Brilliance
Research of Artificial Intelligence

E-ISSN : 2807-9035
Volume 6, **Number** 4, November 2026
https://doi.org/10.47709/brilliance.v6i4.9646

Development of an Adaptive Dynamic Extended Access Control List Model for Data Communication Security

Reski Yulian Fauzan[1*], Ihsan Lumasa Rimra[2], Ummul Khair[3]
[1,2,3]Telecommunications Engineering, Electrical Engineering Department, Padang State of Polytechnic, Indonesia
[1*]reskiyulian@pnp.ac.id, [2]rimra@pnp.ac.id, [3]ummul@pnp.ac.id

open access

***Corresponding Author**

Article History:
Submitted: 13-09-2026
Accepted: 13-10-2026
Published: 01-11-2026

Keywords:
dynamic extended ACL; data communication security; authentication; session-based access control; network automation.

Brilliance: Research of Artificial Intelligence is licensed under a Creative Commons Attribution-NonCommercial 4.0 International (CC BY-NC 4.0).

ABSTRACT

Data communication security requires access control mechanisms that can adapt to changing user and service requirements. Conventional Extended Access Control List (Extended ACL) relies on predefined rules and provides limited flexibility for authentication-based and session-based access control. This study aims to develop and implement an adaptive Dynamic Extended Access Control List model that integrates contextual authentication, automated access-rule activation, and session-based access control. An experimental approach was employed using Cisco Packet Tracer to simulate a client–router–server network topology consisting of public, private, and administrative network segments. A web-based authentication mechanism was designed to identify the client's network segment and determine the appropriate access authorization process. Following successful authentication, Python Netmiko was used to automatically activate service-specific Dynamic ACL rules on the router through SSH. The implemented model was tested using HTTP/HTTPS, ICMP, FTP, and SSH access scenarios under pre-authentication and post-authentication conditions. The results demonstrate that the developed model successfully provides temporary and service-specific access following successful authentication. Public-segment users initially received access only to HTTP/HTTPS services and obtained additional ICMP and FTP access after successful authentication. Private-segment users initially had access to HTTP/HTTPS, ICMP, and FTP services, while SSH access was activated after authentication. The model also implemented a 10-minute idle timeout mechanism to automatically terminate inactive access and restore the initial security policy. These findings demonstrate that integrating contextual authentication, automated Dynamic ACL activation, and session-based control provides a flexible mechanism for managing temporary service access in basic data communication security.

INTRODUCTION

Data communication security is an important aspect of computer network management because access to network resources needs to be controlled according to user requirements and the services available. Access control in computer networks can generally be implemented using Access Control Lists (ACLs) on network devices. ACLs allow network administrators to determine which traffic is permitted or denied based on specific parameters, such as source address, destination address, communication protocol, and port number. The implementation of ACLs has been widely adopted to restrict access to network services and control traffic in VLAN-based and local area networks (Azmi et al., 2022; Tahir et al., 2024; Usior & Sediyono, 2023).

Extended ACLs provide more specific access control capabilities than Standard ACLs because access rules can be defined based on source and destination addresses, communication protocols, and specific services or port numbers. This approach enables administrators to establish more targeted access policies for network services such as HTTP, HTTPS, FTP, ICMP, and SSH (Jung et al., 2022; Usior & Sediyono, 2023). However, the implementation of Extended ACLs in a static configuration still has limitations when access requirements are temporary and dependent on user conditions. Rules configured permanently generally remain active until the administrator manually modifies the configuration. This condition makes the access control mechanism less flexible for scenarios requiring access privileges based on authentication, specific service requirements, and session duration.

These limitations indicate the need for an access control mechanism that is not only based on IP addresses and network services but is also capable of adapting access policies according to user context and session conditions. Integrating an authentication mechanism with Dynamic Extended ACLs can provide a means of granting access privileges dynamically after successful user verification, consistent with recent approaches to dynamic network access control (You et al., 2023; Temoshok et al., 2025). In addition, automating rule changes on network devices can reduce dependence on manual administrator configuration and enable access privileges to be granted only when required. This approach is relevant for supporting temporary and session-based access control in computer networks.

This is an Creative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Brilliance
Research of Artificial Intelligence

E-ISSN : 2807-9035
Volume 6, **Number** 4, November 2026
https://doi.org/10.47709/brilliance.v6i4.9646

Based on these issues, this study aims to develop and implement an Adaptive Dynamic Extended Access Control List model to improve the flexibility of access control in data communication. The proposed model integrates network segment identification, a web-based authentication mechanism, automated activation of Dynamic ACL rules, and session-based access control. The model was implemented using Cisco Packet Tracer with a network scenario consisting of public, private, and administrative segments. The activation of dynamic access rules was integrated with Python Netmiko through an SSH connection, allowing network service access privileges to be granted according to the user's authentication status.

The main contribution of this study is the development of an access control model that integrates network segment context, authentication, dynamic service-specific access authorization, and session-based access termination. With this approach, access to network services is not solely determined by permanently configured ACL rules but can be activated according to user requirements following successful verification and restored to the initial security policy when the session ends. The developed model is expected to provide a basic approach for implementing more adaptive data communication security in computer networks.

LITERATURE REVIEW

Access Control List (ACL) is a network security mechanism used to control traffic based on predefined rules. ACL rules determine whether packets are permitted or denied according to parameters such as source and destination IP addresses, communication protocols, and service ports. Standard ACL generally provides basic filtering based primarily on source addresses, whereas Extended ACL offers more granular control by considering source and destination addresses, protocols, and port numbers (Azmi et al., 2022). Previous studies have demonstrated the application of ACL for controlling access in LAN, WLAN, and VLAN environments (Dewi et al., 2022; Tahir et al., 2024; Usior & Sediyono, 2023).

Extended ACL is particularly suitable for service-specific access control because it allows administrators to define policies for services such as HTTP, HTTPS, FTP, ICMP, and SSH. However, conventional Extended ACL implementations generally rely on manually configured rules that remain active until explicitly modified or removed. This characteristic limits flexibility when access privileges need to be provided temporarily or according to changing user requirements (Rahman & Dasuki, 2025).

Dynamic Access Control provides an alternative by allowing temporary access privileges to be activated according to authentication or contextual conditions. Recent studies have investigated dynamic access control mechanisms that adjust network authorization according to changing security conditions and access requirements (Jung et al., 2022; You et al., 2023). In Cisco network environments, dynamic access mechanisms can also provide temporary user access through dynamically applied access-control entries with configured timeout mechanisms (Cisco Systems, 2025).

Network automation can further support dynamic policy enforcement by reducing the need for administrators to modify router configurations manually. Python-based network automation tools such as Netmiko enable automated communication with network devices through SSH and support the execution of configuration commands. Previous research has demonstrated the use of Netmiko for automating network-device configuration and management tasks in simulated network environments (Gondhalekar et al., 2024).

Although previous studies have established the effectiveness of ACL and Extended ACL for network traffic control, most implementations emphasize predefined or manually managed access policies (Arif et al., 2025; Rahman & Aprianto, 2025). The integration of network-segment context, authentication, automated service-specific Dynamic ACL activation, and session-based access termination within a single access-control model remains insufficiently addressed. Therefore, this study develops an Adaptive Dynamic Extended ACL model that integrates these components to provide temporary and context-aware service authorization in a computer network.

METHOD

This study employed a Research and Development (R&D) approach to develop and implement an Adaptive Dynamic Extended Access Control List (ACL) model for data communication security. The development focused on integrating network-segment identification, authentication, automated access-rule activation, and session-based access control into a unified mechanism. The proposed model was implemented and functionally evaluated using Cisco Packet Tracer to represent the network environment and access-control scenarios.

The research activities consisted of requirement identification, access-control model design, network topology development, authentication mechanism development, Dynamic Extended ACL integration, automation implementation, and functional testing. The requirement identification stage analyzed the limitations of conventional static Extended ACL, particularly its dependence on predefined rules and manual configuration when temporary access privileges are required. Based on these requirements, the proposed model was designed to provide additional service privileges after successful authentication and to automatically terminate temporary privileges when the access session expires.

This is an Creative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Brilliance
Research of Artificial Intelligence
E-ISSN : 2807-9035
Volume 6, Number 4, November 2026
https://doi.org/10.47709/brilliance.v6i4.9646

The network environment was designed using a client–router–server architecture consisting of public, private, and administrative network segments. The router functioned as the central routing device and as the Policy Enforcement Point (PEP), where the Dynamic Extended ACL rules were enforced. Inter-VLAN communication was implemented using a router-on-a-stick configuration through the router's GigabitEthernet interface. Three VLANs were defined to represent different user contexts: VLAN 10 for the private or staff segment, VLAN 20 for the public or guest segment, and VLAN 30 for the administrative segment. In addition, a homepage server and an authentication server were deployed in separate network segments to support web services and user authentication.

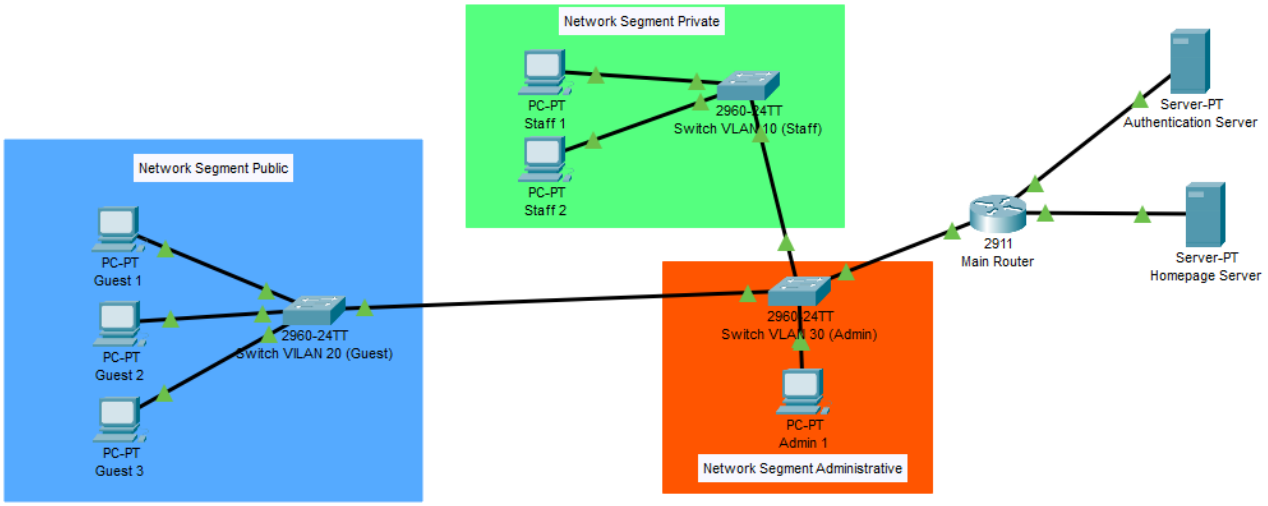


Figure 1. Network topology of the proposed Adaptive Dynamic Extended ACL model

Figure 1 illustrates the network topology used to implement the proposed model. The private segment uses the 192.168.10.0/24 network and contains two staff clients, while the public segment uses the 192.168.20.0/24 network and contains three guest clients. The administrative segment uses the 192.168.30.0/24 network and contains an administrative client. The homepage server is connected to the 172.16.0.0/24 network, whereas the authentication server is connected to the 10.10.0.0/24 network. The router provides routing between the network segments and applies the access-control policies according to the client's network context.
The addressing and network-segmentation scheme used in the implementation is summarized in Table 1.

Table 1. Network segmentation and addressing scheme

Network Segment	VLAN	Network Address	Client / Device	Function
Private/Staff	10	192.168.10.0/24	Staff 1, Staff 2	Internal user access
Public/Guest	20	192.168.20.0/24	Guest 1, Guest 2, Guest 3	Public user access
Administrative	30	192.168.30.0/24	Admin 1	Administrative access
Homepage Server	—	172.16.0.0/24	172.16.0.2	Web information service
Authentication Server	—	10.10.0.0/24	10.10.0.2	Authentication and access-control support

The access policy was designed according to the context of each network segment. Public users in VLAN 20 were initially permitted to access HTTP and HTTPS services, while ICMP and FTP services were restricted. After successful authentication, additional privileges for ICMP and FTP were dynamically activated for the authenticated client. Private users in VLAN 10 were initially permitted to access HTTP, HTTPS, ICMP, and FTP services, while SSH access was restricted. Following successful authentication or access escalation, temporary SSH access was activated. Users in the administrative segment were provided with broader service access according to their administrative role.

The service-specific access policy was implemented using Dynamic Extended ACL rules based on the client's source IP address, destination service, and communication protocol. HTTP and HTTPS access were associated with TCP ports 80 and 443, FTP access was associated with TCP ports 20 and 21, ICMP was controlled using the ICMP protocol, and SSH access was associated with TCP port 22. This configuration enabled the model to provide different service privileges according to the user's network segment and authentication status.

The workflow of the Adaptive Dynamic Extended ACL model consisted of seven sequential stages: requirement identification, network topology and access-policy design, network-segment identification, user authentication, dynamic access-rule generation, automated ACL activation, and session-based access termination. These stages were designed to integrate user authentication with network-level access enforcement, allowing additional service privileges to be activated temporarily according to the client's network context and authentication status. The resulting workflow

This is an Creative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Brilliance
Research of Artificial Intelligence

E-ISSN : 2807-9035
Volume 6, Number 4, November 2026
https://doi.org/10.47709/brilliance.v6i4.9646

can be represented as:

Requirement Identification → Network and Policy Design → Segment Identification → Authentication → Dynamic ACL Generation → Netmiko/SSH Activation → Timeout and Policy Restoration

Following the workflow described above, a web-based authentication mechanism was developed to identify the client's network segment and determine the appropriate authorization process. For public users, authentication was performed through the guest login mechanism. For private users, an authentication or access-escalation mechanism was provided to activate additional privileges. Following successful authentication, the backend application identified the client's IP address and generated the corresponding Dynamic Extended ACL commands

The automation component was implemented using Python and Netmiko. Netmiko established an SSH connection between the backend application and the router and transmitted the generated ACL configuration commands to the router. The automation process allowed the access policy to be activated without requiring the network administrator to manually modify the router configuration for every authentication event. Thus, authentication at the application level could be translated into temporary access privileges at the network level.

The session-based mechanism was implemented using a configured 10-minute timeout for dynamically activated access rules. When the configured inactivity period was reached, the temporary Dynamic Extended ACL entries were removed, and the client's access privileges returned to the initial security policy. The client was then required to perform authentication again to obtain the additional services. This mechanism was designed to minimize the possibility of temporary access privileges remaining active after the user's session was no longer required.

Functional testing was conducted using predefined access scenarios before and after authentication. For the public segment, testing evaluated HTTP and HTTPS access before authentication and ICMP and FTP access after successful authentication. For the private segment, testing evaluated HTTP, HTTPS, ICMP, and FTP access before authentication and SSH access following successful authentication. Additional testing was conducted to verify the automatic termination of dynamically activated privileges after the configured idle timeout.

The functional evaluation used a binary access criterion, namely permitted or denied, according to the expected access-control policy. A test was considered successful when the actual service-access result corresponded to the predefined policy for the client's network segment and authentication status. The testing results were subsequently used to evaluate whether the developed Adaptive Dynamic Extended ACL model was capable of providing temporary, service-specific, and context-aware access control.

RESULT

The implementation of the Adaptive Dynamic Extended ACL model produced an access-control mechanism that could adjust network service privileges according to the client's network segment and authentication status. The implementation consisted of network segmentation, initial access policies, web-based authentication, automated Dynamic ACL activation through Netmiko, and session-based access termination. The resulting access behavior was evaluated through functional testing on public, private, and administrative network segments.

Network and Access-Control Implementation

The implemented network topology consisted of three user segments, namely VLAN 10 for private or staff users, VLAN 20 for public or guest users, and VLAN 30 for administrative users. The router operated as the central routing device and Policy Enforcement Point (PEP), while the homepage server and authentication server provided the required web and authentication services.

The initial access policy was configured according to the role of each network segment. Public users were restricted to basic web services, whereas private users were provided with broader access to common network services but were restricted from SSH access. Administrative users were configured with broader access according to their administrative requirements. This initial policy established the security baseline before any dynamic access privilege was activated

Table 2. Initial access policy according to network segment

Network Segment	HTTP/HTTPS	ICMP	FTP	SSH
Public/Guest (VLAN 20)	Permitted	Denied	Denied	Restricted
Private/Staff (VLAN 10)	Permitted	Permitted	Permitted	Denied
Administrative (VLAN 30)	Permitted	Permitted	Permitted	Permitted

The results in Table 2 show that access privileges were differentiated according to the user's network context. The public segment received the most restrictive initial policy, while the private segment received access to commonly required services. Administrative users were provided with broader access. This configuration served as the initial state before authentication-based privilege escalation.

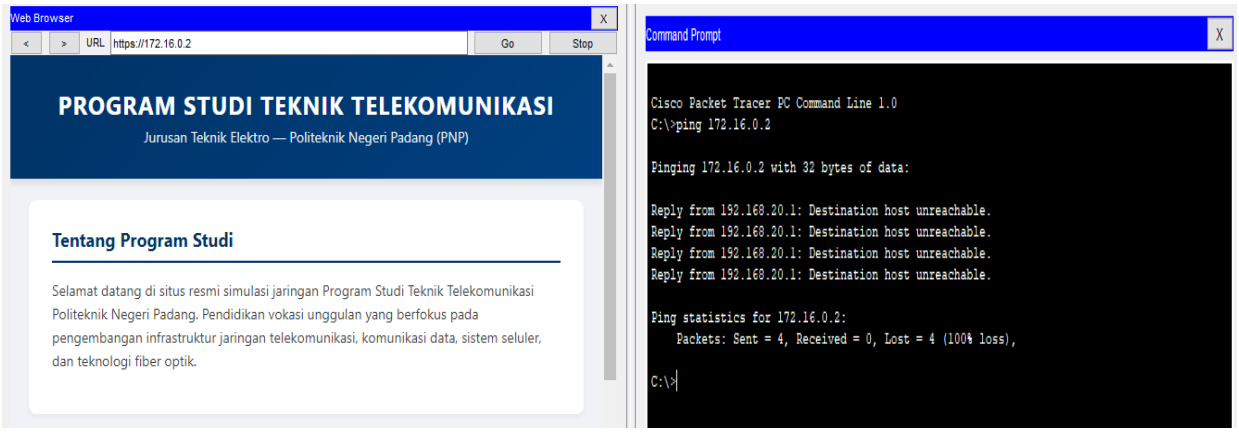
This is an Creative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Brilliance
Research of Artificial Intelligence

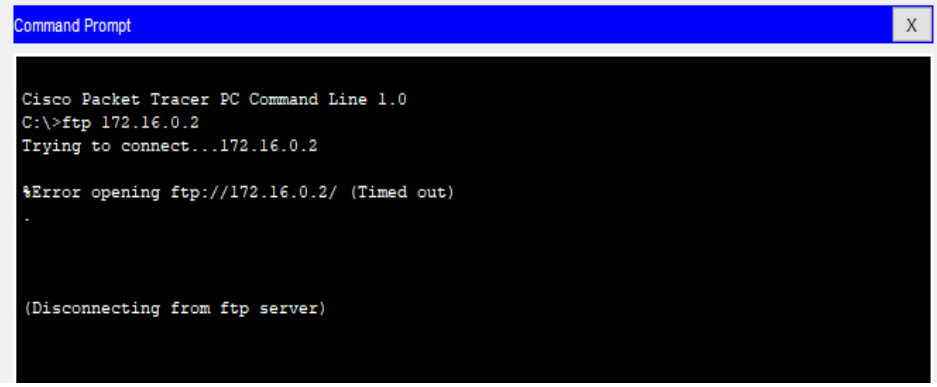
E-ISSN : 2807-9035
Volume 6, **Number** 4, November 2026
https://doi.org/10.47709/brilliance.v6i4.9646

Dynamic Access Activation for Public Users

Testing on the public segment was performed using the guest clients connected to VLAN 20. Before authentication, the guest clients could access the homepage server through HTTP and HTTPS, while ICMP and FTP access were restricted according to the initial Dynamic Extended ACL policy.



(a) (b)



(c)

Figure 2. Public-segment access condition before authentication: (a) HTTP access, (b) ICMP restriction, and (c) FTP restriction

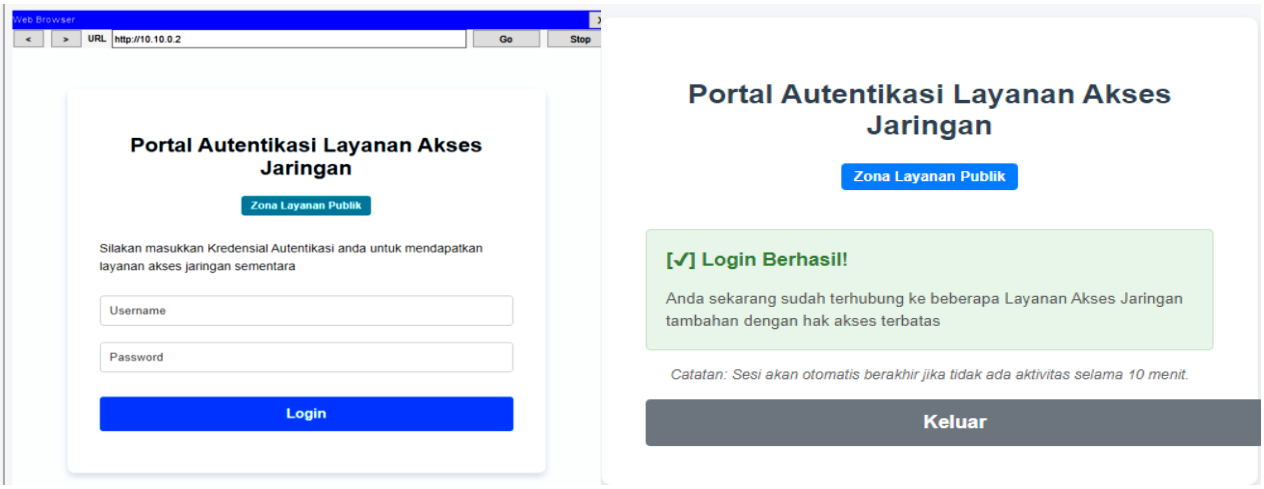
Figure 2 shows the access condition of the public-segment client before authentication. Figure 2(a) shows successful access to the HTTP service, indicating that HTTP access is permitted under the initial access policy. Figure 2(b) shows that ICMP access is restricted, as the client cannot perform communication using the ping protocol. Figure 2(c) shows that FTP access is also restricted before authentication. These results confirm that public users are initially limited to the predefined web services, while additional ICMP and FTP services remain unavailable before successful authentication..

After a guest user successfully completed the authentication process through the web-based authentication mechanism, the backend application identified the client's IP address and generated the corresponding Dynamic Extended ACL rules. The generated rules were subsequently transmitted to the router through an SSH connection using Python Netmiko.

This is an Creative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Brilliance
Research of Artificial Intelligence

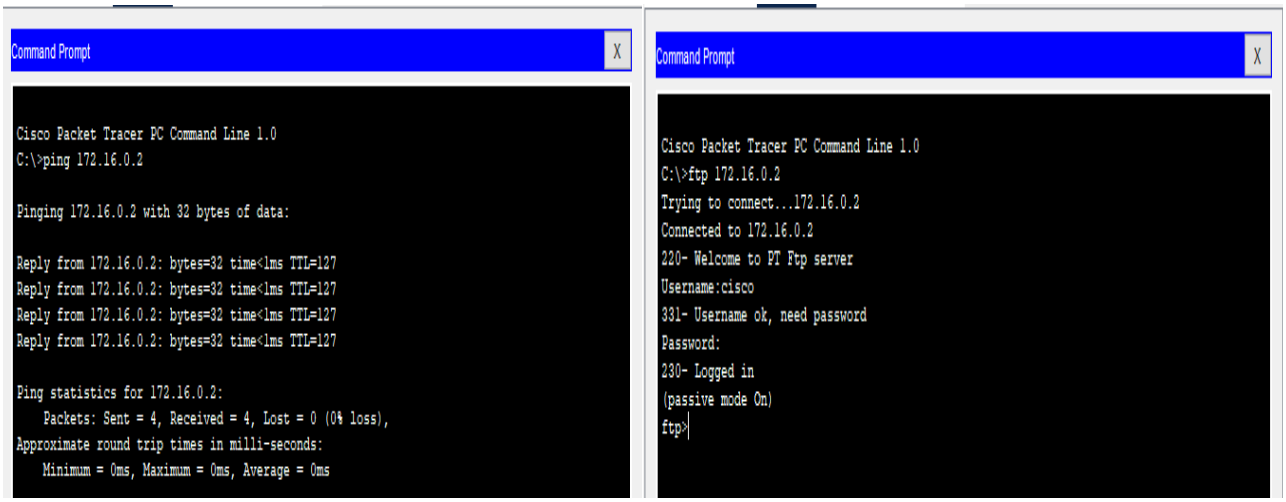
E-ISSN : 2807-9035
Volume 6, **Number** 4, November 2026
https://doi.org/10.47709/brilliance.v6i4.9646



(a) (b)

Figure 3. Web-based authentication process for public users: (a) authentication interface and (b) authentication process

Figure 3 shows the web-based authentication process for public users. Figure 3(a) shows the authentication interface used by the client to submit the required credentials, while Figure 3(b) shows the authentication process after the user's credentials are submitted. This process is used to verify the client's access request before additional network privileges are activated. Successful authentication subsequently becomes the trigger for activating the additional ICMP and FTP access defined in the public-segment policy. The Dynamic ACL activation resulted in additional access privileges for the authenticated client.



(a) (b)

Figure 4. Additional service access after authentication: (a) ICMP access and (b) FTP access

Figure 4 shows the additional service access obtained by the public client after successful authentication. Figure 4(a) shows that ICMP access becomes available after authentication, indicating that the corresponding Dynamic Extended ACL rule has been activated. Figure 4(b) shows that FTP access is also available after authentication. These results demonstrate that authentication activates only the additional services defined in the access policy, while the previously permitted HTTP and HTTPS services remain available.

Specifically, ICMP and FTP access were activated for the authenticated public client, while the initial HTTP and HTTPS privileges remained available. The access behavior before and after authentication is presented in table 3.

Table 3. Public-segment access before and after authentication

Service	Before Authentication	After Authentication
HTTP (TCP 80)	Permitted	Permitted
HTTP (TCP 443)	Permitted	Permitted
ICMP	Denied	Permitted
FTP (TCP 20/21)	Denied	Permitted
SSH (TCP 22)	Restricted	Restricted

This is an Creative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Brilliance
Research of Artificial Intelligence

E-ISSN : 2807-9035
Volume 6, **Number** 4, November 2026
https://doi.org/10.47709/brilliance.v6i4.9646

Table 3 shows that authentication selectively changed the access privileges of public users. Before authentication, ICMP and FTP were denied, while HTTP and HTTPS were permitted. After successful authentication, ICMP and FTP became permitted, whereas SSH remained restricted. These results confirm that authentication activated only the additional services defined by the public-segment policy without granting unrestricted access.

The results demonstrate that authentication did not replace the initial access policy. Instead, successful authentication activated additional service-specific privileges for the authenticated client. Therefore, the access-control mechanism could provide differentiated temporary access without granting unrestricted network access.

Dynamic Access Activation for Private Users

Testing was also conducted on the private or staff segment using VLAN 10. Before authentication or access escalation, staff clients were permitted to access HTTP, HTTPS, ICMP, and FTP services. SSH access remained restricted because administrative remote access was not included in the initial policy.

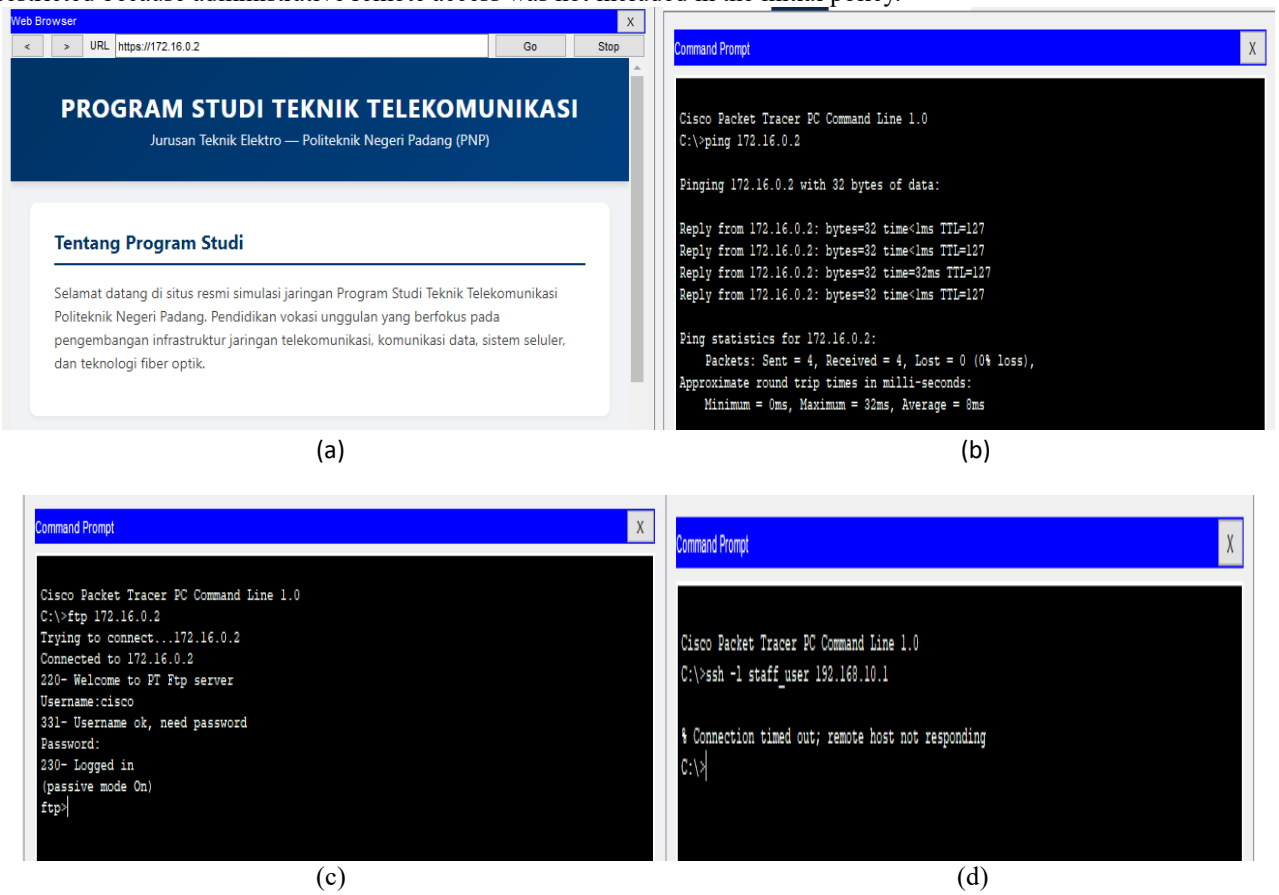


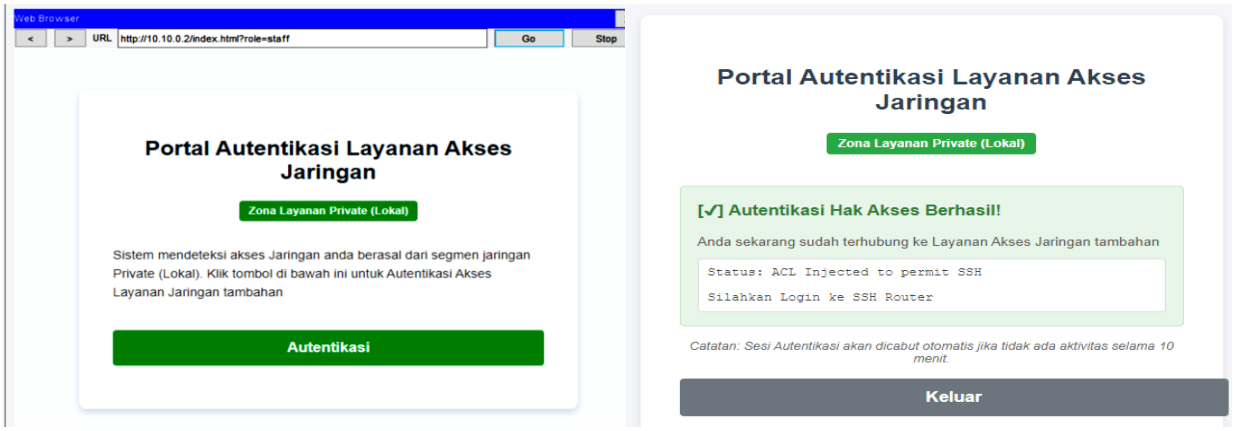
Figure 5. Private-segment access condition before authentication: (a) HTTP/HTTPS access, (b) ICMP access, (c) FTP access, and (d) SSH access denied

Figure 5 shows the initial access condition of the private-segment client before authentication or access escalation. Figure 5(a) shows that HTTP/HTTPS access is permitted, indicating that web services are available under the initial access policy. Figure 5(b) shows that ICMP access is permitted, while Figure 5(c) shows that FTP access is also available to the private client. In contrast, Figure 5(d) shows that SSH access is denied before authentication. These results confirm that private users receive broader access to common network services, while SSH remains restricted until the user successfully completes the authentication process.

Following successful authentication, the backend application generated a Dynamic Extended ACL rule for TCP port 22 based on the authenticated client's IP address.

This is an Creative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Brilliance
Research of Artificial Intelligence
E-ISSN : 2807-9035
Volume 6, Number 4, November 2026
https://doi.org/10.47709/brilliance.v6i4.9646



(a) (b)

Figure 6. Authentication and access escalation process for private users: (a) authentication interface and (b) access escalation process

Figure 6 shows the authentication and access-escalation process for private users. Figure 6(a) shows the authentication interface used by the private client, while Figure 6(b) shows the subsequent authentication or access-escalation process. After successful authentication, the backend application identifies the client's IP address and generates the Dynamic Extended ACL rule required to activate SSH access. This process demonstrates that the authentication status is used as the condition for granting the additional SSH privilege to the private client.

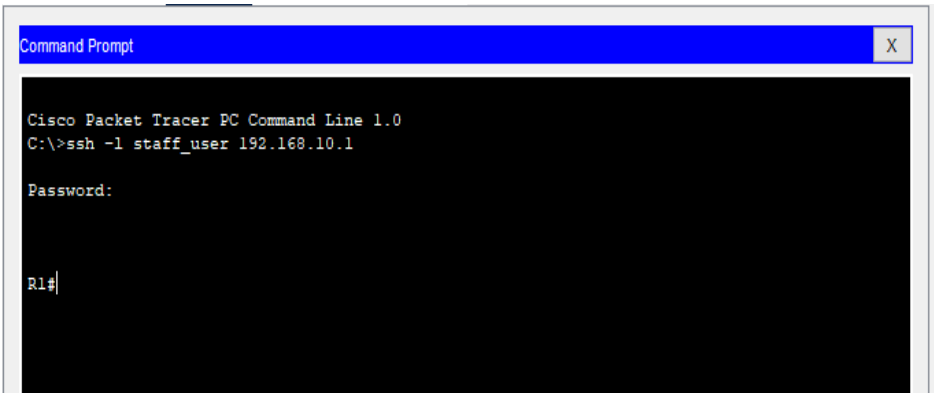


Figure 7. SSH access activated after successful authentication

Figure 7 shows the SSH access condition after successful authentication of the private client. The successful SSH connection indicates that the Dynamic Extended ACL rule for TCP port 22 has been activated and applied to the authenticated client's IP address. This result confirms that the model can selectively activate the additional SSH privilege without changing the other services included in the initial private-segment policy. The rule was automatically applied to the router through the Netmiko-based automation mechanism.

Table 4. Private-segment access before and after authentication

Service	Before Authentication	After Authentication
HTTP (TCP 80)	Permitted	Permitted
HTTP (TCP 443)	Permitted	Permitted
ICMP	Permitted	Permitted
FTP (TCP 20/21)	Permitted	Permitted
SSH (TCP 22)	Denied	Permitted

Table 4 shows that authentication affected only SSH access in the private segment. HTTP, HTTPS, ICMP, and FTP remained permitted before and after authentication, whereas SSH changed from denied to permitted. This result confirms that the Dynamic Extended ACL rule was applied selectively according to the predefined private-segment access policy.

The results show that the authentication process successfully increased the access privilege of the private client only for the service specified by the policy. SSH access was initially denied and became available after successful authentication. This demonstrates the ability of the model to implement service-specific privilege escalation rather than

This is an Creative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Brilliance
Research of Artificial Intelligence

E-ISSN : 2807-9035
Volume 6, **Number** 4, November 2026
https://doi.org/10.47709/brilliance.v6i4.9646

providing unrestricted access after authentication.

Automated Dynamic ACL Configuration

The integration between the authentication mechanism and the router was successfully implemented through Python Netmiko. After successful authentication, the backend application obtained the client's IP address and used it as the basis for generating the Dynamic Extended ACL configuration.

For public users, the generated policy activated ICMP and FTP access. For private users, the generated policy activated SSH access. The router received the configuration through an SSH session established by Netmiko, allowing the Dynamic ACL rules to be applied automatically.

```
commands = [
    'configure terminal',
    'ip access-list extended Dynamic_ACL',
    f'permit tcp host {client_ip} any eq ftp timeout 10',
    f'permit tcp host {client_ip} any eq 20 timeout 10',
    f'permit icmp host {client_ip} any timeout 10',
    'end',
    'write memory'
]
```

(a)

```
commands = [
    'configure terminal',
    'ip access-list extended Dynamic_ACL',
    f'permit tcp host {client_ip} any eq 22 timeout 10',
    'end',
    'write memory'
]
```

(b)

Figure 8. Dynamic Extended ACL rules activated through automated Netmiko configuration : (a) Public-segment (b) Private-segment

This implementation reduced the need for manual configuration changes on the router. The authentication event therefore became the trigger for modifying the client's network-level access privileges. The resulting process can be represented as:

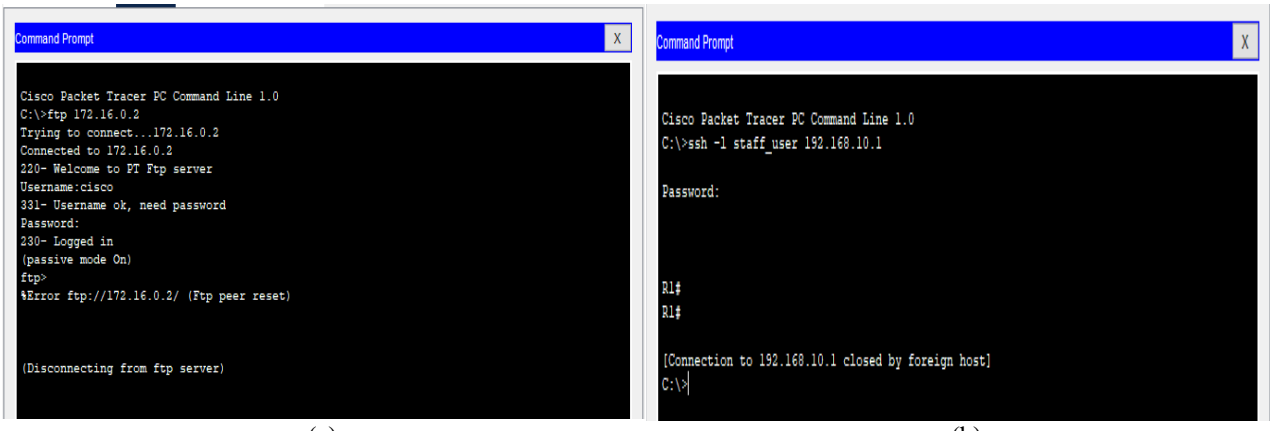
Client → Web Authentication → Authentication Verification → Client IP Identification → Dynamic ACL Generation → Netmiko/SSH → Router → Temporary Service Access

The implementation demonstrates that application-level authentication can be integrated with network-level policy enforcement to provide context-dependent access privileges.

Session Timeout and Access Termination

The developed model also implemented session-based access control through a configured 10-minute timeout for dynamically activated access privileges. The timeout mechanism was tested by allowing a client to obtain additional services through successful authentication and then leaving the dynamically activated access session inactive.

After the configured timeout period was reached, the temporary Dynamic Extended ACL entries were removed and the client's access privileges returned to the initial policy.



(a) (b)

Figure 9. Dynamic access termination after the configured idle timeout: (a) FTP Timeout (b) SSH Timeout

Figure 9 shows the termination of dynamically activated access after the configured timeout period. Figure 9(a) shows the FTP timeout condition for the public client, indicating that the previously activated FTP access is no longer available after the temporary access period ends. Figure 9(b) shows the SSH timeout condition for the private client, indicating that the dynamically activated SSH access has also been terminated. These results confirm that the temporary

This is an Creative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Brilliance
Research of Artificial Intelligence

E-ISSN : 2807-9035
Volume 6, Number 4, November 2026
https://doi.org/10.47709/brilliance.v6i4.9646

Dynamic Extended ACL entries are removed after the configured timeout and that the clients return to their respective initial access policies. Consequently, services that had been dynamically activated, such as ICMP and FTP for public users or SSH for private users, were no longer available without performing the authentication process again.

Table 5. Access behavior according to authentication and session status

User Context	Initial State	Authenticated State	After Idle Timeout
Public/Guest	HTTP/HTTPS	HTTP/HTTPS + ICMP + FTP	HTTP/HTTPS
Private/Staff	HTTP/HTTPS + ICMP + FTP	HTTP/HTTPS + ICMP + FTP + SSH	HTTP/HTTPS + ICMP + FTP
Administrative	Broad access	Broad access	Broad access

Table 5 shows the change in access privileges across three session states: initial, authenticated, and after timeout. Public users gained ICMP and FTP access after authentication, while private users gained SSH access. After the configured timeout, these additional privileges were removed and the clients returned to their respective initial access policies. This result confirms that the activated privileges were temporary and dependent on the authentication session.

The results confirm that the dynamically activated privileges were temporary rather than permanent. The timeout mechanism returned the client to its initial access policy after the session became inactive. Re-authentication was therefore required to obtain the additional privileges again

Summary of Functional Test Results

The functional testing demonstrates that the main components of the Adaptive Dynamic Extended ACL model operated according to the designed access-control policy. Public users initially received restricted service access and obtained additional ICMP and FTP privileges after successful authentication. Private users initially had access to HTTP, HTTPS, ICMP, and FTP, while SSH was restricted and subsequently activated following successful authentication. The dynamically activated privileges were automatically terminated after the configured timeout.

Table 6. Summary of functional implementation results

Test Component	Expected Result	Implementation Result	Status
Network-segment identification	Client identified according to VLAN/IP segment	Client context identified	Successful
Public initial access	HTTP/HTTPS permitted; ICMP/FTP restricted	Access matched policy	Successful
Public authentication	Additional ICMP/FTP access activated	Dynamic privileges activated	Successful
Private initial access	HTTP/HTTPS/ICMP/FTP permitted; SSH restricted	Access matched policy	Successful
Private authentication	SSH access activated	Dynamic SSH privilege activated	Successful
Netmiko automation	ACL rules applied automatically	Router configuration updated automatically	Successful
Session timeout	Temporary privileges removed after inactivity	Dynamic privileges terminated	Successful
Policy restoration	Client returned to initial access policy	Initial policy restored	Successful

Overall, the implementation results indicate that the developed Adaptive Dynamic Extended ACL model successfully integrated network-segment context, authentication, automated policy activation, service-specific authorization, and session-based access termination. The model was able to change access privileges dynamically according to the authentication state while maintaining the initial security policy for unauthenticated users.

DISCUSSION

The implementation results indicate that the proposed Adaptive Dynamic Extended Access Control List (ACL) model successfully transformed access control from a predominantly static mechanism into a conditional and temporary authorization process. The model was able to adjust service privileges according to the client's network segment and authentication status. This finding demonstrates that Extended ACL can be integrated with authentication and session-based mechanisms to provide more flexible access control without eliminating the basic packet-filtering function of ACL.

The differentiation of access privileges between public and private network segments demonstrates the role of network context in determining authorization. Public users initially received access only to HTTP and HTTPS services, while ICMP and FTP were restricted. After successful authentication, these additional services became available to the

This is an Creative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Brilliance
Research of Artificial Intelligence

E-ISSN : 2807-9035
Volume 6, **Number** 4, November 2026
https://doi.org/10.47709/brilliance.v6i4.9646

authenticated client. In contrast, private users were initially permitted to access HTTP, HTTPS, ICMP, and FTP, while SSH remained restricted until authentication was successfully completed. This result indicates that the developed model does not treat authentication as a mechanism for granting unrestricted access. Instead, authentication functions as a condition for activating specific privileges according to the predefined security policy. Such an approach can reduce unnecessary service exposure, particularly for users operating in public or less trusted network segments.

These findings are consistent with previous studies demonstrating the effectiveness of ACL and Extended ACL for controlling network traffic in LAN and VLAN environments (Azmi et al., 2022; Dewi et al., 2022; Usior & Sediyono, 2023). Recent research has also examined dynamic ACL and dynamic access-control mechanisms for adapting network authorization to changing access conditions (Jung et al., 2022; You et al., 2023). However, conventional implementations generally emphasize predefined access policies configured directly on network devices. The developed model extends this approach by linking the authentication state of a client to the activation of Dynamic Extended ACL rules.

The automation mechanism using Python Netmiko further supports this adaptive process. Following successful authentication, the backend application identifies the client's IP address and generates the corresponding Dynamic ACL configuration before transmitting it to the router through an SSH connection. The router consequently functions as the enforcement point for the authorization decision generated by the application. This mechanism reduces the need for repetitive manual configuration and enables access policies to be activated for individual clients based on authentication events, consistent with the role of Python-based network automation in reducing manual network-device configuration tasks (Gondhalekar et al., 2024).

The implementation of session-based access control provides another important security characteristic. Dynamically activated privileges were assigned a configured 10-minute timeout, after which the temporary access rules were removed and the client returned to the initial access policy (Cisco Systems, Inc., n.d.).

The main contribution of this study is therefore the integration of four components within a single access-control model: network-segment context, user authentication, automated service-specific Dynamic ACL activation, and session-based access termination. Previous ACL implementations have primarily focused on traffic filtering and service restriction, whereas the developed model establishes a relationship between user authentication and network-level policy enforcement. The model also demonstrates that adaptive access control can be implemented using commonly available networking and automation technologies without requiring unrestricted access to network resources after authentication.

Despite these findings, the study has limitations. The implementation and functional evaluation were conducted using Cisco Packet Tracer; therefore, the results primarily demonstrate functional feasibility within a simulated network environment and do not yet represent performance under production-scale network traffic. In addition, the authorization decision is currently based on network segment, client IP address, authentication status, and predefined service policies. Factors such as device identity, user behavior, risk level, and real-time traffic characteristics have not been incorporated. Future development can therefore extend the model toward more advanced contextual and behavior-aware access control.

Overall, the results demonstrate that the Adaptive Dynamic Extended ACL model provides a flexible mechanism for controlling temporary network service access while maintaining a restrictive baseline policy. The integration of authentication, automated Dynamic ACL enforcement, service-specific authorization, and session-based termination provides a practical foundation for developing adaptive data communication security mechanisms in computer networks.

CONCLUSION

The This study developed and implemented an Adaptive Dynamic Extended Access Control List (ACL) model for improving flexibility in data communication security. The developed model integrates network-segment identification, web-based authentication, automated Dynamic ACL activation, service-specific authorization, and session-based access termination within a unified access-control mechanism. The implementation using Cisco Packet Tracer demonstrated that access privileges could be dynamically adjusted according to the client's network context and authentication status.

The functional testing showed that public users initially received restricted access to HTTP and HTTPS services, while additional ICMP and FTP privileges could be activated after successful authentication. Private users initially had access to HTTP, HTTPS, ICMP, and FTP, while SSH access could be dynamically activated after authentication. The integration of Python Netmiko enabled the authentication process to trigger automated Dynamic ACL configuration on the router, reducing the need for manual policy modification. The implementation of a configured 10-minute timeout also ensured that temporary privileges were automatically terminated and the client returned to the initial access policy.

These findings indicate that the proposed Adaptive Dynamic Extended ACL model provides a practical approach for implementing temporary, service-specific, and context-aware network access control. Although the current evaluation was conducted in a simulated Cisco Packet Tracer environment, the developed model provides a foundation for further development toward more advanced adaptive access control by incorporating additional contextual factors,

This is an Creative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

E-ISSN : 2807-9035
Volume 6, **Number** 4, November 2026
https://doi.org/10.47709/brilliance.v6i4.9646

device identity, user behavior, and real-time network conditions.

ACKNOWLEDGMENT

The authors gratefully acknowledge the support of Politeknik Negeri Padang through the Penelitian Dosen Pemula (PDP) Research Grant Scheme for the 2026 Fiscal Year. The funding provided through this scheme supported the implementation and completion of this research.

REFERENCES

Anwar, R. S. (2023). Implementasi keamanan jaringan dengan metode Access Control List (ACL) pada RS Bhayangkara TK.I Pusdokkes Polri. *Akrab Juara, 8*(1), 33–41. doi:10.58487/akrabjuara.v8i1.2020

Arif, S. C., Surapati, U., Akbar, Y., & Hidayat, A. Z. (2025). Optimasi Access Control List (ACL) jaringan dalam menangkal akses ilegal jaringan Cisco. *Jurnal Indonesia: Manajemen Informatika dan Komunikasi, 6*(3), 1806–1816. doi:10.63447/jimik.v6i3.1606

Azmi, F., Kalsum, T. U., & Alamsyah, H. (2022). Analysis and application of Access Control List (ACL) methods on computer networks. *Jurnal Komputer, Informasi dan Teknologi, 2*(1), 81–88. doi:10.53697/jkomitek.v2i1.642

Gondhalekar, B., Manna, H., Kothi, S., & Borsae, D. B. (2024). Network automation with multithreading using GNS3 and Netmiko. *Indian Journal of Computer Science, 9*(2), 18–33. https://doi.org/10.17010/ijcs/2024/v9/i2/173860

Cisco Systems, Inc. (2025). *Configuring lock-and-key security (dynamic access lists)*. Cisco Systems.

Dewi, S., Firmansyah, F., & Hasan, U. (2022). Penerapan metode Access Control List pada jaringan VLAN menggunakan router Cisco. *Jurnal IMTechno, 3*(1), 37–41. doi:10.31294/imtechno.v3i1.927

You, H., Ko, D., Kim, D., Wong, R., & Joe, I. (2023). Dynamic access control method for SDP-based network environments. *Journal on Wireless Communications and Networking, 2023*, 94. https://doi.org/10.1186/s13638-023-02305-9

Jung, C., Kim, S., Jang, R., Mohaisen, D., & Nyang, D. (2022). A scalable and dynamic ACL system for in-network defense. *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, 1679–1693. https://doi.org/10.1145/3548606.3560606

Nawawi, S., Tahir, M., Ihsan, R. F., Valahi, A. D., Mufarrohah, M., & Rusmiati, S. (2024). Implementasi Access Control List (ACL) pada perangkat MikroTik untuk mengamankan koneksi jaringan internet di SMK Nurusshaleh Kato Timur. *JNKTI, 7*(4), 586–591. doi:10.32672/jnkti.v7i4.7704

Rahman, M., & Dasuki, M. (2025). Implementasi Access Control List (ACL) sebagai metode proteksi dan traffic control pada infrastruktur jaringan Local Area Network (LAN). *Jurnal CoSciTech, 6*(1), 68–76. doi:10.37859/coscitech.v6i1.9102

Rahman, T., & Aprianto, Q. (2025). Implementation of VLAN and ACL for network security at SDIT Ibnu Hajar Bekasi. *Journal of Electrical Engineering and Computer (JEECOM), 7*(2), 564–571. doi:10.33650/jeecom.v7i2

Tahir, M., Hariyanto, H., Firdausi, M. I., Saim, S., Nuriyah, N., & Maimunah, M. (2024). Peningkatan keamanan jaringan LAN dan WLAN melalui Standard Access Control List. *Jurnal Digital Transformation Technology, 4*(1), 607–614. doi:10.47709/digitech.v4i1.4261

Temoshok, D., Fenton, J., Choong, Y.-Y., Lefkovitz, N., Regenscheid, A., Galluzzo, R., & Richer, J. (2025). *Digital identity guidelines: Authentication and authenticator management* (NIST Special Publication 800-63B-4). National Institute of Standards and Technology. doi:10.6028/NIST.SP.800-63b-4

Usior, O. J., & Sediyono, E. (2023). Simulasi Extended ACL pada jaringan VLAN menggunakan aplikasi Cisco Packet Tracer. *AITI: Jurnal Teknologi Informasi, 20*(1), 32–47. doi:10.24246/aiti.v20i1.32-47

Wijaya, C. C., & Budiman, A. S. (2023). Perancangan keamanan jaringan komputer pada router dengan metode ACL pada PT. Aruna Sinar Jaya Jakarta. *JOURNAL ZETROEM, 5*(2), 180–186. doi:10.36526/ztr.v5i2.3077

This is an Creative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.