased compliance with SLAs, and greater efficiency for vulnerability management team (dw1, 2020).

- YAML based nuclei automation scanner tool is designed to integrate seamlessly as compare to other tools' workflow.
- The ability to process thousands of hosts in a short amount of time.
- Nuclei automate testing process in minutes using this easy custom YAML DSL.

1.2 Research Question:

How can the infrastructure application's attack surface be managed and automated using custom defined YAML templates in Nuclei Automation?

- Automate processes that cause as little disruption as feasible.
- Inspect your vulnerability management system to make sure it can give information without asking users to log in.
- Prioritize vulnerabilities automatically using risk rating
- Transparency on the outcomes, reminders, and escalations
- Select a system that is flexible enough to suit expanding needs.

1.3 Structure of the paper:

Section 2 includes a review of prior studies on the Automation Scanner and also explains what Nuclei Automation is, why Nuclei Automation, and the benefits of Nuclei Automation Scanner with Custom YAML Templates, stressing the advantages and disadvantages of every other vulnerability assessment/management scanning tool in section (2.1) and section (2.2) includes, related study on YAML and its syntax as well as why it is preferred comparatively to JSON and XML in Nuclei Automation Scanner. The suggested research approach for this study is presented in section 3. Section 4 is specification for the research design. Section 5 includes Implementation of research work in this research. Section 6, is Project Evaluation. Section 7, Conclusion & Future Work and in final section, acknowledgement and references that are useful for this study assignment.

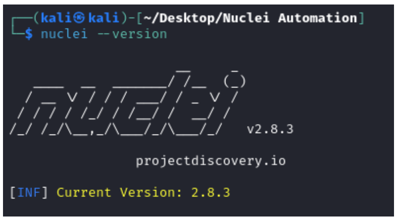
2 Related Work

This section showcases a detailed overview of the related papers. Based on numerous papers that have been published, most of the authors have proposed a variety of methods (Bairwa et al., 2014) for Vulnerability Assessment & Penetration Testing Security Automation for Application Security1 (Haibo Chen et al., 2020), the aspect of using different kind of Security Automation in this research for Vulnerability Assessment using Nuclei Automation Scanner based on Custom Defined vulnerable end point in YAML templates is to mitigate attack surface for all levels of vulnerabilities, including Zero Day attacks for entire Infrastructure Applications. This has been discussed below in detail.

As of date, this project has only ready-to-use YAML nuclei templates in the Nuclei publicly GitHub repository which were previously written and published by other authors. Furthermore, two security conferences were held during Bsides and IWCON in 2022 (Dhiyaneshwaran B., 2022) (Dhiyaneshwaran B., 2022), and no journals, IEEE papers, or white papers have been published on this topic over the internet. Thus, the research project will outline the use of a nuclei scanner, as well as workflows for defining custom YAML templates for Vulnerability Assessment/Management (Scanning) and Monitoring. Furthermore, this research project will define custom YAML templates based on publicly available CVE-IDs, open vulnerabilities, vulnerable systems, hidden sensitive files such as configuration files, database files, password files, etc. It involves infrastructure applications such as Web applications, Web servers, Mobile applications, Networks, Cloud environments/applications, Products, IOT, DNS, etc. As an additional feature, it can monitor the attack surface of production and staging infrastructure applications by scanning and verifying publicly disclosed vulnerabilities. It can be done by installing the Nuclei Automation Scanner Tool on Kali Linux using created custom YAML templates followed by Infrastructure domains list via Linux terminal commands. Research documentation, FAQs, and blogs are available for the templates that authors have already submitted to the Nuclei Team (Nuclei - Community Powered Vulnerability Scanner, 2022). The objective of this research is to develop and compare comparative studies on limiting and managing attack surfaces using Nuclei scanners with their custom YAML templates.

2.1 Nuclei Automation Scanner

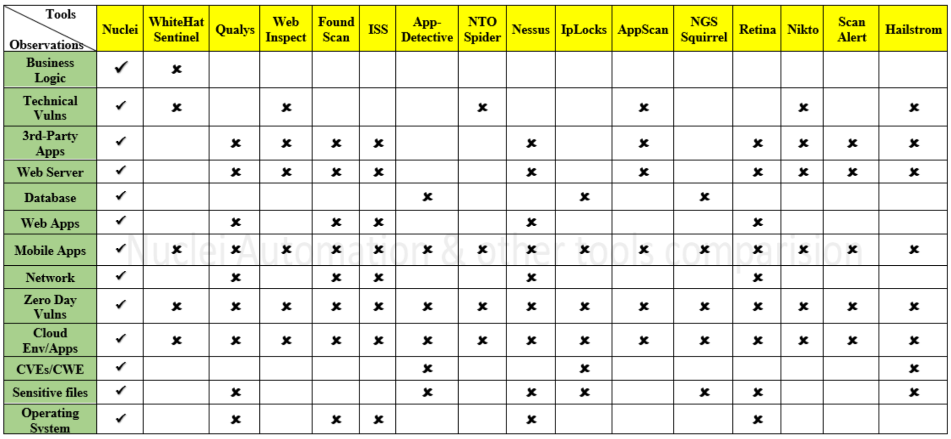
An analysis of the nuclei automation, which generates queries across target using a template and results in no false positives, and it is very quick to scan many hosts at the same time. A few of the protocols that Nuclei can scan is supporting are HTTP, TCP, DNS, HTTP, SSL, File, Whois, Websocket, and Headless, etc. Based on its robustness and adaptability, Nuclei Automation v2.8.3 show below in fig2. is capable of modeling a variety of security checks in this study. It also contains a repository containing over 350+ vulnerability templates, many of which have been created in this research and contributed to the Nuclei project to automate vulnerability assessments, management and monitoring the attack surface (Nuclei - Vulnerability Scanner, 2022).



```
┌──(kali㉿kali)-[~/Desktop/Nuclei Automation]
└─$ nuclei --version

                     __     _
   ____  __  _______/ /__  (_)
  / __ \/ / / / ___/ / _ \/ /
 / / / / /_/ / /__/ /  __/ /
/_/ /_/\__,_/\___/_/\___/_/   v2.8.3

                projectdiscovery.io

[INF] Current Version: 2.8.3
```

Figure 2: Nuclei Automation V2.8.3

In addition to these, there are various open-source and free solutions for assessing vulnerabilities that can be combined with different security platforms, security event information management (SIEM) solutions, or other security tools. While there are various technologies in the toolkit, choosing nuclei automation, which is based on unique YAML templates, addresses a proper security approach need. With Nuclei automation, based on custom developed YAML templates in this research, internal and external Vulnerability Scanning & Pentesting map out underlying technology and hunt for suspicious indicators on the first day of an engagement (Prevent Zero Day Attack). This drastically reduces the amount of time needed to identify vulnerabilities and eliminate false positives. In contrast to other security automation tools, nuclei automation scanner exactly does the same thing, based on YAML Custom templates defined in this research project. Since external pen testing (VAPT) is typically done within a scheduled window, it is important to have a tool that can hit all the targets accurately on the first day. Following Fig.2, examines a few of the tools that are accessible and shows why a custom YAML based Nuclei Scanner is the greatest tool available for this research project for quick Vulnerability Scanning (Jeremiah Grossman, 2006).



Tools / Observations	Nuclei	WhiteHat Sentinel	Qualys	Web Inspect	Found Scan	ISS	App-Detective	NTO Spider	Nessus	IpLocks	AppScan	NGS Squirrel	Retina	Nikto	Scan Alert	Hailstrom
Business Logic	✓	✗														
Technical Vulns	✓	✗		✗				✗			✗			✗		✗
3rd-Party Apps	✓		✗	✗	✗	✗			✗		✗		✗	✗	✗	✗
Web Server	✓		✗	✗	✗	✗			✗		✗		✗	✗	✗	✗
Database	✓						✗			✗		✗				
Web Apps	✓		✗		✗	✗			✗				✗			
Mobile Apps	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Network	✓		✗		✗	✗			✗				✗			
Zero Day Vulns	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Cloud Env/Apps	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
CVEs/CWE	✓						✗			✗						✗
Sensitive files	✓		✗				✗		✗	✗		✗	✗			✗
Operating System	✓		✗		✗	✗			✗				✗			

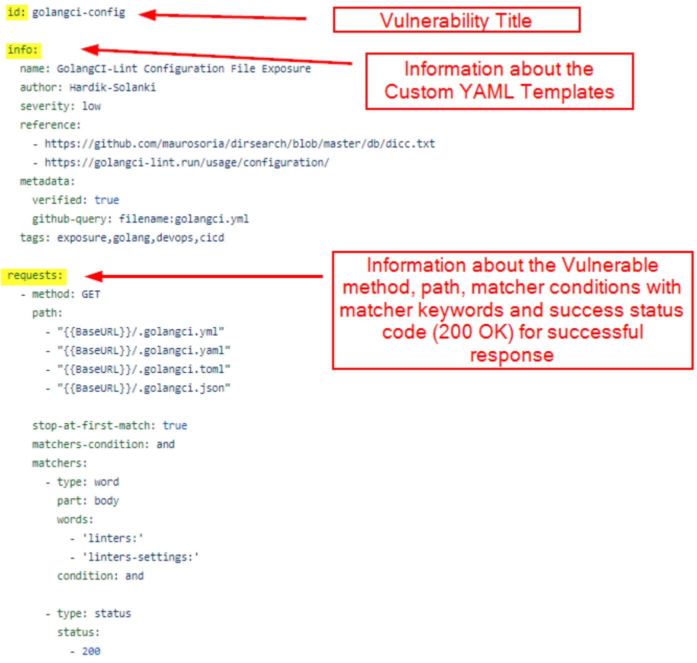
Nuclei Automation & other tools comparision

Figure 3: Vulnerability Assessment Tools Comparison Chart

2.2 Related work on YAML

An example table is provided in Table 1 YAML is a language for serializing data, commonly used for writing configuration files. As another markup language, YAML could also stand for yet another markup language in order to emphasize it's not for papers, but for data. In addition to Perl, C, XML, and HTML, YAML incorporates features from several programming languages. Since, YAML is a superset of the JSON, the YAML Language Development Team said that JSON files are compatible with the YAML. (YAML Language Development Team, 2021).

The only advantages of using YAML for nuclei templating is, its simpler to query comparative to JSON or XML as shown in below Fig 4. an example Fig 4. show the custom YAML template syntax which is developed for this research and Integrated in Nuclei Automation Scanner (Nuclei - Template, 2022).



```
id: golangci-config

info:
  name: GolangCI-Lint Configuration File Exposure
  author: Hardik-Solanki
  severity: low
  reference:
    - https://github.com/maurosoria/dirsearch/blob/master/db/dicc.txt
    - https://golangci-lint.run/usage/configuration/
  metadata:
    verified: true
    github-query: filename:golangci.yml
  tags: exposure,golang,devops,cicd

requests:
  - method: GET
    path:
      - "{{BaseURL}}/.golangci.yml"
      - "{{BaseURL}}/.golangci.yaml"
      - "{{BaseURL}}/.golangci.toml"
      - "{{BaseURL}}/.golangci.json"

    stop-at-first-match: true
    matchers-condition: and
    matchers:
      - type: word
        part: body
        words:
          - 'linters:'
          - 'linters-settings:'
        condition: and

      - type: status
        status:
          - 200
```

Vulnerability Title

Information about the Custom YAML Templates

Information about the Vulnerable method, path, matcher conditions with matcher keywords and success status code (200 OK) for successful response

Figure 4: Custom YAML Syntax developed for this research

3 Research Methodology

This research employs the agile development model because it places a strong emphasis on how risk and work are managed during each phase of the methodology. This includes the testing phase. This methodology was chosen because it ensures the flexibility of making changes quickly (Devharsh Trivedi, 2021), as well as having consistent evaluations throughout the agile process to determine whether the custom YAML template Integration with the expected outcomes requested by Nuclei Team and the Author of the Custom YAML Template achieved those objectives. It is most appropriate to use Agile methodology in order to create, implement, and integrate custom YAML templates into Production "Nuclei" Automation open-source Scanner.

A unique opportunity presented by an agile methodology allows the author to collaborate in all phases of the agile lifecycle. This includes the requirement phase to the deployment phase, which includes the testing phase for all newly developed custom YAML templates. In exchange for this additional benefit of transparency, the author must understand that the Nuclei Team (Project Discovery Team) are viewing a work in progress. Custom YAML Templates Integration in Nuclei Automation Scanner is used as a vulnerability solution to track, manage, prevent, and monitor the attack surface for infrastructure applications and its security because this model contains lots of risks and the agile model is capable of handling the risks (Bharat Choudhary and Shanu K Rakesh, 2016). The Agile methodology phases are depicted in below fig.5 (Charlie Belmer, 2022).

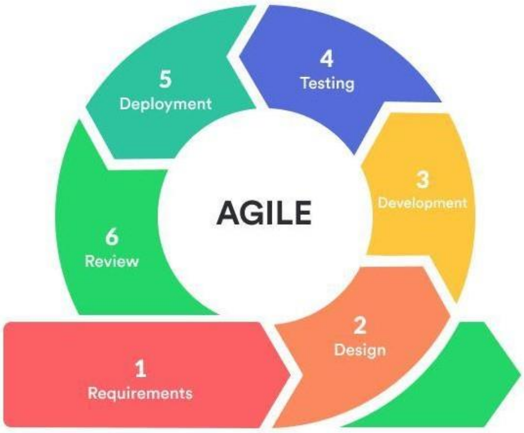


Figure 5: Agile methodology

According to the agile model, the following phases were followed in this research:

Requirements: In this research, to create custom YAML templates, the following sources are utilized to gather information and enumeration about publicly vulnerable end-points and latest vulnerabilities Impacting to Infrastructure Applications.

- Google Dork: An easy-to-miss piece of data can be retrieved using sophisticated search operators through a Google Dork query, often referred to as a dork. In addition, it can produce results for content that is difficult to locate with standard search terms.

- Shodan.io: Shodan is a search engine similar to Google. Google searches for webpages, while Shodan searches for internet-connected gadgets. In Shodan's search engine, authors can search for devices based on their IP addresses, displays names, or cities.
- OSINT: OSINT (Open-Source Intelligence) is a framework/tool that is used to gather intelligence from a wide range of sources. The data collected from OSINT can contribute to the better protection of organizations' networks and systems.
- Awesome-CVE-POC/WPScan/Mend: There is a Vulnerabilities Database, which contains Common Vulnerabilities and Exposures Proof of Concepts (CVE-ID/CWE).
- CVE Program: In order to identify, define, and catalog cybersecurity vulnerabilities publicly disclosed, CVE Program was created.
- Web path scanner: End-points for vulnerable sensitive directories and wordlists (This includes sensitive files such as configuration files, password files, database files, log files, etc.).

Design: The purpose of this stage is to design a custom YAML template using the information gathered during the requirements phase.

Development: Custom YAML templates are personally validated by template creators locally, who may then generate and submit a pull request to Nuclei Github for ongoing development.

Testing: Nuclei Team members and the YAML template's creator will test, examine, and verify customized templates created in this research. In case the templates are newly created, Nuclei Automation will integrate them into its production environment. In addition, Nuclei will not consider the author's request for execution because it has duplicate templates and will close the Pull request.

Additionally, a YAML Template Author and Nuclei Team Members should collaborate throughout this phase (Collaboration) until the Final Deployment in an Automated Nuclei Production Environment.

Deployment: Once the YAML templates have been validated and found to be accurate without any duplication, the template will be deployed in the Nuclei automation production environment by Nuclei Maintainers team.

Review & Launch: Upon integration and deployment, custom YAML templates are ready to be used publicly to manage attack surfaces, monitor vulnerabilities, and run automated vulnerability assessments.

4 Design Specification

This section describes the architectural design & process course for this research, as well as how the Custom YAML Templates workflow (dw1, 2020) has been integrated into open-source "Nuclei" Automation. As shown in figure 6, the custom YAML templates in this study have been reviewed, validated, and merged successfully using this execution workflow.

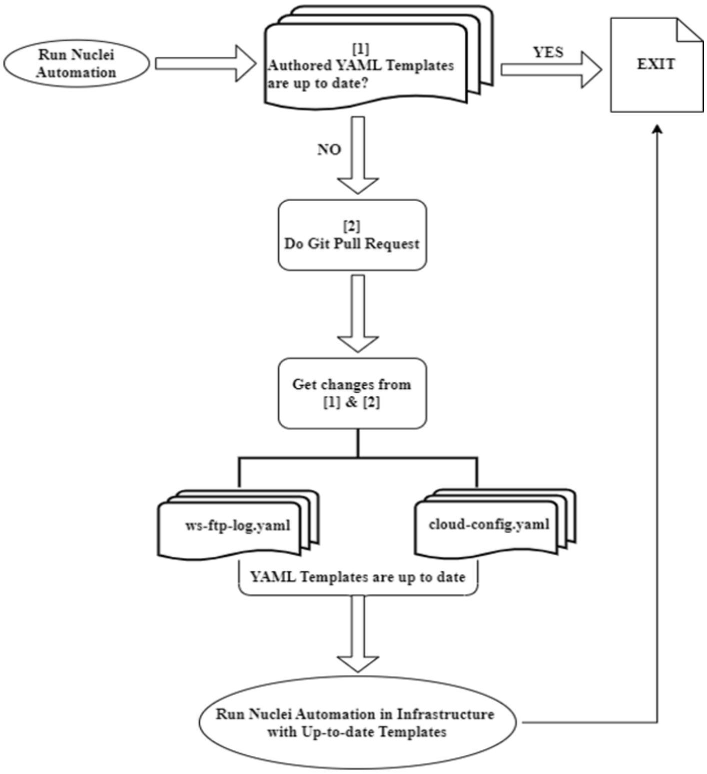

Run Nuclei Automation
[1] Authored YAML Templates are up to date?
YES
EXIT
NO
[2] Do Git Pull Request
Get changes from [1] & [2]
ws-ftp-log.yaml
cloud-config.yaml
YAML Templates are up to date
Run Nuclei Automation in Infrastructure with Up-to-date Templates

Figure 6: Custom YAML Template Execution Workflow

- The custom YAML template in Nuclei GitHub repository should always be up-to-date so that publicly disclosed vulnerabilities can be verified and validated against infrastructure using Nuclei Automation.
- In this research, author-designed custom YAML templates were based on publicly available vulnerabilities related to Infrastructure Applications and their end points.
- Based on the following HTTP Template conditions, YAML templates are designed and created in this research:

I. Base HTTP - Matches from multiple sources - Conditions-based matcher - Match conditions with multiple options - Headers with custom text - Requests for POST - Matchers based on time	**II. Workflow based Templates** - Workflows of a generic nature - Workflows based on conditional logic - The use of multi-condition workflows - A matcher-based conditional workflow - Workflow for multiple matchers
III. Headless Templates - Detection of pollution using a headless prototype - Reproduction of XSS with headless mode - Detection of DOM Cross-Site Scripting (XSS)	**IV. Headless Templates** - RAW requests in multiple batches

- Nuclei Automation scanners run on infrastructure based on custom-defined YAML templates. If respective YAML templates are current and present on GitHub, respective vulnerabilities are analyzed based on the severity of the respective vulnerable domain.

- When the YAML template is not up-to-date, the author needs to make updates by collaborating with both parties (the author & Nuclei Team).
- The templates will then be successfully merged into the production environment of Nuclei Automation Scanner when they are up-to-date, verified and validated by both the Author of the YAML templates and the Nuclei Team. The open-source tool will then be available for public use (since it's an open-source automation tool).

5 Implementation

This research aims to automate the vulnerability assessment process for infrastructure applications and to manage the attack surface by integrating Custom YAML templates into Nuclei Automation Scanner. The proposed design shown in below fig 7. for implementing YAML templates into Nuclei Automation Scanner and finding vulnerabilities into infrastructure has involved various steps for the implementation (Nuclei project maintainer Team, 2022):

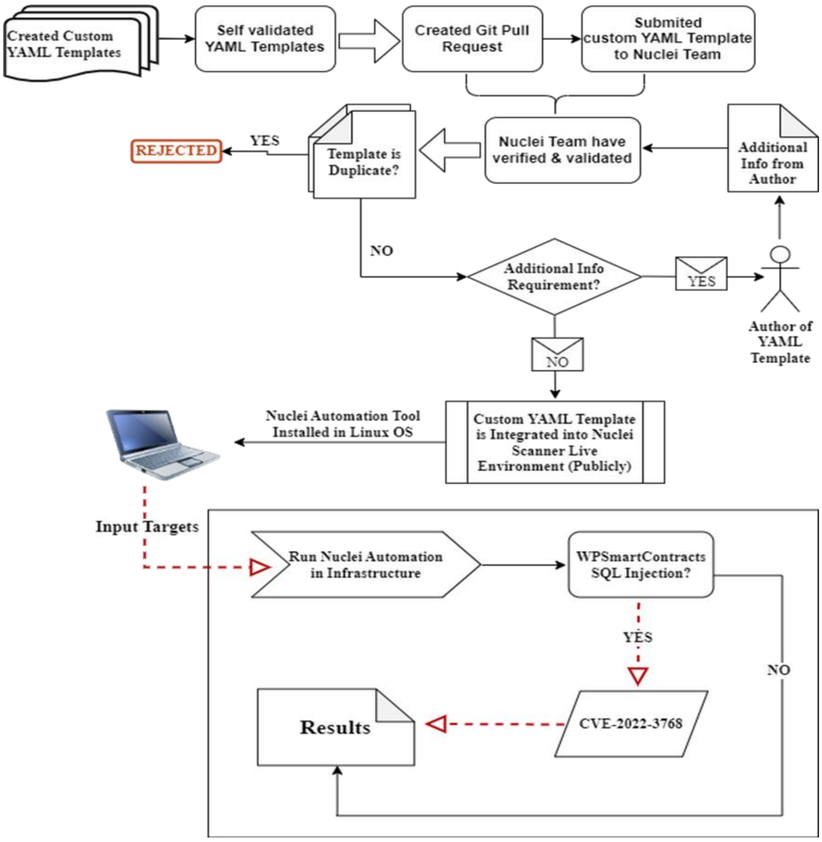

Created Custom YAML Templates
Self validated YAML Templates
Created Git Pull Request
Submited custom YAML Template to Nuclei Team
REJECTED
YES
Template is Duplicate?
Nuclei Team have verified & validated
Additional Info from Author
NO
Additional Info Requirement?
YES
Author of YAML Template
NO
Nuclei Automation Tool Installed in Linux OS
Custom YAML Template is Integrated into Nuclei Scanner Live Environment (Publicly)
Input Targets
Run Nuclei Automation in Infrastructure
WPSmartContracts SQL Injection?
YES
NO
Results
CVE-2022-3768

Figure 7: Custom YAML Templates Implementation in Nuclei Automation Scanner

Step 1: For this research, variety of Automation YAML templates have been designed, created, implemented, and integrated into Nuclei Automation Scanner, which is publicly accessible:

- YAML templates based on Web Application Vulnerabilities
- Mobile Application Vulnerabilities

- Network Application Vulnerabilities
- Cloud based vulnerabilities
- IOT Based vulnerabilities
- Latest CVE-2022 based
- Configuration file exposure
- Log files exposure
- Admin/hidden login Panel exposure

Step 2: The YAML Templates developed in this study have been self-verified, tested, and validated locally through the Nuclei Automation tool installed on Kali Linux via Linux Terminal.

Step 3: In case YAML templates have been verified, tested, and validated successfully without errors and are up-to-date from the author's end, then the template author must create a Git Pull Request. As a rule of thumb, a pull request is an event in Git where a YAML Template Author requests that a Nuclei Automation maintainer of a Git repository review a YAML Template that wants to be merged into a Nuclei project.

Step 4: The Nuclei Team has its own dedicated repository on GitHub named "projectdiscovery/nuclei-templates" where all the templates are being merged and integrated into the main repository. When the Nuclei feature branch is ready, the YAML templates author submits a pull request via their remote server account or forks Nuclei's main GitHub repository.

Step 5: In order to approve a pull request in Git, the Nuclei project maintainer(s) will review the work related to YAML templates. If the template author had not submitted a proper pull request and if the template was duplicate, the Nuclei project maintainer team will reject the YAML template with a Duplicate status and close the Git pull request. In addition, if YAML Template is created but Nuclei Maintainer team wants additional information such as, CVSS Scoring, Severity Clarification, or any other addition to syntax, etc., related to submitted YAML templates, then Nuclei maintainer team provides comments and here collaboration occurs between Author (Who provides the additional information) and Nuclei Project Maintainers team, or if additional information is not required due to proper, valid, and up-to-date YAML template, then git pull request is granted. As a result of this research, all the YAML templates are newly created and were approved on the git pull request, and they were merged directly into the main repository (Publicly Live) by Nuclei project team maintainers.

Step 6: Now, the YAML template created in this research is publicly available in the main repository of the Nuclei Automation project. As a result, the Nuclei Automation scanner tool has been installed locally on the Linux Operating System via Linux Terminal for Infrastructure Vulnerability Assessment scanning.

Step 7: Nuclei Automation tool has been installed in the Linux environment, now simply input the targets for the infrastructure application, where the test can be performed on one specific target or on "N" number of multiple targets specified in a text file.

Step 8: Now, run the Nuclei Automation into Infrastructure on specified targets with a targeted vulnerability YAML template. In this instance, the "WPSmartContracts SQL Injection" YAML template has been checked on specific targets. If there is no "WPSmartContracts SQL Injection" vulnerability detected in the target or detected vulnerable, the output result will be given straightforward without any false positives, including CVE-IDs, severity, and the vulnerable domains related to specific vulnerable YAML Templates.

Therefore, an Open-Source community driven automation tool called "Nuclei" has been implemented into working infrastructure for automating vulnerability assessments and, in addition to, monitoring, managing, and limiting the attack surface of the Infrastructure Applications before any publicly exploit or breach takes place in the infrastructure/organization.

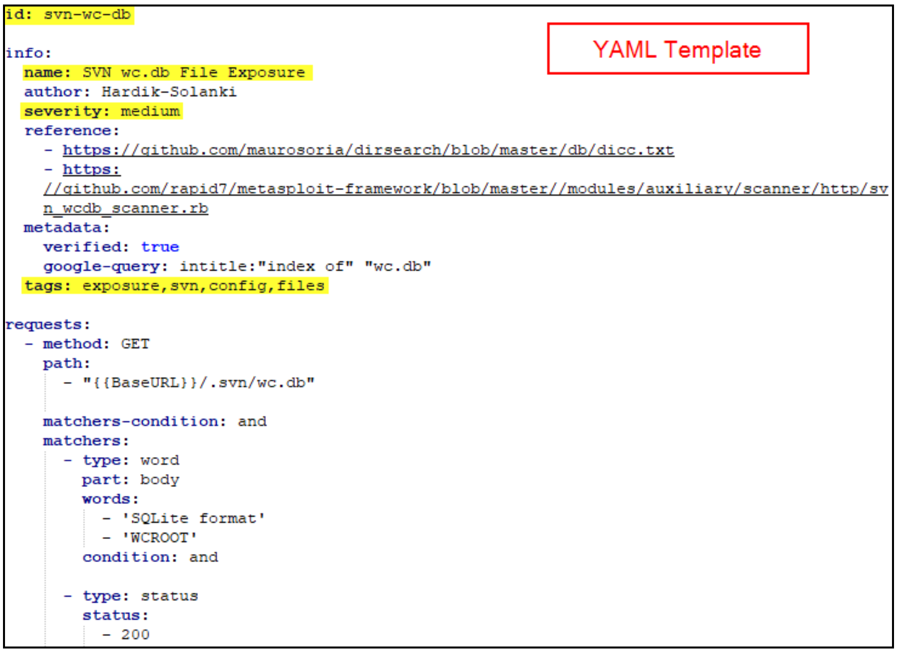
6 Evaluation

Evaluation of the implementation performed in the previous section is the focus of this section of the research paper. Nuclei Automation is being used to evaluate this by creating custom YAML templates, then using the tested output to determine the outcome and the success of the implementation.

The evaluation used the "Nuclei" Automation Tool, which was installed on Kali Linux operating systems, from which targets could automate the Vulnerability Assessment approach via Linux terminal, in combination with the corresponding “. yaml" templates, along with all infrastructure listed domains in a file named "Infrastructure-domains.txt".

6.1 Experiment 1 - “svn-wc-db.yaml” Template

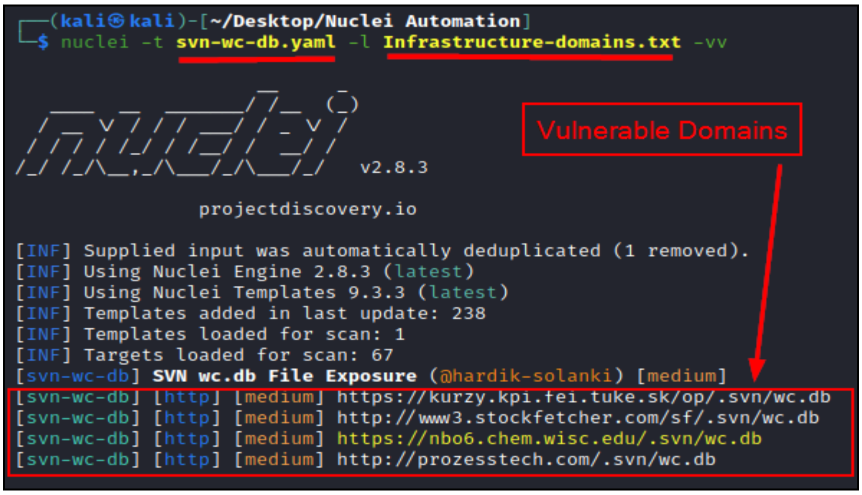
The following exhibit shows the "svn-wc-db.yaml" template that was created as part of this study. This template allows us to identify and scan for servers that allow access to the SVN wc.db file publicly:



```yaml
id: svn-wc-db

info:
  name: SVN wc.db File Exposure
  author: Hardik-Solanki
  severity: medium
  reference:
    - https://github.com/maurosoria/dirsearch/blob/master/db/dicc.txt
    - https:
    //github.com/rapid7/metasploit-framework/blob/master//modules/auxiliary/scanner/http/sv
    n_wcdb_scanner.rb
  metadata:
    verified: true
    google-query: intitle:"index of" "wc.db"
  tags: exposure,svn,config,files

requests:
  - method: GET
    path:
      - "{{BaseURL}}/.svn/wc.db"

    matchers-condition: and
    matchers:
      - type: word
        part: body
        words:
          - 'SQLite format'
          - 'WCROOT'
        condition: and

      - type: status
        status:
          - 200
```

Figure 8: Created YAML Template for "svn-wc-db"

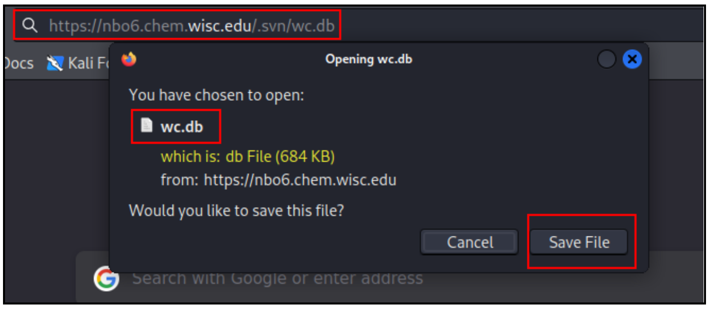
Using nuclei automation, followed by the respective "Infrastructure domains" list and YAML template "svn-wc-db.yaml". Out of 67 infrastructure targets, 4 were found to be vulnerable, exposing sensitive files publicly with a severity of “medium”. As shown in the illustration below.



```
┌──(kali㉿kali)-[~/Desktop/Nuclei Automation]
└─$ nuclei -t svn-wc-db.yaml -l Infrastructure-domains.txt -vv

                     __     _
   ____  __  _______/ /__  (_)
  / __ \/ / / / ___/ / _ \/ /
 / / / / /_/ / /__/ /  __/ /
/_/ /_/\__,_/\___/_/\___/_/   v2.8.3

                projectdiscovery.io

[INF] Supplied input was automatically deduplicated (1 removed).
[INF] Using Nuclei Engine 2.8.3 (latest)
[INF] Using Nuclei Templates 9.3.3 (latest)
[INF] Templates added in last update: 238
[INF] Templates loaded for scan: 1
[INF] Targets loaded for scan: 67
[svn-wc-db] SVN wc.db File Exposure (@hardik-solanki) [medium]
[svn-wc-db] [http] [medium] https://kurzy.kpi.fei.tuke.sk/op/.svn/wc.db
[svn-wc-db] [http] [medium] http://www3.stockfetcher.com/sf/.svn/wc.db
[svn-wc-db] [http] [medium] https://nbo6.chem.wisc.edu/.svn/wc.db
[svn-wc-db] [http] [medium] http://prozesstech.com/.svn/wc.db
```

Figure 8.1: Automation Output for “wc.db” with Vulnerable Domains

In the exhibits below, "wc.db" is also accessible, downloadable, and the content within the file can be read publicly without any restrictions. This proof that the YAML template for the "wc.db" created in this research has been successfully implemented into the Nuclei Automation Tool production environment.



https://nbo6.chem.wisc.edu/.svn/wc.db

Opening wc.db

You have chosen to open:

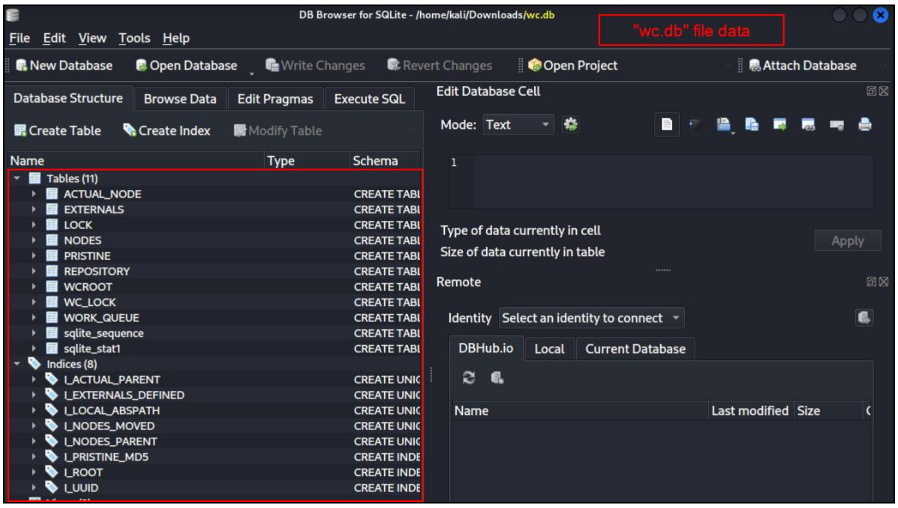
wc.db

which is: db File (684 KB)

from: https://nbo6.chem.wisc.edu

Would you like to save this file?

Figure 8.2: Sensitive file "wc.db" is accessible



DB Browser for SQLite - /home/kali/Downloads/wc.db

```
Name                        Type    Schema
Tables (11)
  ACTUAL_NODE                       CREATE TABL
  EXTERNALS                         CREATE TABL
  LOCK                              CREATE TABL
  NODES                             CREATE TABL
  PRISTINE                          CREATE TABL
  REPOSITORY                        CREATE TABL
  WCROOT                            CREATE TABL
  WC_LOCK                           CREATE TABL
  WORK_QUEUE                        CREATE TABL
  sqlite_sequence                   CREATE TABL
  sqlite_stat1                      CREATE TABL
Indices (8)
  I_ACTUAL_PARENT                   CREATE UNIC
  I_EXTERNALS_DEFINED               CREATE UNIC
  I_LOCAL_ABSPATH                   CREATE UNIC
  I_NODES_MOVED                     CREATE UNIC
  I_NODES_PARENT                    CREATE UNIC
  I_PRISTINE_MD5                    CREATE INDE
  I_ROOT                            CREATE INDE
  I_UUID                            CREATE INDE
```

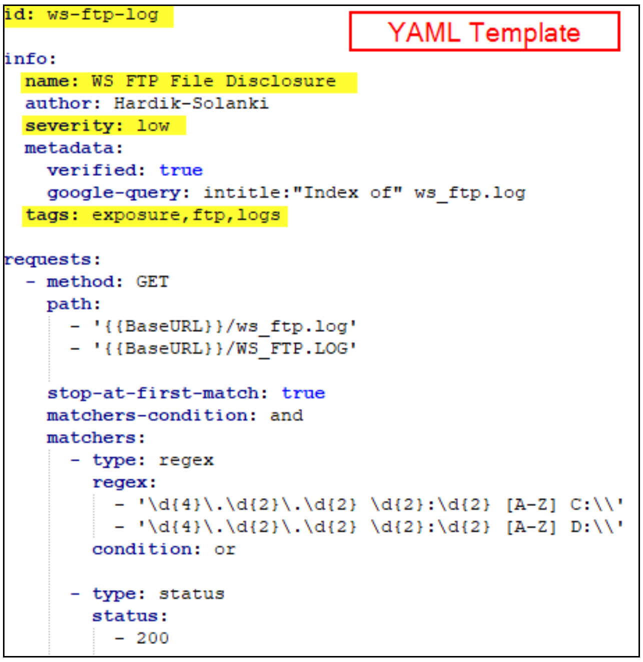
Mode: Text

Identity: Select an identity to connect

Figure 8.3: "wc.db" file content is accessible

6.2 Experiment 2 - “ws-ftp-log.yaml” Template

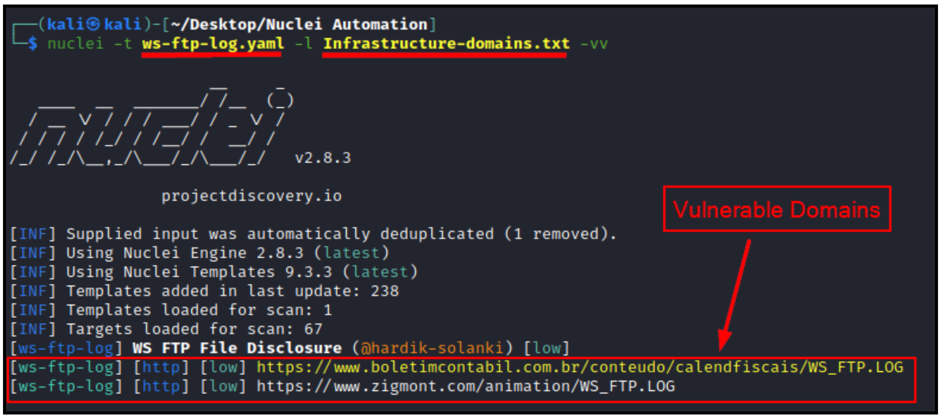
The following exhibit shows the "ws-ftp-log.yaml" template created in this research. This template allows to identify an WS_FTP log file, which is a popular FTP client. The template identifies a WS_FTP log file, which is a commonly used FTP client. This log file is named WS_FTP.LOG contains sensitive data such as file source/destination and file name, date/time of upload etc.



```yaml
id: ws-ftp-log

info:
  name: WS FTP File Disclosure
  author: Hardik-Solanki
  severity: low
  metadata:
    verified: true
    google-query: intitle:"Index of" ws_ftp.log
  tags: exposure,ftp,logs

requests:
  - method: GET
    path:
      - '{{BaseURL}}/ws_ftp.log'
      - '{{BaseURL}}/WS_FTP.LOG'

    stop-at-first-match: true
    matchers-condition: and
    matchers:
      - type: regex
        regex:
          - '\d{4}\.\d{2}\.\d{2} \d{2}:\d{2} [A-Z] C:\\'
          - '\d{4}\.\d{2}\.\d{2} \d{2}:\d{2} [A-Z] D:\\'
        condition: or

      - type: status
        status:
          - 200
```

Figure 9: Created YAML Template for "ws-ftp-log.yaml"

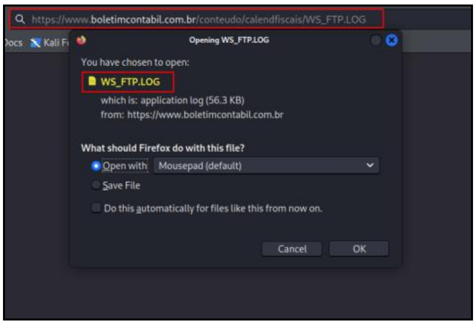
Using nuclei automation, followed by the Infrastructure domains list and YAML template, which is "ws-ftp-log.yaml". It was observed that out of 67 infrastructure targets, 2 were vulnerable, exposing sensitive information publicly with a severity of “low”. As shown in the illustration below.



```
┌──(kali㉿kali)-[~/Desktop/Nuclei Automation]
└─$ nuclei -t ws-ftp-log.yaml -l Infrastructure-domains.txt -vv

                     __     _
   ____  __  _______/ /__  (_)
  / __ \/ / / / ___/ / _ \/ /
 / / / / /_/ / /__/ /  __/ /
/_/ /_/\__,_/\___/_/\___/_/   v2.8.3

                projectdiscovery.io

[INF] Supplied input was automatically deduplicated (1 removed).
[INF] Using Nuclei Engine 2.8.3 (latest)
[INF] Using Nuclei Templates 9.3.3 (latest)
[INF] Templates added in last update: 238
[INF] Templates loaded for scan: 1
[INF] Targets loaded for scan: 67
[ws-ftp-log] WS FTP File Disclosure (@hardik-solanki) [low]
[ws-ftp-log] [http] [low] https://www.boletimcontabil.com.br/conteudo/calendfiscais/WS_FTP.LOG
[ws-ftp-log] [http] [low] https://www.zigmont.com/animation/WS_FTP.LOG
```

Figure 9.1: Automation Output for “ws-ftp-log” with Vulnerable Domains

The following exhibits demonstrate that "ws-ftp-log.yaml" is also accessible, downloadable, and the content in the file is readable by the public without any restriction. This proof that the YAML template for the "ws-ftp-log.yaml" created in this research has been successfully implemented into the Nuclei Automation Tool production environment.



https://www.boletimcontabil.com.br/conteudo/calendfiscais/WS_FTP.LOG

Opening WS_FTP.LOG

You have chosen to open:

WS_FTP.LOG

which is: application log (56.3 KB)
from: https://www.boletimcontabil.com.br

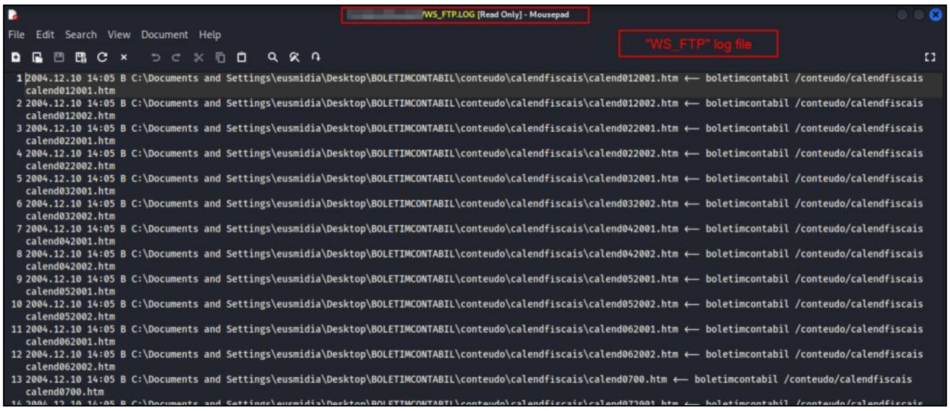
What should Firefox do with this file?

● Open with Mousepad (default)

○ Save File

□ Do this automatically for files like this from now on.

Figure 9.2: Sensitive file "WS_FTP.LOG" is accessible



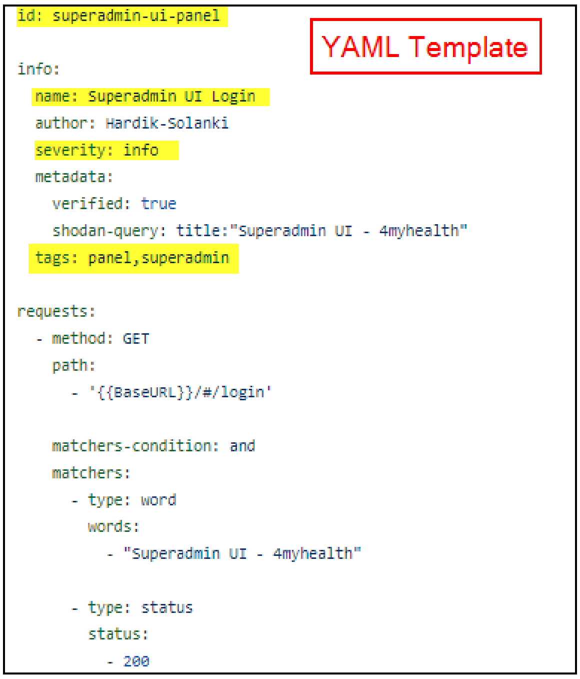
[redacted]/WS_FTP.LOG [Read Only] - Mousepad

```
2004.12.10 14:05 B C:\Documents and Settings\eusmidia\Desktop\BOLETIMCONTABIL\conteudo\calendfiscais\calend012001.htm ⟵ boletimcontabil /conteudo/calendfiscais
calend012001.htm
2004.12.10 14:05 B C:\Documents and Settings\eusmidia\Desktop\BOLETIMCONTABIL\conteudo\calendfiscais\calend012002.htm ⟵ boletimcontabil /conteudo/calendfiscais
calend012002.htm
2004.12.10 14:05 B C:\Documents and Settings\eusmidia\Desktop\BOLETIMCONTABIL\conteudo\calendfiscais\calend022001.htm ⟵ boletimcontabil /conteudo/calendfiscais
calend022001.htm
2004.12.10 14:05 B C:\Documents and Settings\eusmidia\Desktop\BOLETIMCONTABIL\conteudo\calendfiscais\calend022002.htm ⟵ boletimcontabil /conteudo/calendfiscais
calend022002.htm
2004.12.10 14:05 B C:\Documents and Settings\eusmidia\Desktop\BOLETIMCONTABIL\conteudo\calendfiscais\calend032001.htm ⟵ boletimcontabil /conteudo/calendfiscais
calend032001.htm
2004.12.10 14:05 B C:\Documents and Settings\eusmidia\Desktop\BOLETIMCONTABIL\conteudo\calendfiscais\calend032002.htm ⟵ boletimcontabil /conteudo/calendfiscais
calend032002.htm
2004.12.10 14:05 B C:\Documents and Settings\eusmidia\Desktop\BOLETIMCONTABIL\conteudo\calendfiscais\calend042001.htm ⟵ boletimcontabil /conteudo/calendfiscais
calend042001.htm
2004.12.10 14:05 B C:\Documents and Settings\eusmidia\Desktop\BOLETIMCONTABIL\conteudo\calendfiscais\calend042002.htm ⟵ boletimcontabil /conteudo/calendfiscais
calend042002.htm
2004.12.10 14:05 B C:\Documents and Settings\eusmidia\Desktop\BOLETIMCONTABIL\conteudo\calendfiscais\calend052001.htm ⟵ boletimcontabil /conteudo/calendfiscais
calend052001.htm
2004.12.10 14:05 B C:\Documents and Settings\eusmidia\Desktop\BOLETIMCONTABIL\conteudo\calendfiscais\calend052002.htm ⟵ boletimcontabil /conteudo/calendfiscais
calend052002.htm
2004.12.10 14:05 B C:\Documents and Settings\eusmidia\Desktop\BOLETIMCONTABIL\conteudo\calendfiscais\calend062001.htm ⟵ boletimcontabil /conteudo/calendfiscais
calend062001.htm
2004.12.10 14:05 B C:\Documents and Settings\eusmidia\Desktop\BOLETIMCONTABIL\conteudo\calendfiscais\calend062002.htm ⟵ boletimcontabil /conteudo/calendfiscais
calend062002.htm
2004.12.10 14:05 B C:\Documents and Settings\eusmidia\Desktop\BOLETIMCONTABIL\conteudo\calendfiscais\calend0700.htm ⟵ boletimcontabil /conteudo/calendfiscais
calend0700.htm
2004.12.10 14:05 B C:\Documents and Settings\eusmidia\Desktop\BOLETIMCONTABIL\conteudo\calendfiscais\calend072001.htm ⟵ boletimcontabil /conteudo/calendfiscais
```

Figure 9.3:"ws-ftp-log" file content is accessible

6.3 Experiment 3 - “superadmin-ui-panel.yaml” Template

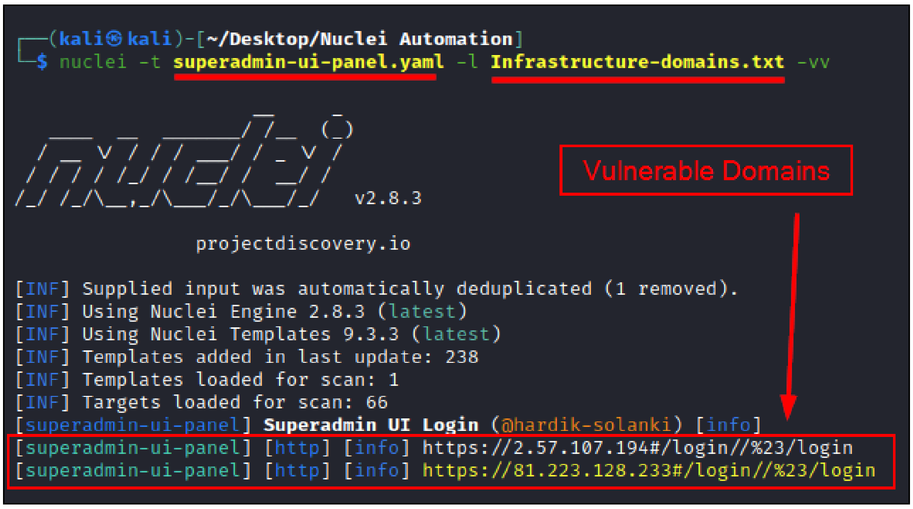
This exhibit shows the "superadmin-ui-panel.yaml" template that was created in this research. This template allows to identify an internal "Superadmin Login Panel" that is publicly visible.



```
id: superadmin-ui-panel

info:
  name: Superadmin UI Login
  author: Hardik-Solanki
  severity: info
  metadata:
    verified: true
    shodan-query: title:"Superadmin UI - 4myhealth"
  tags: panel,superadmin

requests:
  - method: GET
    path:
      - '{{BaseURL}}/#/login'

    matchers-condition: and
    matchers:
      - type: word
        words:
          - "Superadmin UI - 4myhealth"

      - type: status
        status:
          - 200
```

Figure 10: Created YAML Template for "superadmin-ui-panel.yaml"

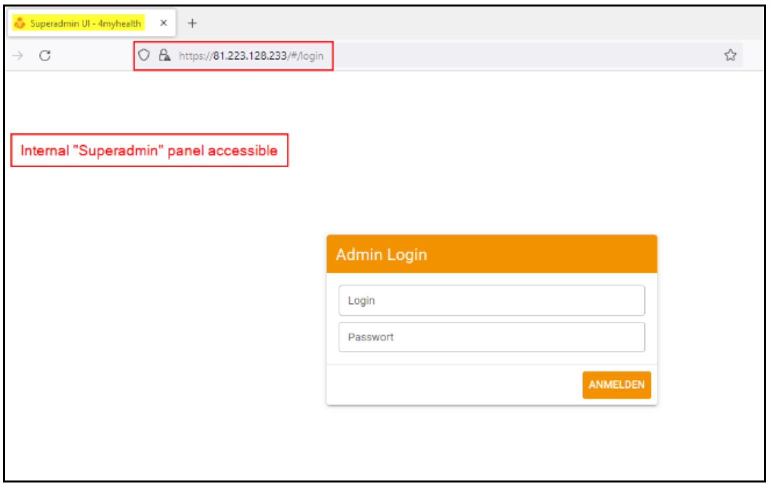
By using nuclei automation, followed by the "Infrastructure domains" list and YAML template, "superadmin-ui-panel.yaml". It was observed that, out of 67 Infrastructure targets, 2 were vulnerable and expose the “Superadmin Login Panel” publicly whose severity is “info”. As shown in the below exhibit.



```
┌──(kali㉿kali)-[~/Desktop/Nuclei Automation]
└─$ nuclei -t superadmin-ui-panel.yaml -l Infrastructure-domains.txt -vv

                     __     _
   ____  __  _______/ /__  (_)
  / __ \/ / / / ___/ / _ \/ /
 / / / / /_/ / /__/ /  __/ /
/_/ /_/\__,_/\___/_/\___/_/   v2.8.3

                projectdiscovery.io

[INF] Supplied input was automatically deduplicated (1 removed).
[INF] Using Nuclei Engine 2.8.3 (latest)
[INF] Using Nuclei Templates 9.3.3 (latest)
[INF] Templates added in last update: 238
[INF] Templates loaded for scan: 1
[INF] Targets loaded for scan: 66
[superadmin-ui-panel] Superadmin UI Login (@hardik-solanki) [info]
[superadmin-ui-panel] [http] [info] https://2.57.107.194#/login//%23/login
[superadmin-ui-panel] [http] [info] https://81.223.128.233#/login//%23/login
```

Figure 10.1: Automation Output for "superadmin-ui-panel" with Vulnerable Domains

In the exhibit below, the "Superadmin Login Panel" is also available to the public without restriction. This research proves that the created custom YAML template for "superadmin-ui-panel.yaml" is successfully implemented into Nuclei Automation Tool's production environment.



Superadmin UI - 4myhealth

https://81.223.128.233/#/login

Internal "Superadmin" panel accessible

Admin Login

Figure 10.2: Internal "Superadmin Login Panel” is accessible Publicly

6.4 Discussion

This research shows and performs several tests as part of these implementations and evaluations. These tests are to review the automation functionality of Nuclei's automation Scanner tool, which is based on customized YAML templates that were created in this research. In parallel, in this research, conducted an Automated Vulnerability Assessment and Management process in both a production environment and a UAT infrastructure environment. This process focused on evaluating the security capability, monitoring, managing, and limiting the attack surface of the organization's infrastructure. According to the findings, the test results offered insight into an automated risk assessment and this research states that the proposed solutions have significant potential to secure organizations infrastructure by Automating the Vulnerability Assessment and Management approach, which has been integrated in Nuclei Automation Tool, and eliminating the latest publicly available vulnerabilities including Zero Day attacks prior to any attack occurring into infrastructure. Moreover, these solutions are scalable, making them applicable to a wide range of use cases from small to large organizations performing Vulnerability Assessment, Penetration Testing, and Security Testing.

Also, one potential area of research surrounding the use of custom YAML templates in Nuclei Automation Scanner could be to explore the ways in which these templates can be used to improve the effectiveness and efficiency of vulnerability scans. This could involve studying how different types of templates (e.g., templates for specific vulnerabilities, templates for different types of systems, etc.) impact the accuracy and completeness of scans, as well as how well they scale to large or complex environments. Also, another area of research could be to examine the user experience of creating and using custom YAML templates in Nuclei. This could include studying the ease of use and flexibility of the template creation process, as well as how well the templates integrate with the overall workflow of the tool. Overall, the integration of custom YAML templates into Nuclei Automation Scanner offers many opportunities for research, and further study in this area could help to improve the capabilities and usefulness of the tool for users.

7 Conclusion and Future Work

In conclusion, it states that the main research question was successfully addressed by identifying a vulnerability, monitoring it, managing it, and limiting its attack surface within an organization using custom YAML templates defined in a Nuclei Automation scanning tool. On the basis of custom YAML templates, the Vulnerability Assessment Approach can be easily automated to detect the latest publicly available vulnerabilities into Infrastructure applications. Using a tool like Nuclei Automation Scanner can be an effective way to manage and reduce an organization's attack surface. By regularly scanning systems and identifying potential vulnerabilities, organizations can take steps to fix those vulnerabilities and make it more difficult for attackers to successfully compromise their systems. This, in turn, can help to reduce the risk of a successful cyber-attack and protect the organization's sensitive data and assets. A drawback in YAML template implementation in Nuclei Automation is, the created Git Pull Reject will be rejected by the Nuclei Maintainers Team if the submitted custom YAML templates were already created by another author.

One potential area of future work for Nuclei Automation Scanner based on custom YAML templates could be to expand the range of templates available to users. Currently, Nuclei provides a number of pre-built templates for common vulnerabilities, but there may be situations where users need to scan for more specialized or niche vulnerabilities. By allowing users to create and share their own custom YAML templates, Nuclei could make it easier for users to scan for these more specialized vulnerabilities and further improve their security posture. Another area of potential future work for Nuclei Automation Scanner could be to improve its scalability and performance. As organizations continue to grow and the volume of data and systems to be scanned increases, it may become necessary to optimize the tool to handle larger scale scans more efficiently. This could involve optimizing the algorithms used by the tool, improving the parallelization of scans, and implementing other performance enhancements. Overall, the scope of future work for Nuclei Automation Scanner based on custom YAML templates is quite broad, and there are many potential directions that development could take. The specific focus of future work will likely depend on the needs and feedback of users, as well as the priorities of the development team.

Acknowledgement

I wanted to take a moment for Prof Vikas Sahni to express my sincere gratitude and appreciation for all of his guidance and support throughout my master's research project. His expertise and insights have been invaluable to my growth as a researcher, and I am deeply grateful for the time and effort Prof Vikas Sahni have dedicated to my project. His consistent encouragement and constructive feedback have helped me to stay motivated and on track, and I am so grateful for the opportunity to work with such a knowledgeable and supportive supervisor. I feel confident that the skills and knowledge I have gained through this project will serve me well in my future endeavours, and I am deeply grateful for his role in helping me to achieve this personal and professional milestone.

References

Ashwini Sheth, Sachin Bhosale, & Farish Kurupkar. (2021). Research Paper on Cyber Security. CONTEMPORARY RESEARCH IN INDIA. https://www.researchgate.net/publication/352477690_Research_Paper_on_Cyber_Security

Bairwa, S., Mewara, B., & Gajrani, J. (2014). Vulnerability Scanners: A Proactive Approach to Assess Web Application Security. International Journal on Computational Science & Applications, 4(1), 113–124. https://doi.org/10.5121/IJCSA.2014.4111

Bharat Choudhary, & Shanu K Rakesh. (2016). An Approach using Agile Method for Software Development. IEEE Xplore. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7542304

Charlie Belmer. (2022). Integrating Security With Agile Development. Null Sweep. https://nullsweep.com/integrating-security-with-agile-development/

Devharsh Trivedi. (2021). Agile Methodologies. https://www.researchgate.net/publication/356924683_Agile_Methodologies#fullTextFileContent

Dhiyaneshwaran B. (2022). How to write your First Nuclei Template | Bsides Conference - 2022. Bsides Conference. https://bsidesahmedabad.in/2022/Schedule.html

Dhiyaneshwaran B. (2022, October). Security-Automation-(re)defined. IWCON (1st Edition). https://github.com/DhiyaneshGeek/My-Presentation-Slides/blob/main/slides/Security-Automation-(re)defined.pdf

dw1. (2020). Scan Continuously with Nuclei? | by dw1 | Medium. https://dwisiswant0.medium.com/how-to-scan-continuously-with-nuclei-fcb7e9d8b8b9

Haibo Chen, Junzuo Chen, Jinfu Chen, Shang Yin, Yiming Wu, & Jiaping Xu. (2020). An Automatic Vulnerability Scanner for Web Applications. IEEE . https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9343173

Jeremiah Grossman. (2006). Vulnerability Stack. https://blog.jeremiahgrossman.com/2006/11/vulnerability-stack.html

Nuclei - Automated Vulnerability Scanning Tool | All About Testing. (n.d.). Retrieved December 21, 2022, from https://allabouttesting.org/nuclei-automated-vulnerability-scanning-tool/

Nuclei - Community Powered Vulnerability Scanner. (2022). Project Discovery. https://nuclei.projectdiscovery.io/
Nuclei - Template. (2022). https://nuclei.projectdiscovery.io/templating-guide/#template-details

Nuclei - Vulnerability Scanner. (2022). https://nuclei.projectdiscovery.io/faq/nuclei/ Project Discovery, & nuclei engine. (2022). projectdiscovery/nuclei-templates: Community curated list of templates for the nuclei engine to find security vulnerabilities. https://github.com/projectdiscovery/nuclei-templates

Raydel Montesino, & Stefan Fenz. (2011). Information security automation. IEEE Xplore. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6045951

YAML Language Development Team. (2021, October). YAML Ain’t Markup Language (YAMLTM) revision 1.2.2. https://yaml.org/spec/1.2.2/

dw1. (2020). YAML Templates execution workflow in Nuclei Automation. https://dwisiswant0.medium.com/how-to-scan-continuously-with-nuclei-fcb7e9d8b8b9

Nuclei project maintainer Team. (2022). projectdiscovery/nuclei-templates: Community curated list of templates for the nuclei engine to find security vulnerabilities. Project Discovery. https://github.com/projectdiscovery/nuclei-templates