Automating SDN-ACLs with User Groups and Authentication Events

The Twenty-Third International Conference on Networks ICN 2024 Barcelona, Spain

Florian Grießer[1], Atsushi Shinoda[2], Hirokazu Hasegawa[3], Hajime Shimada[2]

[1]Technical University of Munich, Germany

[2]Nagoya University, Japan

[3]National Institute of Informatics, Tokyo, Japan

May 21, 2024

Florian Grießer

florian.griesser@tum.de

Professional Experience

- Security Software Engineer at Elektrobit Automotive
- Master Student in Electrical Engineering with focus on Embedded Security at Technical University Munich
- Member of the QuantumRisc Research Project for Post-Quantum Cryptography

Publications and Activities

- QuantumRisc: Work Package 6, Deliverables 6.1 and 6.2 Evaluation, integration and demonstration of use cases [1]
- Research about Information Set Decoding for Code-Based Post-Quantum Cryptography

[1] Heymann et al., *Evaluation, Integration and Demonstration of Use Cases*

Introduction

- Digital transformation has exponentially increased the complexity of network architectures
- Cybersecurity threats are becoming more sophisticated, exploiting modern infrastructures at an alarming rate
- Traditional security mechanisms, relying on static configurations and manual oversight, are proving inadequate

The Problem

- Traditional network security mechanisms are characterized by:
  - Inflexibility
  - Slow response times
- These limitations underline the urgent need for more adaptable, responsive security measures
- Dynamic and evolving threats require a new approach to network security

Software Defined Networking (SDN)

The Significance

- SDN separates network control logic from hardware, enabling centralized, programmable frameworks
- Enhances flexibility and adaptability in network management
- The SDN market was valued at USD 28.2 billion in 2023 and is projected to grow significantly at an annual rate exceeding 17% from 2024 to 2032[2].

Key Features

- Centralized Control: Simplifies network management
- Programmability: Allows for dynamic adjustments and rapid response to threats
- Flexibility: Adaptable to evolving security needs, as demonstrated by studies like Ali et al.[3]

[2] Global Market Insights, *Software Defined Networking (SDN) Market*
[3] Ali et al., *"Dynamic ACL Policy Implementation in Software Defined Networks"*

Proposed Solution and Contributions

- Automating ACL Generation:
  - Leveraging SDN for dynamic and responsive access control
  - Reducing manual configuration and errors
- Centralized Management:
  - Access control tied to user database, reducing manual work
  - Streamlined permissions based on user groups
  - Ensuring only authorized devices and users with 802.1X port-based access control
- Dynamic Adaptation:
  - Adjust ACLs based on real-time authentication events and traffic inspection
  - Respond to security incidents with immediate ACL modifications
- Empirical Validation:
  - Use cases demonstrating improved security and efficiency
  - Comparison with existing methods to highlight advancements

Outline

The Role of OpenFlow Switches in SDN

- SDN is a paradigm shift in network management.
- OpenFlow switches are a cornerstone, enabling:
  - Dynamic network control
  - Efficient routing
  - Robust access control mechanisms [4]
- Security challenges include:
  - Controller security
  - Flow table vulnerabilities [5]

[4] McKeown et al., *"OpenFlow: enabling innovation in campus networks"*
[5] Farooq et al., *"Security and Privacy Issues in Software-Defined Networking (SDN): A Systematic Literature Review"*

Strengthening Network Defenses with 802.1X

- IEEE 802.1X Port-Based Network Access Control:
  - Restricts network entry to verified devices/users
  - Ensures high network integrity and protection [6]
- Interaction sequence:
  - Initiation
  - Identity presentation
  - Authentication verification using EAP over LAN [7]
- Mitigates replay and impersonation threats

[6] IEEE, *"IEEE Standard for Local and Metropolitan Area Networks–Port-Based Network Access Control"*
[7] Vollbrecht et al., *Extensible Authentication Protocol (EAP)*

Access Management with AD and LDAP

- Active Directory (AD) [8] and Lightweight Directory Access Protocol (LDAP) [9]:
  - AD: Streamlines user/group/role management in Windows networks
  - LDAP: Provides a unified authentication framework across multiple services
- Centralized user database with:
  - Access Rights
  - User Groups and Roles
- Benefits:
  - Scalability
  - Ease of administration
  - Seamless integration

[8] Desmond et al., *Active Directory: Designing, Deploying, and Running Active Directory*
[9] Sermersheim, *Lightweight Directory Access Protocol (LDAP): The Protocol*

Previous Works: Problem and Approach

Building on our established framework for automating ACL generation through statistical analysis of communication patterns:

- Initial efforts were documented in the papers *"Construction of Secure Internal Network with Communication Classifying System Using Multiple Judgment Methods"* by Hasegawa et al.[10] and *"Construction of Secure Internal Networks with Communication Classifying System"* by Sato et al.[11]
- Key challenge: Authentication proof for IP addresses
- Leveraging Active Directory for user and group management[12]
- Streamlining the process and eliminating redundant tasks for administrators

[10] Hasegawa Hirokazu, *"Construction of Secure Internal Network with Communication Classifying System Using Multiple Judgment Methods"*
[11] Sato Yuya, *"Construction of Secure Internal Networks with Communication Classifying System"*
[12] Desmond et al., *Active Directory: Designing, Deploying, and Running Active Directory*

Architecture of the Proposed System

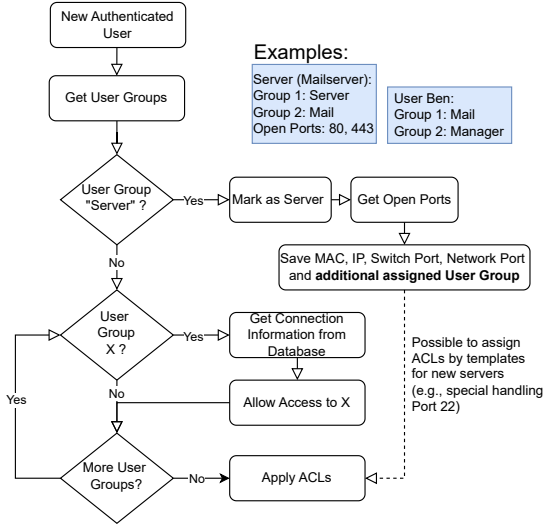
Enhancing network security through Port Access Control:

- Limits network port access to authenticated users
- Initial configuration allows only EAPOL messages of 802.1X to an authenticator component
- SDN Controller checks pre-configured users based on MAC and IP addresses
- Authentication messages forwarded to a Radius server
- Successful authentication assigns IP to MAC address via DHCP

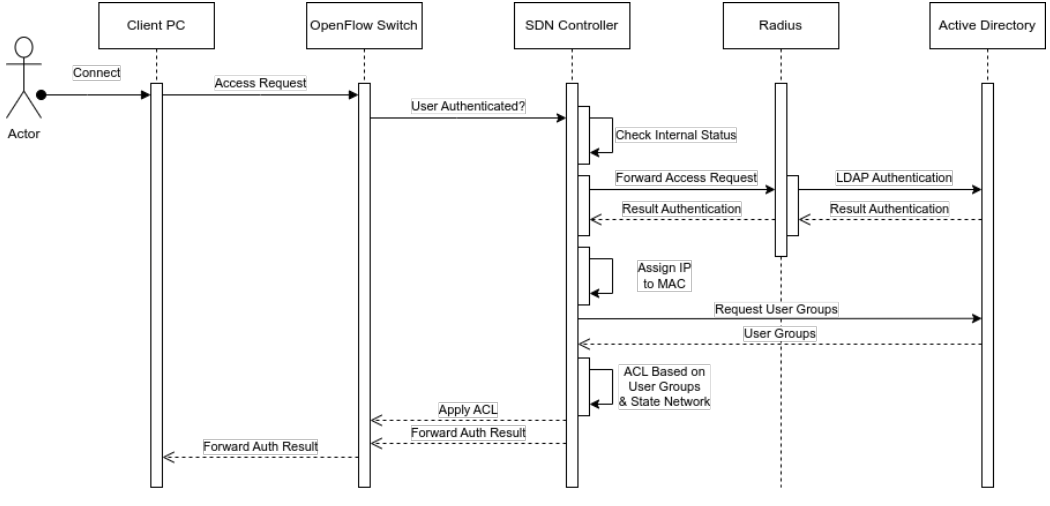
Architecture of the Proposed System

Generating User-Specific ACLs:

- Retrieve user groups from Active Directory via LDAP
- Create ACLs allowing traffic based on user group memberships
- Identify servers through unique identifier group
- Conduct port scans for servers to identify open ports and protocols
- Construct ACLs based on user groups and database information
- Only additional effort is assigning user groups to server



Initial ACL Generation Sequence



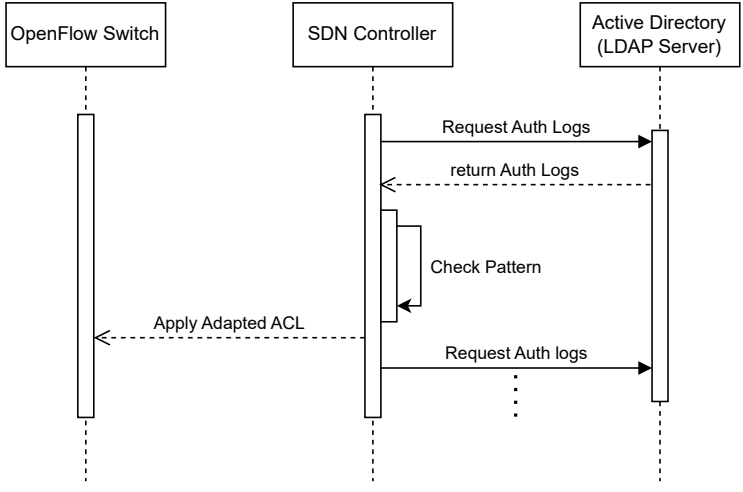
Authentication Log Approach

Dynamic ACL adaptation based on authentication logs:

- LDAP proxy monitors authentication activities
- Detect suspicious activities based on failed login events
- Modify network layout and ACLs for users with suspicious behavior
  - Block suspicious hosts
  - Redirect traffic to an Intrusion Detection System (IDS) for further analysis
- Inspired by Alert-ID [13] with direct network changes on malicious behavior detection

[13] Chu et al., *"ALERT-ID: analyze logs of the network element in real time for intrusion detection"*

Authentication Log Adaption



Quantitative Analysis of Access Control Lists

- N_{global}: Global number of ACLs
- G_i: Total number of groups for user i
- P_{g_i}: Number of ports for group g for user i
- U: Total number of users
- N_s: Number of ACLs per switch
- S: Set of users connected to the switch

Number of ACLs per User i

$$N_{u_i} = \sum_{g=1}^{G_i} P_{g_i} \qquad (1)$$

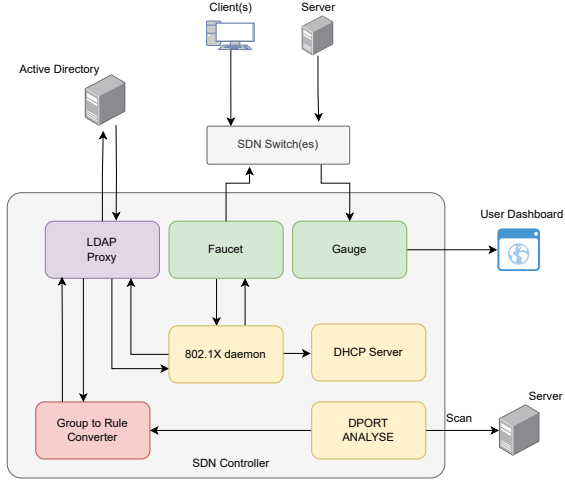
Global Number of ACLs

$$N_{global} = \sum_{i=1}^{U} N_{u_i} \qquad (2)$$

Number of ACLs per Switch

$$N_s = \sum_{i \in S} N_{u_i} \qquad (3)$$

Implementation Architecture



Example Whitelist ACL

- **Whitelist ACLs**: Allow specified traffic and block all other traffic to enhance security
- **Port-Specific Rules**: ACLs tied to specific ports ensure fine-grained control over network access
- **Scalability**: System can easily scale across multiple switches, applying ACLs to each port as needed
- **Dynamic Adjustments**: ACLs can be dynamically adjusted based on real-time authentication events and detected threats

```
acls:
  mac_whitelist_user_ben:
    - rule:
      dl_type: 0x800   # ipv4
      nw_proto: 6      # tcp
      tcp_dst: 80      # port
      eth_src:  32:90:43:57:f2:01
      ipv4_src: 192.168.0.1
      ipv4_dst: 192.168.0.9
      actions:
          allow: 1
    - rule:
        actions:
          allow: False
```

Experimental Conditions

Our experimental setup mirrors a realistic corporate network:

- Multiple physical PCs and servers
- Five Windows clients and one Linux client
- Clients assigned to different user groups in Active Directory

Client Name	Port	Source MAC	Groups
Client1	3	1C:69:7A:6D:C6:27	mail, gitlab
Client2	4	1C:69:7A:43:7C:12	mail
Client3	5	1C:69:7A:6D:C8:B0	mail, gitLab
Client4	6	1C:69:7A:6D:C7:EE	mail
Client5	7	1C:69:7A:6D:C8:16	mail

Feasibility

Demonstrating the feasibility of the proposed system:

- Verified system operability through multiple experiments
  - Experiment 1: Connection to the Network
  - Experiment 2: Failed Logins
- Example setup: Configured server and client access with dynamic ACLs

Experiment 1: Connection to the Network

Testing initial configuration and ACL application:

- Server and client connection with EAP authentication
- Dynamic ACLs generated based on user group memberships
- Example: Developer access to mail server and GitLab

Port	Source MAC	Source IP	Group	Destination IP	Destination Port	Description
3	1C:69:7A:6D:C6:27	192.168.11.11	mail	192.168.11.101	25, 993, 995	Mailserver
3	1C:69:7A:6D:C6:27	192.168.11.11	gitlab	192.168.11.102	22, 80, 443	GitLab
4	1C:69:7A:43:7C:12	192.168.11.12	mail	192.168.11.101	25, 993, 995	Mailserver
5	1C:69:7A:6D:C8:B0	192.168.11.13	mail	192.168.11.101	25, 993, 995	Mailserver
5	1C:69:7A:6D:C8:B0	192.168.11.13	gitlab	192.168.11.102	22, 80, 443	GitLab
6	1C:69:7A:6D:C7:EE	192.168.11.14	mail	192.168.11.101	25, 993, 995	Mailserver
7	1C:69:7A:6D:C8:16	192.168.11.15	mail	192.168.11.101	25, 993, 995	Mailserver

Experiment 2: Failed Logins

Evaluating the system's response to failed login attempts:

- Series of failed login attempts on GitLab server
- SDN Controller adjusts ACLs after six failed attempts
- Traffic mirroring to a specific port for further analysis
- Verified with tcpdump, simulating an IDS integration

Complexity and Efficiency

Comparing system complexity and efficiency:

Scenario	Faucet ACLs	OpenFlow Rules
No ACLs	0	27
Basic ACLs	8	67
VLANs	N/A	67
Dynamic ACLs	32	91

- **No ACLs:** Minimal rules, simplest configuration, but lowest security
- **Basic ACLs:** Rules for direct IP access, moderate complexity, improved security
- **VLANs:** Segments users into groups, similar complexity to Basic ACLs, but allows intra-department communication
- **Dynamic ACLs:** Most complex, highest number of rules, provides fine-grained access control, adapts to network changes efficiently

Comparison of different Systems

Metric	No ACLs	Basic ACLs	VLANs	Resonance [14]	ACL Based on X812 [15]	Proposed System
Security Level	Low	Medium	Medium	High	Very High	Very High
Port Security	None	None	None	None	Full	Full
Performance Impact	Low	Medium	Medium	Moderate	Moderate	Moderate
Scalability	High	Moderate	Good	Moderate	Moderate	Excellent
Manageability	Easy	Moderate	Moderate	Moderate	Moderate	Easy
Centralization	None	Low	Low	Medium	Medium	High
Flexibility	Low	Moderate	Low	Very High	Very High	High
Cost Efficiency	High	Moderate	Moderate	Low	Moderate	High
Integration Capability	Seamless	Moderate	Challenging	High	High	Low
Resilience	Low	Medium	Medium	High	High	High
Automation & Dynamic Response	None	None	None	Semi-Automated	Semi-Automated	Fully Automated
ACLs based on Authentication Events	None	None	None	None	None	Supported

[14] Nayak et al., *"Resonance: dynamic access control for enterprise networks"*

[15] A. Martins et al., *"An Extensible Access Control Architecture for Software Defined Networks based on X.812"*

Discussion

Summary of findings and implications:

- Enhanced security through automated, fine-grained whitelist ACLs
- Centralized security decisions ensure consistent access rights
- Scalability and ease of integration across multiple switches
- Performance maintained by focusing on TCP header inspections
- Balance between security measures and business continuity

Conclusion

- Implements a straightforward framework for **Port Access Control** to enhance security via fine-grained rules.
- Uses **Active Directory** for authenticating user access to network ports based on MAC addresses.
- **Mirrors traffic** from suspicious hosts to optimize IDS performance, targeting repeated login failures.
- **Future research** to focus on:
  - Optimizing algorithms for concurrent access requests.
  - Managing large rule sets without performance loss.
  - Evaluating traffic mirroring's impact on IDS efficiency.

References

[1] Maurice Heymann et al. *Evaluation, Integration and Demonstration of Use Cases.* Project Report Version 1.0. QuantumRISC Work Package 6, Deliverables 6.1 and 6.2. Darmstadt, Germany: Fraunhofer Institute for Secure Information Technology, Dec. 2023.

[2] Global Market Insights. *Software Defined Networking (SDN) Market.* https://www.gminsights.com/industry-analysis/software-defined-networking-sdn-market. Report ID: GMI2395, Accessed: 2024-02-14. 2024.

[3] Farrukh Shoukat Ali et al. "Dynamic ACL Policy Implementation in Software Defined Networks". In: *2022 International Conference on IT and Industrial Technologies (ICIT)*. Oct. 2022, pp. 01–07.

[4] Nick McKeown et al. "OpenFlow: enabling innovation in campus networks". In: *ACM SIGCOMM computer communication review* 38.2 (2008), pp. 69–74.

[5] Muhammad Shoaib Farooq, Shamyla Riaz, and Atif Alvi. "Security and Privacy Issues in Software-Defined Networking (SDN): A Systematic Literature Review". In: *Electronics* 12.14 (2023). ISSN: 2079-9292.

[6] IEEE. "IEEE Standard for Local and Metropolitan Area Networks–Port-Based Network Access Control". In: *IEEE Std 802.1X-2020 (Revision of IEEE Std 802.1X-2010 Incorporating IEEE Std 802.1Xbx-2014 and IEEE Std 802.1Xck-2018)* (2020), pp. 1–289.

[7] John Vollbrecht et al. *Extensible Authentication Protocol (EAP).* RFC 3748. June 2004.

[8] Brian Desmond et al. *Active Directory: Designing, Deploying, and Running Active Directory.* " O'Reilly Media, Inc.", 2008.

[9] Jim Sermersheim. *Lightweight Directory Access Protocol (LDAP): The Protocol.* RFC 4511. June 2006.

[10] Hiroki Takakura Hasegawa Hirokazu Yuya Sato. "Construction of Secure Internal Network with Communication Classifying System Using Multiple Judgment Methods". In: *International Journal on Advances in Telecommunications* 13.3 & 4 (2020).

[11] Hiroki Takakura Sato Yuya Hasegawa Hirokazu. "Construction of Secure Internal Networks with Communication Classifying System". In: *ICISSP*. 2019, pp. 552–557.

[12] Jie Chu et al. "ALERT-ID: analyze logs of the network element in real time for intrusion detection". In: *Research in Attacks, Intrusions, and Defenses: 15th International Symposium, RAID 2012, Amsterdam, The Netherlands, September 12-14, 2012. Proceedings 15*. Springer. 2012, pp. 294–313.

[13] Ankur Kumar Nayak et al. "Resonance: dynamic access control for enterprise networks". In: *Proceedings of the 1st ACM Workshop on Research on Enterprise Networking*. WREN '09. Barcelona, Spain: Association for Computing Machinery, 2009, pp. 11–18. ISBN: 9781605584430.

[14] Bruno José C. de A. Martins et al. "An Extensible Access Control Architecture for Software Defined Networks based on X.812". In: *2019 IEEE Latin-American Conference on Communications (LATINCOM)*. 2019, pp. 1–6.