SECAUTO TOOLKIT - HARNESSING ANSIBLE FOR ADVANCED SECURITY AUTOMATION

Aakarsh Mavi (Corresponding Author)[1], Sanat Talwar[2]
[1]mavi.aakarsh4@gmail.com
[2]sanattalwar1994@gmail.com

Abstract

In tandem with the rapid evolution of technology, malicious actors are also evolving and adapting. Although putting new ideas into practice takes time and careful planning, attackers constantly take use of every resource at their disposal to compromise security. Since configuration vulnerabilities are more likely to occur in modern systems due to their increasing complexity, preventive measures are crucial.

It is in this situation that our Security Automation Toolset Framework becomes essential. Using automation, it finds environmental vulnerabilities and incorporates them into a governance model to fix problems as soon as they appear. An 8u research states that human mistake accounts for 95% of cybersecurity breaches, highlighting the significance of having such a toolbox. Our framework helps remain resilient in the face of changing threats by being simple to adopt, and able to log data for effective issue resolution.

A key component of our framework is the NIST Cybersecurity Framework, but first we need to understand what this framework is and how it integrates with our framework. The NIST Cybersecurity Framework helps businesses of all sizes better understand, manage, and reduce their cybersecurity risk and protect their networks and data. The Framework is voluntary. It gives your business an outline of best practices to help you decide where to focus your time and money for cybersecurity protection. You can put the NIST Cybersecurity Framework to work in your business in these five areas: Identify, Protect, Detect, Respond, and Recover.

Keywords *Cybersecurity, NIST, Ansible, Compliance, Automation, Playbooks, Incident Response, Risk Assessment.*

Introduction

In today's complex digital environments, organizations are tasked with securing a wide range of interconnected systems and data. Given the increasing sophistication of cyber threats and the growing number of endpoints and services, security teams are under pressure to respond faster and more efficiently. The quick and reliable answer to the issue at hand is automation, and Ansible will help us in creating this automation framework. Automation helps us by enabling timely and consistent application of security controls, system hardening, and incident response.

This research aims at exploring a way on how to integrate the NIST Cybersecurity Framework with Ansible automation to create a solution for security management proactively. While trying to achieve this, we aim to build a toolkit that identifies vulnerabilities in the systems and also fixes them by automated remediations. This all is being done in alignment with organizational governance, compliance standards, and risk management. This toolkit will help in building resilient infrastructures that are capable of responding to threats dynamically and fixing them as required. This paper primarily focuses on two core components of the security framework which are Governance Layer and Automation Layer. Before we deep dive into the paper, a basic understanding of both these is required.

Governance Layer - This is the layer in infrastructure that ensures compliance with security standards such as NIST and CIS.

Automation Layer - This layer enforces security measures like patching, system hardening and incident response.

Literature Review

How Existing Works Fall Short

- **Lack of Security Framework Integration**
  Many existing automation solutions address individual security tasks like patching, system compliance, or risk management needs. But fail to integrate these automations with comprehensive security frameworks like NIST. Without this integration, automations remain isolated, not addressing the broader picture.

- **Insufficient Emphasis on Governance**
  Governance is of utmost importance when it comes to security. With large-scale organizations, there’s usually a significant gap that exists in most automation tools. Governance should be a fundamental part of security automation rather than an afterthought. Some of the significant integral parts of governance include defining policies, ensuring regulatory compliance, and managing risk.

- **Inadequate Incident Response**
  Automation has always been used as a measure to reduce manual work of IT engineers and daily tasks like patching, but lesser attention has been paid to automated incident response. In real-world scenarios, vulnerabilities need to be fixed rapidly during active issues, but most solutions today lack dynamic response or don’t have a structured method for incident remediation.

- **Absence of Contextual Playbooks**
  Automation frameworks based on Ansible provide generic playbooks that do not meet the organization's needs for specific security requirements. This often leads to a lack of customization and inefficiencies, which might introduce new risk going forward.

Proposed Framework

The proposed framework integrates NIST and Ansible to build an effective, automated, and scalable security toolset. The framework is designed to align security automations with governance standards while automating the detection, response, and recovery processes.

Framework Design

Governance Layer Objectives

Governance Layer Objectives	NIST Framework	Category	Details
Defining objectives and standards for compliance	Identify	Governance	Defining compliance goals, cybersecurity policies, and protocols.
Mapping Ansible tasks to NIST Framework	Identify	Risk Assessment	Use Ansible to convert compliance requirements into automated controls.
Define policies (e.g., password policies, logging)	Protect	Access Control	Automating the enforcement of system configurations and access management.
Set benchmarks for the environment	Protect	Information Protection	Synchronizing security automation with standards for compliance.

Automation Layer Objectives

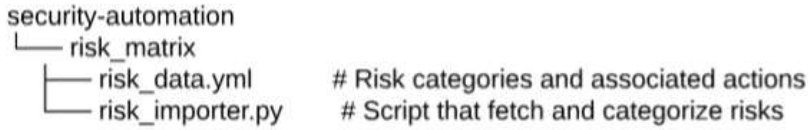
Automation Layer Objectives	NIST Framework	Category	Details
Enforce benchmarks using Ansible	Protect	Information Protection	Automating configuration, encryption, and patching processes related to compliance.
Automate patching	Protect	Maintenance	Automating system patching on a regular basis.
Create incident response playbooks	Respond	Response Planning	Automating remedial tasks like system isolation or account lockouts.
Log actions performed by automated tasks	Detect	Continuous Monitoring	Log and monitor tasks with Ansible to integrate with existing monitoring tools.
Automate recovery tasks	Recover	Recovery Planning	Automating the recovery of systems and configurations after an event.

Implementation

In the implementation we are going to define all the components that are required for this and have a tree structure of Ansible automation that can be used in this toolkit.

1. Risk Assessment and Categorization

This component of the framework toolkit first checks the vulnerabilities in the environment and puts them into categories based on the nature of the vulnerability (high, medium, or low). This is achieved by utilizing various sources such as the CVE database or real-time threat intelligence.



```
security-automation
└── risk_matrix
    ├── risk_data.yml        # Risk categories and associated actions
    └── risk_importer.py      # Script that fetch and categorize risks
```

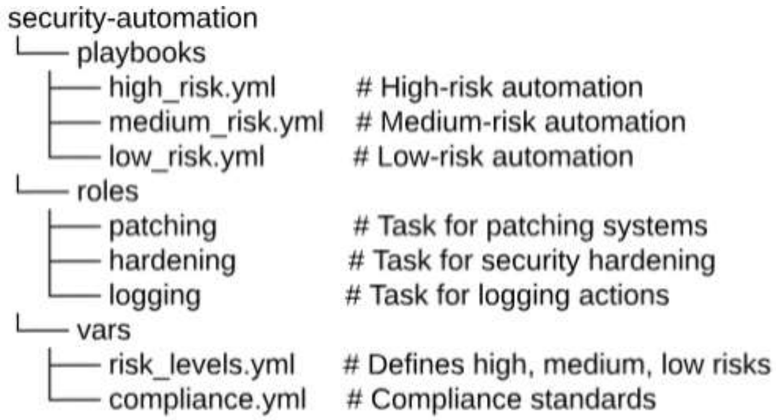
How it works:

- The process starts by initiating the risk_importer.py script that fetches CVE or security incident data using an API to get risk information; this gets info on CVSS scores and affected systems.
- Risk is then classified into high, medium, or low risk categories based on the data gathered.

- Once this information is gathered, risk_data.yml comes into action that maps risk level to appropriate actions such as patching or any other required actions based on the threat.

2. Defining Ansible Automation File Structure

Once the risks are identified and categorized accordingly, the next step is to establish the structure for automation tasks that need to be performed. This includes playbooks, roles, and variables that are required for remediation.



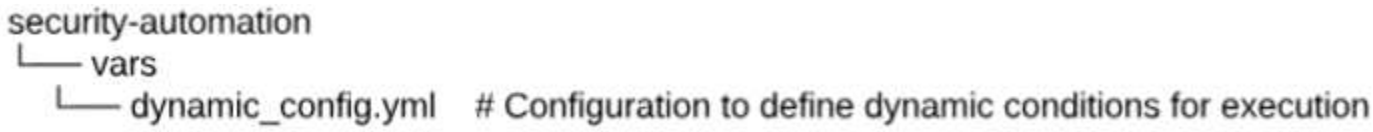
```
security-automation
└── playbooks
    ├── high_risk.yml       # High-risk automation
    ├── medium_risk.yml     # Medium-risk automation
    └── low_risk.yml        # Low-risk automation
└── roles
    ├── patching            # Task for patching systems
    ├── hardening           # Task for security hardening
    └── logging             # Task for logging actions
└── vars
    ├── risk_levels.yml     # Defines high, medium, low risks
    └── compliance.yml      # Compliance standards
```

How it works:

- **Playbooks:** Tasks to be performed are grouped in specific YAML playbooks based on the risk (patching for high risk, scheduled patching for medium risk, and enabling monitoring for low risk).
- **Roles:** Reusable roles are created that could be used across multiple playbooks.
- **Variables:** Risk levels are mapped to specific actions, compliance checks, and system configurations using the variable playbook.

3. Dynamic Task Execution Using Conditions

To make sure the framework is adaptable to varied environments and situations, tasks will be executed conditionally based on parameters, risk levels, and environment specifics.



```
security-automation
└── vars
    └── dynamic_config.yml    # Configuration to define dynamic conditions for execution
```

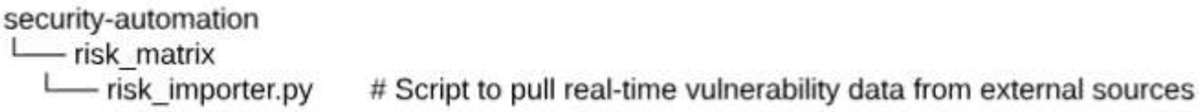
How it works:

- The dynamic_config.yml file includes conditions like the system's state, such as its patching status and uptime.

- Ansible tasks will adapt based on these conditions. For example, patching tasks will run only if the system isn't already patched, and services will restart only if the patching succeeds.

4. Using External Data Sources

Integrating external data sources like threat intelligence platforms or vulnerability databases can enhance the framework by updating risk levels and guiding decision-making.



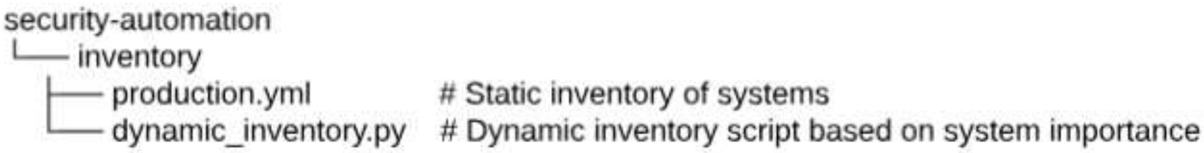
```
security-automation
└── risk_matrix
    └── risk_importer.py      # Script to pull real-time vulnerability data from external sources
```

How it works:

- **External Data Integration**: The **risk_importer.py** script retrieves live data from outside threat intelligence sources (such as CVE databases).
- This script processes the information to automatically refresh risk levels and categorize vulnerabilities.

5. Prioritize Systems Dynamically

After assessing the risks, prioritize systems for remediation based on their exposure, significance, and critical role in the organization.



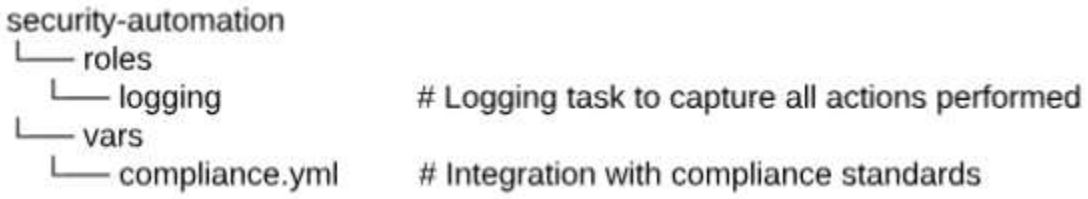
```
security-automation
└── inventory
    ├── production.yml          # Static inventory of systems
    └── dynamic_inventory.py    # Dynamic inventory script based on system importance
```

How it works:

- **Dynamic Inventory**: The **dynamic_inventory.py** script retrieves system data based on their **criticality** and **exposure**. It updates the inventory list to focus on systems needing urgent attention (for instance, a system accessible from the internet may be prioritized over one that is internal).
- Tasks will be automated according to the system priority.

6. Logging Actions and Auditing

All actions taken by the automation framework must be logged for auditing to ensure compliance and governance, particularly for organizations following standards like NIST.



```
security-automation
└── roles
    └── logging          # Logging task to capture all actions performed
└── vars
    └── compliance.yml   # Integration with compliance standards
```

How it works:

- The **logging/** role records all actions performed by the automation framework (like applying patches and restarting services) into a central log file.
- These logs serve **audit** needs and ensure compliance with **NIST** or other applicable standards.

Conclusion

The Security Automation Toolset Framework offers a methodical way to handle vulnerabilities in an organization, adhering to industry standards like NIST. By combining automation with a governance framework, it presents several important benefits.

- **Standard Governance:** This framework ensures that all security tasks and processes align with well-defined governance goals, ensuring compliance with set standards. By connecting security requirements to tools like Ansible and utilizing frameworks like NIST, organizations can implement uniform and repeatable security practices. This lowers the chances of errors, misconfigurations, and non-compliance, ultimately enhancing their security posture.
- **Enhanced Efficiency:** Using Ansible playbooks to automate security tasks such as patching, system hardening, and incident response greatly minimizes manual work and speeds up response times. Tasks that used to take hours or even days can now be completed in much less time, allowing security teams to concentrate on more intricate issues. The capacity to dynamically adapt to varying risk levels and system priorities further boosts operational efficiency, enabling organizations to swiftly address critical vulnerabilities. A research paper recently came out, "AN OVERVIEW OF DNS DOMAINS/SUBDOMAINS VULNERABILITIES SCORING FRAMEWORK," that can be integrated into our framework as well, making it more robust and advanced. We can look into vulnerabilities of DNS and, as per the scoring, create automated remediations for the same.
- **Improved Compliance:** Adhering to regulatory standards like NIST, CIS, and others is a central aspect of the framework. By automating tasks and incorporating compliance checks, organizations consistently meet regulatory requirements. The framework's features allow for action logging and compliance auditing, providing a thorough method to maintain a compliant and accountable environment, which serves as evidence of due diligence during an audit.
- **Dynamic Risk Management:** This framework assesses risks in real-time and responds effectively, representing a major advancement over traditional static methods. It utilizes real-time data from external sources and continuously reviews the risk environment. By organizing vulnerabilities and prioritizing remedial efforts based on their severity and the importance of the systems involved, organizations can take proactive steps to mitigate risks, ensuring that the most pressing issues are tackled first. To tackle things more dynamically we can again look at the earlier mentioned in the

Enhanced Efficiency conclusion, the DNS framework helps us in more ways to automate the issue in DNS. The author of the paper went ahead and published another research on S3-linked domains. Similarly, we can get into the continuous phase of evolvement, and once the scoring frameworks are integrated, a broader aspect of security, the cloud-based entities can be included as well.

- **Future Realization and Full Automation**: Although the current model effectively automates key security tasks, it can be expanded to completely automate the entire security lifecycle. Future versions might include advanced machine learning for predicting risks, detailed playbooks for various security scenarios, and integration with other automation tools to create a comprehensive security management system. This all-encompassing automation would enable organizations to proactively address emerging threats and maintain continuous, real-time management of their security posture.

Reference

1. Nist Cybersecurity Framework - Https://Www.Ftc.Gov/Business-Guidance/Small-Businesses/Cybersecurity/Nist-Framework
2. Aakarsh Mavi, Sanat Talwar, "An Overview Of Dns Domains/Subdomains Vulnerabilities Scoring Framework", International Journal Of Applied Engineering & Technology. 15, S4 (2023). Https://Romanpub.Com/Resources/Vol.%205%20no.%20s4%20(July%20-%20aug%202023)%20-%2027.Pdf
3. 8u. (N.D.). "Cybersecurity Statistics - Https://8u.Com/Cybersecurity-Statistics
4. Nist. (2018). "Framework For Improving Critical Infrastructure Cybersecurity." National Institute Of Standards And Technology, U.S. Department Of Commerce - Https://Www.Nist.Gov/Cyberframework
5. Red Hat. (N.D.). "Ansible Documentation" - Https://Docs.Ansible.Com/
6. Cve. (N.D.). "Cve - Common Vulnerabilities And Exposures." Https://Cve.Mitre.Org/
7. Cis. (2020). "Cis Controls." Center For Internet Security. Https://Www.Cisecurity.Org/
8. Nvd. (2020). "National Vulnerability Database." National Institute Of Standards And Technology. Https://Nvlpubs.Nist.Gov/Nistpubs/
9. Gartner. (2021). "Magic Quadrant For Security Information And Event Management." Https://Www.Gartner.Com/En/Documents
10. Sans Institute. (2019). "Incident Handling And Response." Https://Www.Sans.Org/Cyber-Security-Courses
11. Sanat Talwar, "Securing Cloud-Native Dns Configurations: Automated Detection Of Vulnerable S3-Linked Subdomains", International Journal Of Applied Engineering & Technology. 4,2 (2022). Https://Romanpub.Com/Resources/Vol.%204%20no.%202%20(September%2c%202022)%20-%2033.Pdf
12. Sanat Talwar, Aakarsh Mavi, "An Overview Of Dns Domains/Subdomains Vulnerabilities Scoring Framework", International Journal Of Applied Engineering & Technology. 15, S4 (2023). Https://Romanpub.Com/Resources/Vol.%205%20no.%20s4%20(July%20-%20aug%202023)%20-%2027.Pdf