UDC: 004.056.5, 004.7, 004.056.53 https://doi.org/10.20535/2786-8729.6.2025.334139

AUTOMATIC NETWORK RECONFIGURATION METHOD WITH DYNAMIC IP ADDRESS MANAGEMENT

Anatolii Haidai *
National Technical University of Ukraine
“Igor Sikorsky Kyiv Polytechnic Institute”, Kyiv, Ukraine
http://orcid.org/0000-0001-9330-414X

Iryna Klymenko
National Technical University of Ukraine
“Igor Sikorsky Kyiv Polytechnic Institute”, Kyiv, Ukraine
http://orcid.org/0000-0001-5345-8806

*Corresponding author: tolya.hei@gmail.com

In the context of growing cyber threats, systems capable not only of detecting anomalies in the operation of network infrastructure but also of promptly responding to them without administrator intervention are becoming increasingly relevant. This paper proposes a method for automatic network reconfiguration based on the integration of the *Zabbix* monitoring system with the *pfSense* network gateway functionality. Such a system enables centralized control of the operating system status, resource usage, and network activity, while also allowing for automatic changes to host IP addresses, routing adaptation, and connection restrictions according to defined security policies.

The aim of the study is to develop a method for automatic network monitoring and reconfiguration with dynamic IP address changes to improve the effectiveness of cyber threat mitigation. The object of the study is the processes of information security management in computer networks. The subject of the study includes methods of anomaly detection and automatic response through modification of network parameters using *Zabbix* and *pfSense*.

In the context of automatic response to detected threats, the method of comprehensive monitoring of client host operating systems has been formalized, including subsequent analysis of logs, user actions, resource load, network port usage, and interaction with external services. A methodology for network reconfiguration after anomaly detection has been developed and implemented: in particular, changing the IP address while maintaining functionality in a minimal network access configuration and isolating the node using *pfSense*. Scripts for Windows client OS were employed, interacting with the *Zabbix* and *pfSense* APIs, thus ensuring dynamic and fully automated operation.

Testing results of the proposed system in a simulated environment confirm its effectiveness. Compared to manual or partially automated solutions, incident response time was reduced, and the risk of attack propagation within the network was minimized.

Keywords: *Zabbix*, *pfSense*, network security, anomaly detection, automated response.

1. Introduction

In the current context of rapid digital technology development and the increasing dependence of society on computerized systems, the issue of ensuring information security is becoming particularly critical. New threats emerge daily – from automated attacks using malicious software to targeted intrusions employing social engineering, phishing, or software vulnerabilities. In response, there is a growing demand for monitoring systems capable not only of detecting anomalies but also of automatically responding to them in real time.

Existing solutions implement effective approaches for multipurpose monitoring of information infrastructure using the *Zabbix* monitoring system. For example, approaches to monitoring IoT infrastructures using *Zabbix* were investigated on a prototype integrating Arduino and Raspberry Pi devices [1]. In the studies [2], the performance of *Zabbix* was comprehensively analyzed in high-load infrastructural environments built on network servers, where there is an urgent need for effective monitoring of hardware and software resources. In [3], it was demonstrated that the combination of network monitoring based on *Zabbix* with *remote control* via mobile devices ensures efficient dissemination of monitoring information and real-time alert notifications.

New opportunities arise from combining the functionality of *Zabbix* with the capabilities of the *pfSense* network gateway. This integration makes it possible to effectively address modern information security challenges, in particular by not only centrally collecting data on operating system status, resource usage, and network activity, but also automating changes in the network structure, such as route reconfiguration and IP address modification on client hosts.

The methods and practical approaches discussed in this paper are aimed at solving tasks related to the prevention of unauthorized access to systems and their control, using *Zabbix tools*, *pfSense*, and scripts for the Windows operating system. Exploring the implementation of such solutions, their advantages, and their practical application in modern information systems is a relevant topic of high scientific and practical value.

2. Literature review and problem statement

In today's digital world, information is not only a resource but also a strategic asset. The computer environment in which it is stored, processed, and transmitted is constantly under threat due to the increasing number and complexity of cyberattacks. As evidenced by a large number of current cybersecurity studies [4], there is growing interest in innovative approaches to ensuring key aspects of cyber protection, such as threat detection and prevention, attack mitigation, incident response, and risk assessment and management.

In [5], the authors describe an approach that uses artificial intelligence to detect and respond to cyber threats in cloud environments. Although the authors demonstrate that their trained models can detect access attempts or unauthorized interactions, they note that the models need to be tested on large-scale practical datasets in real cloud environments. A high detection rate was achieved on the given dataset; however, due to the rapid evolution and adaptation of other Artificial Intelligence (AI) models to enhance cyberattacks, existing datasets may cover only a limited portion of known threats and attack attempts.

The cybersecurity challenges in Industrial Control Systems (ICS), which have become accessible via the Internet due to technological advances, are addressed in [6]. This openness introduces numerous vulnerabilities that can be exploited by attackers to infiltrate and disrupt the operation of critical infrastructure, such as power plants or water treatment facilities. The authors point out that traditional threat detection in ICS is typically performed manually, requiring expert involvement and suffering from scalability and efficiency limitations during fast-paced attacks. To address this, the authors propose an automated threat hunting framework based on the MITRE ATT&CK (Adversarial Tactics, Techniques, and Common Knowledge) for ICS knowledge base. This base contains structured information about adversary Tactics, Techniques and Procedures (TTPs), enabling the formulation of attack hypotheses. The solution leverages tools such as *Elastic Stack, Kibana, Metasploit*, and others to collect, analyze, and visualize data in real time. The proposed system significantly reduces manual labor, increases threat detection speed and accuracy, and creates a flexible, repeatable environment for testing attack scenarios.

In the modern digital landscape, cyber threats are becoming increasingly complex and sophisticated. Traditional protection methods based on signatures or manual rules no longer provide an adequate level of security. For this reason, AI and Machine Learning (ML) are gaining particular relevance as tools for proactive, adaptive, and automated threat detection and response.

In [7], the authors thoroughly examine the main ML methods applied in cybersecurity. Supervised learning is used to classify malware or detect spam. Unsupervised learning enables the detection of anomalies in large datasets without prior labeling. Reinforcement learning is used to develop systems that autonomously adapt to changes in the threat environment. Among the most commonly used algorithms are decision trees, neural networks, Support Vector Machines (SVM), Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN). The authors demonstrate how these methods are effectively used in various areas: malware detection, network intrusion monitoring, fraud detection, and User Behavior Analytics (UBA). For example, *Darktrace* uses unsupervised learning to build a profile of “normal” network activity, while *SparkCognition’s DeepArmor* system uses deep learning to identify new types of malware.

Despite significant achievements, the implementation of AI in cybersecurity presents several challenges. These include the need for large volumes of high-quality labeled data, the threat of adversarial attacks on ML models, the difficulty of interpreting model decisions (especially deep neural networks), and the problem of concept drift. Scalability and integration of such solutions into real IT infrastructures also remain critical issues.

As noted in [5, 7], AI and ML are expected to play an increasingly important role in cybersecurity. Promising research directions include the development of explainable models, protection against adversarial ML attacks, the use of transfer learning for small datasets, and the evolution of autonomous, self-learning security systems. The integration of such solutions with automated response platforms will enable the creation of more reliable and adaptive digital defenses.

Current issues and threats in the field of information and computer environment protection are further presented as a structured overview of information security threats and the associated protection challenges.

Key Information Security Issues [8]:

1. Insufficient automation of monitoring, lack of around-the-clock supervision of infrastructure, and absence of rapid initial analysis using at least monitoring systems;
2. Human factor – low awareness of threats and failure to follow basic security recommendations, such as filtering suspicious emails or regularly changing passwords;
3. Outdated or vulnerable software, lack of consistent updates or patching for known vulnerabilities;
4. Network infrastructure vulnerabilities – due to poor segmentation and lack of firewall-based restrictions;
5. Insufficient control over external connections, open ports, and poorly configured services.

Main Types of Information Security Threats:

1. *Malware* [7–9] – viruses, trojans. They may create backdoors, steal data, or encrypt files for ransom (ransomware);
2. *Phishing and Social Engineering* [5, 7–9] – attacks that exploit users’ psychological vulnerabilities via email, social media, or phone calls;
3. *Advanced Persistent Threats* (*APTs*) [5, 7–9]:
   - Long-term and complex attack campaigns, often state-sponsored or conducted by large organizations;
   - Involve reconnaissance, system profiling, stealthy penetration, and prolonged presence in the target environment.
4. *Insider Threats* [5, 8–9] – employees or former staff members with access to critical data who misuse it for personal gain or revenge;
5. *Zero-Day Attacks* [5, 8–9] – exploitation of vulnerabilities that have not yet been publicly disclosed or patched by vendors.

As a result of the conducted systematization of threats and issues in information security, it has been established that the modern computing environment is a complex, dynamic system that constantly faces risks from both external attacks and internal threats. Traditional security measures no longer provide a sufficient level of reliability in detecting, preventing, and counteracting threats. This highlights the need for intelligent detection tools, automated response mechanisms, continuous staff education, and clearly defined access and monitoring policies.

Considering the above, although existing artificial intelligence models are theoretically capable of detecting security incidents, in practice they remain imperfect. They lack access to high-quality real-world training data, and their integration into actual IT systems is complex and resource-intensive. Therefore, developing methods for automated network reconfiguration using practical monitoring tools such as *Zabbix* and the *pfSense* network gateway offers a more applied and effective approach to enhancing the flexibility and security of information infrastructure. Such an approach is easier to implement in real systems and can be adapted to specific infrastructure architectures.

3. The aim and objectives of the study

The aim of this work is to develop network monitoring tools to enhance the security of the information environment by enabling the detection of anomalies in the computer network and ensuring its automatic reconfiguration through dynamic IP address changes on the network hosts.

To achieve this aim, the following objectives are set:

– To justify the methodology of comprehensive monitoring of the operating system state and network environment based on the *Zabbix* monitoring system.

– To develop a method for automatic network reconfiguration with dynamic IP address management using *Zabbix* monitoring tools and the *pfSense* network gateway.

– To implement a practical solution for automatic network reconfiguration with dynamic IP address management based on *Zabbix* monitoring tools and the *pfSense* network gateway.

4. Materials and methods for automatic network reconfiguration with dynamic IP address management

4.1. Methodology for comprehensive operating system monitoring based on Zabbix

As a key aspect of maintaining the stability and security of information infrastructure – including enterprise computer systems, data centers, and networks – continuous collection and analysis of operating system parameters is considered essential for tracking the state of servers. These parameters include active processes, consumption of system resources (CPU, memory, disk), and active network connections, with the ability to determine which processes interact with external IP addresses. Monitoring also covers changes to system files related to access rights, passwords, and user accounts.

To support comprehensive monitoring of operating system parameters and detect anomalies, the *Zabbix* monitoring system is employed. *Zabbix* is widely used for tasks of this nature and enables centralized data collection, providing a complete real-time picture of host load.

To implement this monitoring approach in practice using Zabbix, the methodology for comprehensive OS monitoring is summarized and justified below.

Step 1. Data Collection on Running Processes.

Zabbix includes built-in capabilities for monitoring individual processes. Standard items such as `proc.num, proc.cpu`, and `proc.mem` allow tracking of:

– The number of concurrently running processes with a specific name;

– The percentage of CPU usage by these processes;

– Memory consumption by processes.

For example, the item `proc.cpu[nginx]` collects information on how much CPU time is consumed by the `nginx` process. This helps identify resource-intensive processes and diagnose potential resource leaks. For extended monitoring – such as obtaining a list of the most resource-consuming processes – `UserParameters` are used. By defining shell commands in the agent configuration file (`zabbix_agentd.conf`), *Zabbix* can receive a list of top system-consuming processes. This gives administrators regular analytics on the processes that impose the heaviest load.

Step 2. Monitoring System Resource Usage, Log Collection and File Integrity Checking. The *Zabbix Agent*, installed on client hosts, enables interaction with and retrieval of system parameters.

Zabbix provides a broad range of data items for monitoring system resources:

– CPU Monitoring.

– RAM Monitoring.

– Disk Subsystem Monitoring.

– Swap Memory Monitoring.

– Account and password hash file monitoring.

– System log retrieval Monitoring.

Collected data is displayed in *Zabbix* as graphs and tables, allowing for long-term trend analysis. File modifications can trigger alerts sent via external channels (email, SMS, etc.).

Step 3. Analysis of Network Connections Linked to Processes and External IPs. Monitoring active network connections allows detection of unauthorized activity, including connections to external hosts and potential malware.

Since *Zabbix* lacks native support for mapping connections to processes and IPs, custom scripts via `UserParameter` are used. This allows generation of a list of active external connections including:

– Connection IP address;

– Process or PID that initiated the connection.

Further filtering may include:

– Port-specific rules;

– Unique IP counters;

– Connection frequency statistics.

On Windows systems, *PowerShell* is used for collecting this data. This provides a complete overview of active connections initiated by local processes.

Collected results can be:

– Logged;

– Displayed in *Zabbix* as data items;

– Used to trigger security alerts, e.g., "a process established a connection to an IP not included in the whitelist."

Step 4. Practical Integration into *Zabbix Templates*

To apply these configurations across multiple hosts, *Zabbix* templates should be created, including:

– Data items for processes, system resources, and connections;

– Low-level discovery (LLD) rules for detecting new connections or processes;

– Triggers to signal anomalies (e.g., a process consuming >80% CPU, or a new connection to an unknown IP);

– Graphs and dashboards for real-time visualization of load.

For this method, it is essential to maintain IP whitelists based on system behavior and interactions with other services. A profiling process should be conducted to analyze which services or programs are running, what resources they use, and whether such behavior is typical. For most applications and services, resource usage profiles can be compiled, and any deviations should trigger immediate administrator alerts.

Based on the summarized methodology, it can be concluded that combining standard and extended data collection tools in *Zabbix* enables a comprehensive view of the operating system state, covering both internal processes and external interactions. This level of monitoring is critically important for maintaining information security, optimizing resource usage, and enabling timely incident response.

4.2. Method for automatic network reconfiguration with dynamic ip address management

To enhance the methodology of comprehensive monitoring, an integration of the *Zabbix* monitoring system with the *pfSense* network gateway is proposed. *pfSense* is an open-source software solution functioning as a network router and firewall, based on the *FreeBSD* operating system.

Based on the integration of *Zabbix* and *pfSense*, a method is proposed for automatic network reconfiguration with support for dynamic IP address changes on client Windows hosts. The method leverages *Zabbix's* capabilities for collecting network state data and *pfSense's* functionality for automatically updating routing rules and traffic filtering.

In dynamic environments – such as during the implementation of isolation policies, testing of new subnets, or responding to security incidents – there is often a need for rapid reconfiguration of the network topology. Such reconfiguration includes changing IP addresses on endpoint hosts and updating the corresponding routing and filtering rules in *pfSense*.

The proposed method consists of the following stages:

Stage 1. System Preparation
– Definition of the system's operational baseline.
– Preparation of scripts to execute configuration changes.
– Configuration of monitoring elements within the *Zabbix* system.

Stage 2. Detection of Changes via *Zabbix* Monitoring System
– Monitoring the state of hosts.
– Anomaly detection.
– Suspicious external connections.
– Environment changes.
– Modification of system files that should remain unchanged.

Stage 3. Impact Assessment and Trigger Activation
– Upon detecting anomalies, *Zabbix* triggers are activated. These triggers can initiate external scripts using *Zabbix's* `Remote Command` or Media type mechanisms, which then modify the network environment.

Stage 4. Execution of Response Script to Counter Anomalous Activity
– Unknown or unauthorized connections are blocked, or the network is restructured according to defined parameters using *pfSense's* network management features (e.g., `firewall rules, routing updates, DHCP IP reassignment`).

Stage 5. Incident Analysis by Administrators
– After automated reconfiguration, system administrators can analyze the incident, roll back changes if necessary, or log and classify the event as a cybersecurity incident.

The expected outcomes of implementing this method include:
– Automated Response: Reduced time between incident detection and response actions.
– Centralized Control: *Zabbix* serves as the centralized control point for all automated responses.
– Secure Reconfiguration: *pfSense* processes only pre-approved commands via its API, allowing for audit trails.
– Scalability: The approach is easily adaptable to large-scale environments with multiple hosts.

According to the theoretical analysis, the integration of *Zabbix* and *pfSense* has the potential to support the creation of a dynamic network infrastructure, capable of automatically reconfiguring in response to specific events. This includes adapting host parameters to enhance security – by minimizing open ports and limiting possible connections – and managing routing to prevent contact with unknown external sources that may pose a threat.

The proposed method is intended to isolate suspicious nodes or redirect network load between segments automatically, without administrator intervention. This approach is expected to improve the overall security posture of the network. Following automated reconfiguration, administrators are granted the opportunity to review and confirm the changes, roll them back, or register the event as a confirmed cyber incident.

5. Practical implementation of the method for automatic network reconfiguration with dynamic IP address management

To verify the functionality of the proposed method for automatic network reconfiguration with dynamic IP address management, an experimental implementation was developed.

Two experimental scenarios were created, each modeling a typical situation that requires rapid response by the network infrastructure. These scenarios were implemented in a laboratory environment using the *Zabbix* monitoring system, the *pfSense* network gateway, and client Windows hosts. The primary goal was to evaluate the system's ability to detect security-related events and respond automatically without administrator intervention.

Scenario 1: Attempt to Connect to an IP Address Outside the Whitelist

In the first scenario, a situation is simulated where a client host attempts to connect to an IP address that is not included in the predefined whitelist of allowed IP addresses. This whitelist defines the range of resources that are permitted for connection according to a set security policy.

Step 1. Preparation of Scripts and Monitoring System Configuration

At this stage, the appropriate whitelist of legitimate IP connections is defined. The whitelist file, located at `/etc/zabbix/whitelist.txt`, contains allowed IP addresses, each listed on a separate line. *Zabbix* agent parameters are configured on the Windows system to retrieve all active network connections' IP addresses. This is done by using PowerShell-based custom scripts that return the IP addresses of established outbound connections from the host.

```
UserParameter=custom.netstat.active_ips,powershell   -NoProfile   -
Command  "Get-NetTCPConnection  |  Where-Object  {  $_.State  -eq
'Established' } | Select-Object -ExpandProperty RemoteAddress | Sort-
Object -Unique"
```

The algorithm for retrieving parameters of active network connections and filtering them against the whitelist is implemented in the `script ip_discovery.sh` and illustrated in Fig. 1*a*.

The algorithm for generating a blocking rule is implemented in the script `block_ip_pfsense.sh` and illustrated in Fig. 1*b*.

To collect information via the monitoring system, an auto-discovery rule was added to the monitored Windows host object. It includes a name, check type (external check), and a key pointing to the IP discovery script.

Prototype items and triggers were created next:

– `Items` define the name, key, and polling interval.

– `Triggers` are configured to activate when the item value equals zero.

A blocking script is triggered through `Actions → Trigger Actions` when a `Trigger` is fired. This script runs via the *Zabbix* server and blocks the suspicious IP address.

Step 2. Detection of suspicious IPs

IP discovery runs every minute via the `ip_discovery.sh` script. After execution, the system creates item and trigger instances for each unique detected IP address based on the prototypes, as shown in Fig. 2.

Step 3. `Trigger` activations

Once the IP is detected, the `Trigger` is activated, which initiates the `blocking Action`. A separate `Action` is configured to send alerts upon activation of any `high-severity Trigger`.

After `Action` execution, *pfSense* receives corresponding firewall rules, as shown in Fig. 3.

Step 4. Administrators can then analyze the incident and determine the process or reason behind the connection attempt.

Scenario 2: Unauthorized Access

In the second scenario, anomalous application behavior is simulated, potentially indicating unauthorized access – an intrusion or unauthorized use of IT resources.

Step 1. Preparation phase:

A custom parameter is added to the *Zabbix agent* running on the host:

```
UserParameter=custom.bad_cpu_proc.njson,powershell   -NoProfile   -
ExecutionPolicy Bypass -File "C:\Scripts\bad_cpu_proc.ps1"
```

The algorithm of the script `C:\Scripts\bad_cpu_proc.ps1` used for data collection is shown in Fig. 4*a*. A data item is created in Zabbix, configured to retrieve data via the agent. A derived item processes the whitelist of known legitimate processes using a post-processing script, illustrated in Fig. 4*b*.

Next, a `Trigger` is defined based on whitelist post-processing logic. When triggered, it launches an *Action* that executes an IP address change script.

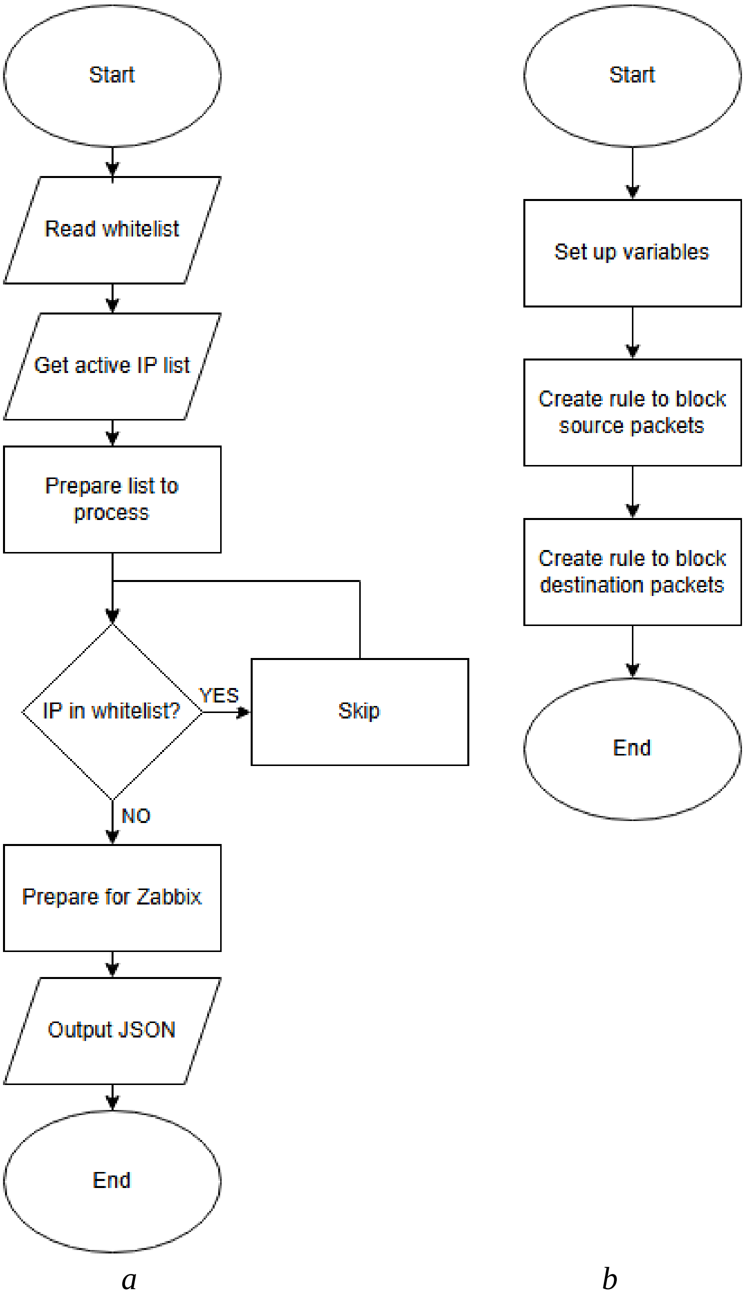
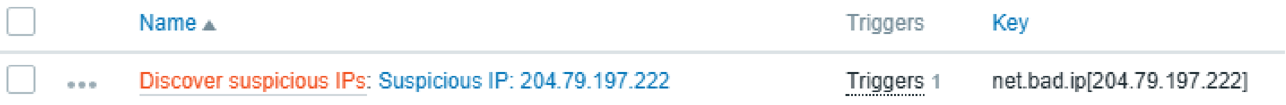
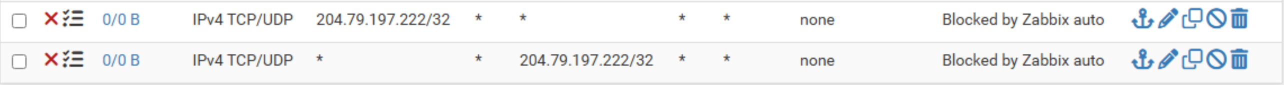


Fig. 1. Algorithms for obtaining the list of IP addresses and blocking those not in the whitelist: *a* – IP discovery algorithm; *b* – IP blocking rule generation algorithm



□		Name ▲	Triggers	Key
□	•••	Discover suspicious IPs: Suspicious IP: 204.79.197.222	Triggers 1	net.bad.ip[204.79.197.222]

Fig. 2. View of Items containing a suspicious IP address in the Zabbix interface



□	✕	0/0 B	IPv4 TCP/UDP	204.79.197.222/32	*	*	*	*	none	Blocked by Zabbix auto	
□	✕	0/0 B	IPv4 TCP/UDP	*	*	204.79.197.222/32	*	*	none	Blocked by Zabbix auto	

Fig. 3. Blocking rules for IP addresses in *pfSense*

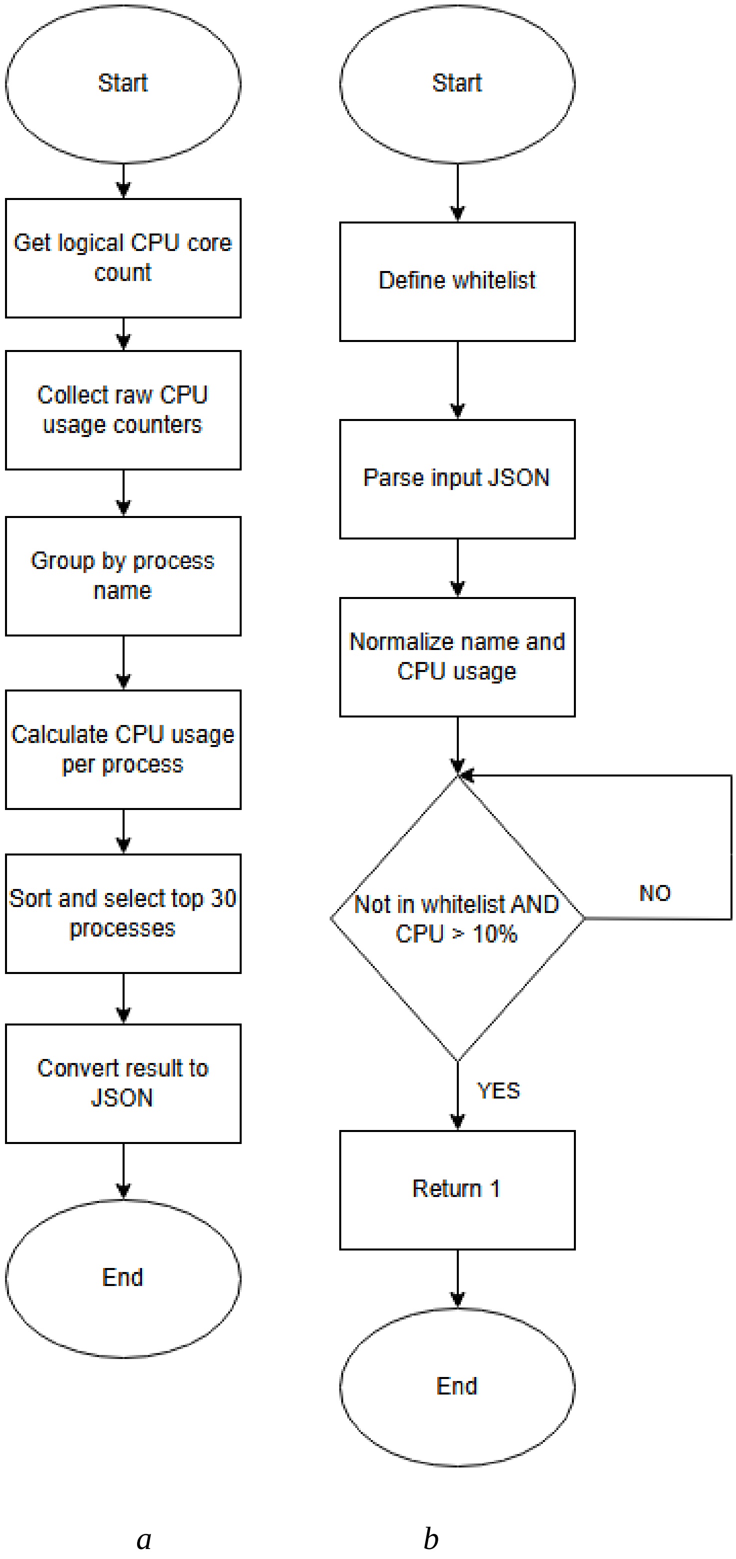

Start
Get logical CPU core count
Collect raw CPU usage counters
Group by process name
Calculate CPU usage per process
Sort and select top 30 processes
Convert result to JSON
End
Start
Define whitelist
Parse input JSON
Normalize name and CPU usage
Not in whitelist AND CPU > 10%
NO
YES
Return 1
End
a
b

Fig. 4. Algorithms for processing CPU load data: *a* – data collection script; *b* – whitelist validation and resource usage filtering

The algorithm of the script for modifying the host's IP and routing rules (`/usr/local/bin/change_node_ip.sh`) is presented in Fig. 5*a*.

Fig. 5*b* shows the algorithm of the `script zbx_change_ip.sh` which updates the IP address in the *Zabbix* monitoring system.

Fig. 5*c* presents the script `zbx_remote_ip_change.sh`, responsible for updating the IP address on the Windows host itself.

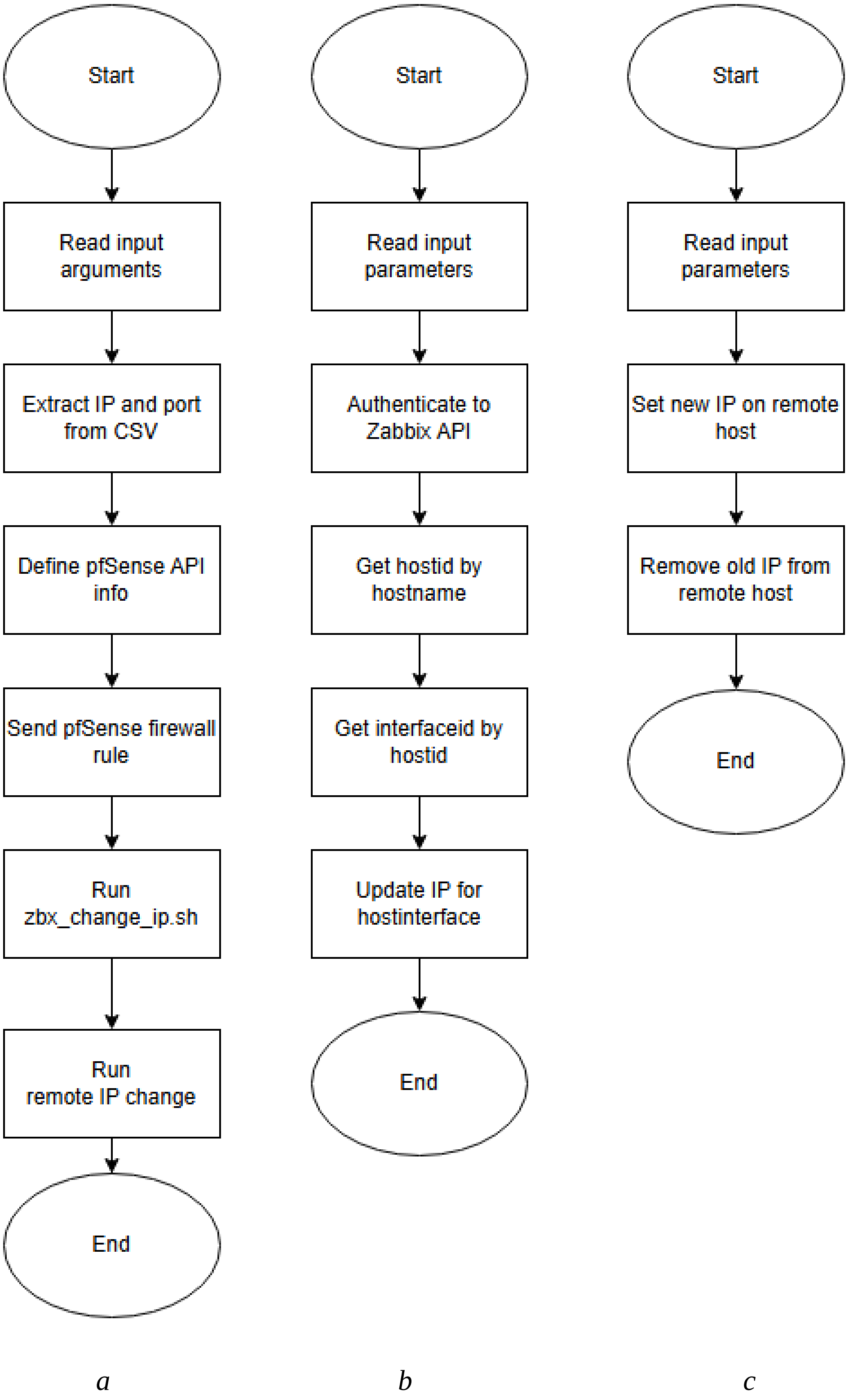

Start
Read input arguments
Extract IP and port from CSV
Define pfSense API info
Send pfSense firewall rule
Run zbx_change_ip.sh
Run remote IP change
End
Start
Read input parameters
Authenticate to Zabbix API
Get hostid by hostname
Get interfaceid by hostid
Update IP for hostinterface
End
Start
Read input parameters
Set new IP on remote host
Remove old IP from remote host
End
a
b
c

Fig. 5. Algorithms for changing the host IP address: *a* – IP and routing change; *b* – updating the IP in the monitoring system; *c* – IP change on the Windows host

Step 2. Continuous system polling to detect abnormal CPU usage patterns.

Step 3. Comparison of detected processes against the known whitelist.

Step 4. `Trigger` activation upon deviation from baseline behavior.

Step 5. Execution of the `IP change procedure`, `routing update`, and reconfiguration in *Zabbix*.

Step 6. Administrator notification and incident logging.

6. Discussion of results

This work examined a method for modifying network parameters using a monitoring system.

In cases where there are issues with access to *pfSense* or the *Zabbix agent*, administrators are also notified. However, this aspect is not considered in this study, as it is a standard polling operation from the *Zabbix* side using built-in tools.

The discussed scenarios illustrate how the method can be applied. The first scenario may be harmless – for example, when Windows services attempt to connect to the cloud. Since Microsoft has a wide range of IP addresses, such connections may trigger the response mechanism. However, even these cases can negatively affect the system by pushing it into an unstable state and causing excessive resource usage.

The second scenario involves malicious activity, where an attacker has launched an unauthorized service or replaced a legitimate one, thereby starting to exfiltration or modify data on the server. In such a case, changing the IP address and enabling a minimal set of access rules by port helps to block the unauthorized data access attempt. Subsequent isolation protects the environment from further compromise or lateral movement.

These examples merely demonstrate the method's capabilities and should be tailored to specific systems with their own sets of active services, IPs, and other parameters.

Future developments may include handling more complex environmental impact scenarios and enhancing anomaly mitigation actions by integrating elements of artificial intelligence and widely-used vulnerability and attack scenario dissemination systems into the method.

Conclusion

This paper addressed current information security challenges in the context of increasingly complex cyber threats and proposed practical solutions for building a dynamic monitoring and response system. Based on the analysis of modern threats and approaches, particularly involving artificial intelligence, it was concluded that despite the high potential of such technologies, their implementation in real-world IT systems faces numerous difficulties, including the need for large data volumes and the complexity of scaling.

The method of automatic network reconfiguration proposed in this work – based on the integration of the *Zabbix* monitoring system and the *pfSense* network gateway – is more practical and suitable for deployment in corporate environments. The system not only detects anomalies in operating system and network behavior but also responds promptly by automatically changing the host's IP address, isolating suspicious nodes, or adapting routing and firewall rules.

Practical implementation has shown that this approach provides effective control over internal and external threats, increases the level of automation in security incident management, and enables administrators to respond to suspicious activity in a timely manner. At the same time, the system remains scalable and flexible, allowing it to be adapted to specific network architectures and security policies.

Thus, the combination of *Zabbix* and *pfSense* tools enables the creation of a dynamic, secure, and self-governing information environment, representing a promising direction in modern cybersecurity.

References

1. A. Mosteiro Vázquez, C. Dafonte, and Á. Gómez, *"*Open Source Monitoring System for IT Infrastructures Incorporating IoT-Based Sensors,*"* *Proceedings*, vol. 54, no. 1, p. 56, 2020. [Online]. Available: https://doi.org/10.3390/proceedings2020054056
2. Y. Zhao, L. Zhang, X. Li, *"*Research on Zabbix Monitoring System for Large-scale Smart Campus Network from a Distributed Perspective, " *Journal of Engineering Sciences*, vol. 54, no. 1, pp. 56–65, 2024. [Online]. Available: https://journal.esrgroups.org/jes/article/view/5153
3. A. Mardiyono, W. Sholihah, and F. Hakim, "Mobile-based Network Monitoring System Using Zabbix and Telegram," *2020 3rd International Conference on Computer and Informatics*

Engineering (IC2IE), Yogyakarta, Indonesia, 2020, pp. 473–477, https://doi.org/10.1109/IC2IE50715.2020.9274582.

4. A. Suparman, E. P. A. Akhmad, and B. M. Dinata, “Leveraging Artificial Intelligence for Enhancing Cybersecurity: A Deep Learning Approach to Real-Time Threat Detection,” *The Journal of Academic Science*, vol. 1, no. 7, pp. 835–842, Nov. 2024, https://doi.org/10.59613/0yv79c49
5. N. Ahmed, M. E. Hossain, Z. Hossain, and F. Kabir, “AI-Enabled System for Efficient Cyber Incident Detection and Response in Cloud Environments: Safeguarding Against Systematic Attacks,” *Indonesian Journal of Educational Science and Technology*, Nov. 2024, https://doi.org/10.55927/nurture.v3i4.16.
6. M. Arafune et al., “Design and Development of Automated Threat Hunting in Industrial Control Systems,” 2022 *IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*, Pisa, Italy, 2022, pp. 618–623, https://doi.org/10.1109/PerComWorkshops53856.2022.9767375.
7. N. Katiyar, S. Tripathi, P. Kumar, and S. Verma, “AI and Cyber-Security: Enhancing threat detection and response with machine learning,” *Educational Administration Theory and Practice Journal*, vol. 30, no. 4, Apr. 2024, https://doi.org/10.53555/KUEY.V30I4.2377.
8. Ö. Aslan, S. S. Aktuğ, M. Ozkan-Okay, A. A. Yilmaz, and E. Akin, “A Comprehensive Review of Cyber Security Vulnerabilities, Threats, Attacks, and Solutions,” *Electronics*, vol. 12, no. 6, Art. no. 1333, Mar. 2023, https://doi.org/10.3390/electronics12061333.
9. A. I. Jony and S. A. Hamim, “Navigating the Cyber Threat Landscape: A Comprehensive Analysis of Attacks and Security in the Digital Age,” *Journal of Information Technology and Cyber Security*, vol. 1, no. 2, pp. 53–67, Jul. 2023, https://doi.org/10.30996/jitcs.9715.

УДК 004.056.5, 004.7, 004.056.53

МЕТОД АВТОМАТИЧНОЇ РЕКОНФІГУРАЦІЇ МЕРЕЖІ З ДИНАМІЧНИМ КЕРУВАННЯМ ІР-АДРЕСАЦІЄЮ

Анатолій Гайдай
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна
http://orcid.org/0000-0001-9330-414X

Ірина Клименко
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна
http://orcid.org/0000-0001-5345-8806

У контексті зростання кіберзагроз особливої актуальності набувають системи, здатні не лише виявляти аномалії в роботі мережевої інфраструктури, а й оперативно реагувати на них без втручання адміністратора. У статті запропоновано метод автоматичної реконфігурації мережі на основі інтеграції системи моніторингу *Zabbix* із функціональністю мережевого шлюзу *pfSense*. Така система дозволяє не лише централізовано контролювати стан операційної системи, використання ресурсів і мережеву активність, а й автоматично змінювати IP-адреси хостів, адаптувати маршрутизацію та обмежувати з'єднання згідно з визначеними політиками безпеки.

Метою дослідження є розробка методу автоматичного моніторингу та реконфігурації мережі з динамічною зміною IP-адрес для підвищення ефективності протидії кіберзагрозам. Об'єктом дослідження виступають процеси управління інформаційною безпекою в комп'ютерних мережах. Предметом дослідження є методи виявлення аномалій та автоматичного реагування шляхом зміни мережевих параметрів за допомогою *Zabbix* та *pfSense*.

У контексті автоматичного реагування на виявлені загрози здійснено формалізацію методу комплексного моніторингу операційної системи клієнтських хостів, з подальшим аналізом логів, дій користувача, завантаження ресурсів, використання мережевих портів та взаємодії з зовнішніми сервісами. Розроблено та реалізовано методологію реконфігурації мережі після фіксації аномалії: зокрема зміну IP-адреси з збереженням працездатності в мінімальній конфігурації мережевих доступів, ізоляцію вузла за допомогою *pfSense*. Задіяно скрипти для клієнтських ОС Windows, що взаємодіють з API *Zabbix* та *pfSense*, тим самим забезпечуючи динамічність і повну автоматизацію.

Результати тестування запропонованої системи на симульованому середовищі підтверджують її ефективність. У порівнянні з ручними або частково автоматизованими рішеннями, час реагування на інцидент було зменшено, а ризик поширення атаки всередині мережі – зведено до мінімуму.

Ключові слова: *Zabbix*, *pfSense*, мережева безпека, виявлення аномалій, автоматизоване реагування.