You *et al. J Wireless Com Network* (2023) 2023:94
https://doi.org/10.1186/s13638-023-02305-9

EURASIP Journal on Wireless Communications and Networking

RESEARCH Open Access

Dynamic access control method for SDP-based network environments

Hyunjin You[1], Doochan Ko[1], Daniel Kim[1], Richard Wong[1] and Inwhee Joe[1*]

*Correspondence: iwjoe@hanyang.ac.kr

[1] Department of Computer Science, Hanyang University, Seoul, South Korea

Abstract

With online work environments and other distributed computing systems—such as cloud technologies or Internet of Things systems—becoming increasingly popular today due to the COVID-19 pandemic and general technological advances, the question of how to keep them secure has also become a pertinent concern. With this increased dependence on online systems for companies, cyberattacks have also been on the rise. To protect terminal devices, many companies have resorted to implementing a single boundary-defense model. This method has yielded positive results in securing the network from external threats, but it does not effectively protect network from internal security threats. With the vulnerabilities in the internal network security in mind, a dynamic access control method used with a zero-trust software-defined perimeter security model could be a viable solution. This study proposes a dynamic access control method using an engine with a new reward and penalty point-based system (RP Engine) and a dynamic task engine (DT Engine) for a zero-trust SDP security model.

Keywords: Dynamic access control, Software-defined perimeter, Security model

1 Introduction

Network security has always been an important aspect of the growing technology sector. With the constantly increasing usage of computers and computer networks in the world, it will continue to grow in importance [1–3]. This idea has been further reinforced by the increasing intensity and frequency of cyberattacks today, and they are projected to become worse as time goes on [4].

There are many ways to ensure network security, and there are also many aspects of network security that can be studied as well. When it comes to ensuring internal network security, there are a couple of different approaches that can be taken. Two prominent approaches are using a virtual private network (VPN) [5–7] or using an SDP [8, 9].

With a VPN, there is a network that exists to connect a device with a specific server—or servers. While VPNs can be useful in securing a network, it unfortunately still comes with some weaknesses [10–12]. VPNs can be troublesome to set up as they require complex network linkages, and can be more difficult to manage in terms of allowing multiple level accessing. If an unwanted user or attacker gains access to a VPN, it can become a

single point of failure vulnerability. This could result in them gaining access to an entire network.

An SDP is essentially a way to implement the zero-trust security model [13]. Its principle is, essentially, to 'never trust and always verify'. In other words, SDPs pre-validate and authenticate users—and their devices—and creates a specific network connection for the users in question. This approach, by vetting every user and device, arguably provides enhanced security. Additionally, it reduces potential attacking points since it lacks an accessible existing connection similar to a VPN. [14].

While this quick overview of what an SDP can do may make it seem straight forward, there are still many details to consider when it comes to actually setting them up. For example, it is known that SDPs work by pre-validating and authenticating users and their devices before creating a network connection [15]; however, there is the question of how it is actually implemented. There is also the question of how to grant users specific network access depending on what they need, as one of the main proponents of SDP is to grant users access to what they specifically need.

In light of the challenges regarding how SDPs are implemented and how users are granted specific network access, this study proposes a dynamic access control method using the RP engine and a dynamic task engine. The RP engine essentially grants access to users in a Least Privilege Rule framework. It assesses a user or device's access history to determine whether access should be granted or not in real time. The DT engine, on the other hand, handles the dynamic tasks in granting access to tasks to users—including non-predefined tasks.

To assess the effectiveness of the proposed method, a dynamic access control simulation is conducted. The assessment itself is done through a scenario analysis procedure, and its performance will be compared against the performance of traditional access control methods.

2 Proposed approach

2.1 SDP environment

To evaluate the performance of the dynamic access control system proposed in this paper, a theoretical SDP environment was set up for a simulation.

As Fig. 1 shows, there are multiple components involved in the proposed SDP with the RP and DT engines.

At the higher levels of the diagram, there are DT, RP, and access decision (AD) engines. As mentioned earlier, the RP engine is the proposed engine that will determine scores with reward or penalty points for anyone trying to access a network judging from their past access histories and general security status. The DT engine is the dynamic task engine which helps with providing the scope what a user or device is authorized to access in a network. The AD engine essentially delivers a list of objects that are accessible to authenticated users as determined by the DT and RP engines.

In the lower areas of the diagram are the SDP controller and its associated SDP blocks. SDP controller determines which SDP hosts can communicate with each other and deliver information about them to an external authentication service. The Initiating SDP Host (IH) block communicates with the SDP controller to request a list

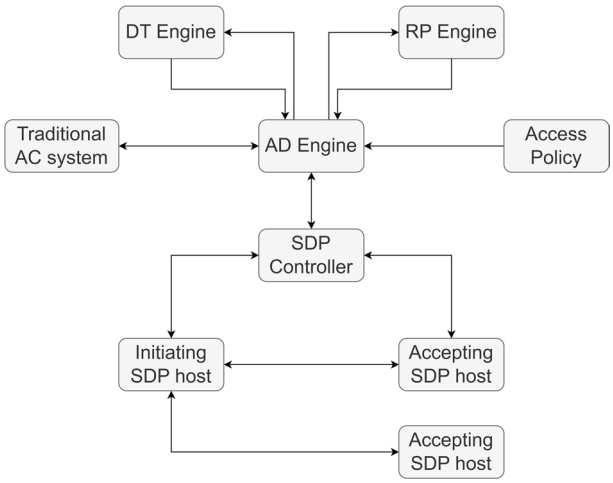

Fig. 1 Dynamic access control elements in the SDP environment

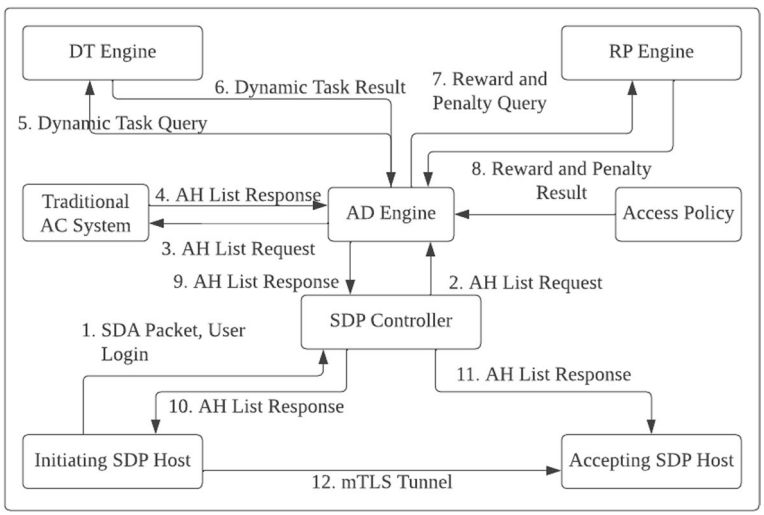

Fig. 2 Flowchart of the dynamic access control in the SDP environment

of accepting hosts (AH) to connect to. The SDP controller can also request further information about the hosts' software or hardware through the IH.

To actually accept a host, there is another block called the Accepting SDP Host (AH). The functionality of the AH is to essentially reject connections with all hosts and external networks except for the SDP controller itself and IH's authorized by the controller.

Figure 2 shows an overview of how the proposed SDP environment works. Particularly, it explains how each of the blocks interact with each other in an intuitive way.

The following paragraphs explain each of the steps of Fig. 2.

Step 1: When a user attempts to log in, an SPA packet is sent from the IH to the SDP controller for identification purposes and will execute the user login procedure as a whole.

Step 2: To initiate authentication steps, an AH list is requested from the AD engine. The authentication steps verify whether the user from the attempted login can be found within the AH list.

Step 3: After authentication steps are completed, the SDP controller requests user information and the AH list the terminal can access from the AC System.

Step 4: Upon receiving a request for an AH list from the SDP controller, the AD engine then requests an AH list from the existing (traditional) AC system. The AC system then responds with a list of AHs the user can access.

Steps 5 and 6: After receiving an AH list from the AC System, the AD engine then sends the DT engine a dynamic task query to determine the list of AHs required. The DT engine then responds with an AH list accordingly.

Steps 7 and 8: Then, the AD engine proceeds to send the RP engine a reward and penalty query to start calculating the score of the hosts from the AH list. The RP engine would then respond accordingly to the AD engine with the scores.

Step 9: When scores and an access scope are determined by the RP and DT engines, the AD engine then responds to the SDP controller's earlier request for an AH list.

Steps 10 and 11: The SDP controller relays this list to the Initiating SDP Host and Accepting SDP Host blocks.

Step 12: To access authorized AHs, the IH makes access through the mTLS tunnel protocol.

2.2 Reward and penalty engine

With the overall concept of how the proposed RP engine can be used in an SDP environment, it would be beneficial to explain exactly how the RP engine itself works.

As mentioned in previous sections, the RP engine is essentially used to dynamically calculate a score to see how safe a user is based on their previous access results and general security status. To explain the RP engine, some categories and terms used in the RP engine's scoring system should be explained. The details are shown in Table 1.

While the reward and penalty points used to score users are shown in Table 2, it should be noted that the factor of time is not accounted for on the table.

Imagine a scenario with two users—users A and B. Suppose user A might be given ten points ten days ago, and thirty points fifty days ago. Let us also say that user B was given thirty points ten days ago, and ten points fifty days ago. While the total sum of points users A and B may be the same (at sixty points accrued), user B should be considered to be more reliable than user A under the proposed engine.

To implement an RP engine that can also take into account the score trends over time of a user, weights need to be used in the scoring equation. To do so, more recent data

Table 1 Standard data definition

Category	Definition
RR (Reward for result)	Compensation points based on access history results
PR (Penalty for result)	Penalty points based on access history results
RD (Reward for device)	Compensation points based on the device
PD (Penalty for device)	Penalty points based on the device

Table 2 Reward and penalty points according to the approach result

Category	Criteria	Points
Reward	Successful and normal access	+1
	Appropriate security conditions	+2
	Antivirus program scans	+1
	Application of the appropriate security patch	+1
Penalty	SDP Gateway authentication failure	−2
	Attempting IH access without authentication	−2
	Three consecutive SDP Controller authentication failures	−1
	SDP Controller SPA authentication failure	−1
	Lack of an antivirus program	−2
	Malicious code detection	−2

points are given greater weights through the exponential moving average (EMA) formula. As the smoothing coefficient increases, more recent results would also have a greater influence on the EMA.

$$\mathrm{EMA}(t) = \mathrm{Value}(t) \cdot a + \mathrm{EMA}(t-1) \cdot (1-a) \tag{1}$$

Smoothing coefficient ($0 <= a <= 1$). If the smoothing coefficient is small, the effect of the EMA is smaller. If the smoothing coefficient is large, the effect of the EMA is greater.

While using an EMA does help with allowing more recent data to influence scores more, there is a limit to simply multiplying recent data with smoothing coefficient. To effectively distribute weights over time, it is necessary to adjust the EMA using the reward for result (RR), penalty for result (PR), reward for device (RD), and penalty for device (PD) values as shown in Fig. 2.

$$\mathrm{ERR} = \mathrm{RR}(t) \cdot \lambda + \mathrm{ERR}(t-1) \cdot (1-\lambda) \tag{2}$$

Reward adjusted for the EMA results.

$$\mathrm{EPR} = \mathrm{PR}(t) \cdot \lambda + \mathrm{EPR}(t-1) \cdot (1-\lambda) \tag{3}$$

Penalty adjusted for the EMA results.

$$\mathrm{ERD} = \mathrm{RD}(t) \cdot \lambda + \mathrm{ERD}(t-1) \cdot (1-\lambda) \tag{4}$$

Reward for the device adjusted for the EMA results.

$$\mathrm{EPD} = \mathrm{PD}(t) \cdot \lambda + \mathrm{EPD}(t-1) \cdot (1-\lambda) \tag{5}$$

Penalty for the device adjusted for the EMA results. λ is the correction factor for penalty points for past approach results.

There is also the matter of calculating reward and penalty point values for historical interactions between the user and AH. To do so, two types of histories are used to calculate the reward and penalty point values. For the reward values, the reward value for past access results (PRV), reward information value (RRI), and history value for total compensation (SRV) are calculated. The PRV is the reward value assigned based on past access results from connections between the AH and the user. The RRI is assigned based on the user's

historical reputation with other AHs. The SRV is assigned based on combined effects of the PRV and PRI.

$$\mathrm{PRV(User, AH)} = \left(\frac{\mathrm{ERR}}{(\mathrm{ERR} + \mathrm{EPR})}\right) \times \beta^{\frac{1}{\mathrm{ERR}+1}} \tag{6}$$

$$\mathrm{RRI(User, AH)}_i = \left(\frac{\mathrm{ERR}}{(\mathrm{ERR}_i + \mathrm{EPR}_i)}\right) \times \beta^{\frac{1}{\mathrm{ERa}+1}} \tag{7}$$

$$\mathrm{SRV(User, AH)} = \delta \mathrm{PRV(User, AH)} + \varepsilon \mathrm{RRI(User, AH)}_i \tag{8}$$

Here, δ is the PRV weight, and ε is the RRI weight. PRV has a direct effect on the user's access to AH_i, while the RRI can set different weights that have an indirect effect. β is the correction factor of compensation values for past approach results.

Since the rewards have now been calculated with the aforementioned equations to determine PRV, RRI, and SRV, the penalty points also need to be calculated through the PPV, RPI, and SPV values. These values are effectively the penalty point version of the reward values, and their calculations are fairly similar for each respective value category.

$$\mathrm{PPV(User, AH)} = \left(\frac{\mathrm{EPR}}{\mathrm{ERR+EPR}}\right) \times \beta^{\frac{1}{\mathrm{EPR}+1}} \tag{9}$$

$$\mathrm{RPI(User, AH)}_i = \left(\frac{\mathrm{ERR}}{ERP_i + \mathrm{EPR}_i}\right) \times \beta^{\frac{1}{\mathrm{ERa}+1}} \tag{10}$$

$$\mathrm{SPV(User, AH)} = \delta \mathrm{PPV(User, AH)} + \varepsilon \mathrm{RPI(User, AH)}_i \tag{11}$$

Here, δ is the PPV weight, and ε is the RPI weight. PPV has a direct effect on the user's access to AH_i, while the RRI can set different weights that have an indirect effect.

Traditionally, access control was mostly limited to the relationship between a subject and an object. However, with modern trends, access control is now done through various devices. With that in mind, it is necessary to assess whether a single device can impact overall reliability and risk for a network of multiple devices. To conduct this assessment, the stability of a device can also be assessed with reward and penalty points through the two equations below.

$$\mathrm{SRVD}(d) = \left(\frac{\mathrm{ERD}}{\mathrm{ERD} + \mathrm{EPD}}\right) \times \zeta^{\frac{1}{\mathrm{ERD}+1}} \tag{12}$$

$$\mathrm{SPVD}(d) = \left(\frac{\mathrm{EPD}}{\mathrm{ERD} + \mathrm{EPD}}\right) \times \eta^{\frac{1}{\mathrm{EPD}+1}} \tag{13}$$

Here, ERD and EPD are the reward and penalty values for the IH and AH. To prevent an inappropriate increase in reliability points when either reward or penalty points are missing, the weights ζ and η are multiplied and calculated.

3 Performance results and evaluation

To assess the proposed RP and DT engine's effectiveness, a theoretical simulation was conducted. Reward and penalty points were obtained according to the results from access to user AH. Different events were simulated, and the values used for the study are shown in Tables 3, 4, and 5.

To actually assess the performance of the RP engine, the raw values calculated were then accumulated for comparative purposes in a series of charts as illustrated in Fig. 3.

From the results in Fig. 3, it can be seen that the gap between penalty points and reward points widened and the difference became more negative over time. What this means is that the number of normal connections got overwhelmed by the number of abnormal connections in the environment. Additionally, as the simple traditional mechanism in SDP network environments does not consider the time factor, old rewards and penalties have the same weight as new rewards and penalties. Thus, it can be concluded that existing mechanisms in SDP network environments are ineffective.

After conducting the test using a traditional SDP, another test was run with slightly modified parameters. Access restrictions were strengthened by observing previous access results and security management scores. EMA as shown in Fig. 4 was used to increase the weight of reward and penalty points in accordance with their respective chronological orders. With the adjustments brought into effect through EMA, access

Table 3 Event history related to RPE test rewards/penalties

Run	Test contents	RR	RD	PR	PD
1	Successful and normal access	1			
2	Appropriate security conditions		1		
3	Antivirus program scans		1		
4	Application of the appropriate security patch		1		
5	SDP gateway authentication failure			2	
6	Attempting IH access without authentication			2	
7	Three consecutive SDP controller authentication failures			1	
8	SDP Controller SPA authentication failure			1	
9	Lack of an antivirus program				1
10	Malicious code detection				2

Table 4 Coefficients set in the experiment

Variable	Weight	Definition
α	0.30	EMA smoothing coefficient of rewards and penalties
β	0.30	Correction factor of reward values for past approach results
λ	0.40	Correction factor for penalty points for past approach results
δ	0.50	Weight for PRV, RRI
ε	0.05–0.1	Weight for PPV, RPI
ζ	0.30	Correction factor for the device compensation history SRV
η	0.30	Correction factor for the device compensation history SPV

Table 5 RPE experiment results

Number	Event	Event result				EMA of Event Result				PRV	PPV	Aceess history		Device history	
		RR	RD	PR	PD	ERR	EPR	ERD	EPD			SRV (user, ah)	SPV (user, ah)	SRVD (d)	SPVD (d)
1	1	1	0	0	0	1.000	0.000	1.000	0.000	0.548	0.000	1.041	0.000	0.548	0.000
2	1	1	0	0	0	0.750	0.147	1.059	0.000	0.420	0.090	0.765	0.211	0.557	0.000
3	9	0	0	0	1	0.525	0.103	0.741	0.300	0.380	0.087	0.672	0.305	0.357	0.114
4	6	0	0	2	0	0.368	0.672	0.519	0.810	0.147	0.427	0.528	0.507	0.177	0.313
5	10	0	0	0	2	0.257	0.470	0.363	1.167	0.136	0.403	0.497	0.553	0.098	0.438
6	3	0	1	0	0	0.480	0.329	0.554	0.817	0.263	0.242	0.632	0.371	0.186	0.307
7	4	0	1	0	0	0.636	0.231	0.688	0.572	0.352	0.151	0.515	0.454	0.268	0.211
8	1	1	0	0	0	0.738	0.027	0.744	0.286	0.483	0.018	0.473	0.546	0.362	0.109
9	6	0	0	2	0	0.517	0.019	0.521	0.500	0.436	0.018	0.568	0.372	0.231	0.220
10	10	0	0	0	2	0.362	0.313	0.365	0.350	0.221	0.274	0.534	0.408	0.211	0.201
11	1	1	0	0	0	0.253	0.219	0.255	0.540	0.205	0.263	0.457	0.464	0.122	0.312
12	4	0	1	0	0	0.177	0.154	0.179	0.682	0.193	0.254	0.344	0.629	0.075	0.387
13	9	0	0	0	1	0.124	0.107	1.025	0.477	0.184	0.248	0.436	0.482	0.377	0.141
14	3	0	1	0	0	0.408	0.003	0.875	0.434	0.422	0.004	0.751	0.159	0.352	0.143
15	3	0	1	0	0	0.286	0.602	0.612	0.304	0.126	0.440	0.477	0.524	0.317	0.132
16	6	0	0	2	0	0.200	0.421	0.489	0.812	0.118	0.417	0.563	0.418	0.149	0.337
17	9	0	0	0	1	0.140	0.295	0.300	1.169	0.112	0.397	0.566	0.378	0.081	0.457
18	1	1	0	0	0	0.398	0.207	0.510	0.818	0.278	0.192	0.700	0.237	0.173	0.381
19	10	0	0	0	2	0.579	0.145	0.657	0.573	0.373	0.109	0.557	0.371	0.258	0.217
20	2	0	1	0	0	0.589	0.001	0.589	0.711	0.468	0.001	0.564	0.386	0.212	0.271
21	4	0	1	0	0	0.412	0.001	0.412	0.798	0.426	0.001	0.526	0.416	0.145	0.337
22	8	0	0	1	0	0.589	0.000	0.589	0.558	0.468	0.000	0.641	0.307	0.241	0.225
23	6	0	0	2	0	0.412	0.000	0.412	0.691	0.426	0.000	0.651	0.240	0.159	0.307

Table 5 (continued)

Number	Event	Event result				EMA of Event Result				PRV	PPV	Aceess history		Device history	
		RR	RD	PR	PD	ERR	EPR	ERD	EPD			SRV (user, ah)	SPV (user, ah)	SRVD (d)	SPVD (d)
24	8	0	0	1	0	0.288	0.600	0.288	0.434	0.127	0.438	0.590	0.377	0.147	0.278
25	10	0	0	0	2	0.502	0.420	0.502	0.338	0.244	0.280	0.689	0.280	0.268	0.164
26	3	0	1	0	0	0.351	0.594	0.351	0.237	0.152	0.407	0.608	0.325	0.245	0.152
27	9	0	0	0	1	0.546	0.416	0.546	0.166	0.261	0.265	0.539	0.361	0.352	0.083
28	4	0	1	0	0	0.399	0.421	0.703	0.088	0.206	0.315	0.377	0.623	0.438	0.037
29	10	0	0	0	2	0.280	0.295	0.492	0.662	0.190	0.300	0.491	0.465	0.190	0.278
30	3	0	1	0	0	0.196	0.206	0.345	1.063	0.178	0.289	0.351	0.647	0.100	0.421

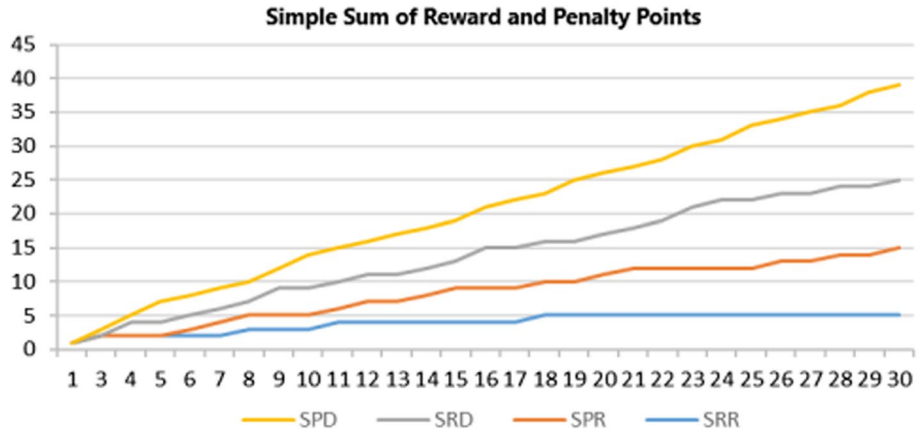

Fig. 3 Simple sum of reward and penalty points

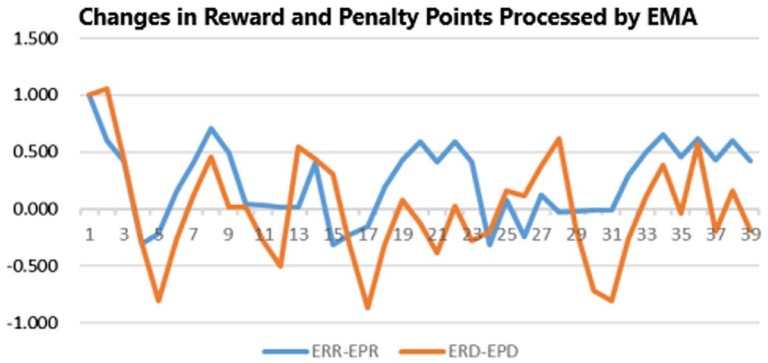

Fig. 4 Changes in reward and penalty points processed by EMA

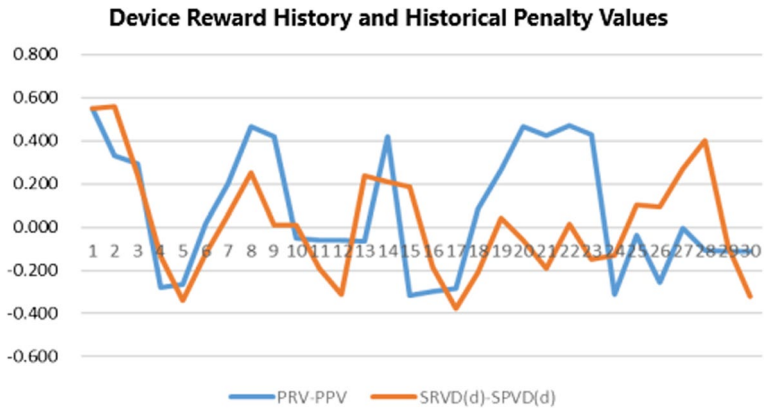

Fig. 5 Device reward history and historical penalty values

restrictions appeared to get strengthened. It should be noted, however, that there is a problem in predicting the trend of access according to the chart. This happened because the difference between reward and penalty points fluctuated greatly due to more recent events in the simulation.

After the previous tests, a final test was run with the proposed new RP Engine. Figure 5 observes the difference between PRV and PPV values, as well as the device reward history SRVD and the device penalty history SPVD between the user and AH for the final test. When the graph starts going down to negative values, it means existing access rights were getting deprived. The graph looks slightly smoother in comparison with the graph from figure 3.2, meaning there was a better consistency in results.

4 Conclusion

Current network connection methods can expose server information during remote access sessions and may potentially even allow access to other network resources beyond the session. These are the kinds of severe security issues that are alleviated through the SDP environment proposed in this study. The SDP environment assures security by essentially only allowing designated devices to join designated services.

Another issue tackled by the proposed SDP environment is the issue of authorizing specific scopes of access to users through a dynamic task engine. In a traditional access control system, the scopes of access are heavily predefined, and so users would not be able to access different parts of the network at all. However, with the proposed environment, users can be safely granted access through risk assessments and the dynamic access control mechanism.

Results from the study have also shown that the proposed solution of using RP and DT engines in an SDP environment can work effectively. Values calculated from the study indicated that the system's performance was enhanced as there was a greater number of normal connections compared to the number of abnormal connections to the system.

Abbreviations

AC	Access control
AD	Access decision
AH	Accepting SDP host
DT	Dynamic task
EMA	Exponential moving average
EPD	Penalty for device adjusted with EMA
EPR	Penalty for result adjusted with EMA
ERD	Reward for device adjusted with EMA
ERR	Reward for result adjusted with EMA
IH	Initiating SDP host
PD	Penalty for device
PR	Penalty for result
RD	Reward for device
RP	Reward and penalty
RR	Reward for result
SDP	Software-defined perimeter
SPA	Single packet authorization
VPN	Virtual private network

Acknowledgements

This work was supported by the Institute for Information and Communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (Development of the technology to automate the recommendations for big data analytic models that define data characteristics and problems), under Grant 2020-0-00107.

Author contributions

HY and Doochan Ko proposed the method, drafted the manuscript, and carried out experiments. Daniel Kim and RW helped to draft the manuscript. IJ supervised this work. All authors read and approved the final manuscript.

Funding
Institute for Information and Communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (Development of the technology to automate the recommendations for big data analytic models that define data characteristics and problems)

Availability of data and materials
The data used to support the findings of this study are available from the corresponding author upon request.

Declarations

Competing interests
The authors declare that they have no competing interests.

Received: 7 February 2023 Accepted: 7 September 2023
Published online: 18 September 2023

References

1. K. Shaukat, S. Luo, V. Varadharajan, I.A. Hameed, M. Xu, A survey on machine learning techniques for cyber security in the last decade. IEEE Access **8**, 222310–222354 (2020). https://doi.org/10.1109/ACCESS.2020.3041951
2. K.A.P. da Costa, J.P. Papa, C.O. Lisboa, R. Munoz, V.H.C. de Albuquerque, Internet of things: A survey on machine learning-based intrusion detection approaches. Comput. Netw. **151**, 147–157 (2019). https://doi.org/10.1016/j.comnet.2019.01.023
3. M.M. Dhanvijay, S.C. Patil, Internet of things: a survey of enabling technologies in healthcare and its applications. Comput. Netw. **153**, 113–131 (2019). https://doi.org/10.1016/j.comnet.2019.03.006
4. E. Bout, V. Loscri, A. Gallais, How machine learning changes the nature of cyberattacks on iot networks: a survey. IEEE Commun. Surv. Tutor. **24**(1), 248–279 (2022). https://doi.org/10.1109/COMST.2021.3127267
5. P.J. Ezra, S. Misra, A. Agrawal, J. Oluranti, R. Maskeliunas, R. Damasevicius, Secured communication using virtual private network (vpn), in *Cyber Security and Digital Forensics*. ed. by K. Khanna, V.V. Estrela, J.J.P.C. Rodrigues (Springer, Singapore, 2022), pp.309–319
6. F. Hauser, M. Häberle, M. Schmidt, M. Menth, P4-ipsec: Site-to-site and host-to-site vpn with ipsec in p4-based sdn. IEEE Access **8**, 139567–139586 (2020). https://doi.org/10.1109/ACCESS.2020.3012738
7. M. Juma, A.A. Monem, K. Shaalan, Hybrid end-to-end vpn security approach for smart iot objects. J. Netw. Comput. Appl. **158**, 102598 (2020). https://doi.org/10.1016/j.jnca.2020.102598
8. A. Moubayed, A. Refaey, A. Shami, Software-defined perimeter (sdp): State of the art secure solution for modern networks. IEEE Netw. **33**(5), 226–233 (2019). https://doi.org/10.1109/MNET.2019.1800324
9. J. Singh, A. Refaey, A. Shami, Multilevel security framework for nfv based on software defined perimeter. IEEE Netw. **34**(5), 114–119 (2020). https://doi.org/10.1109/MNET.011.1900563
10. S. Jahan, M.S. Rahman, S. Saha, Application specific tunneling protocol selection for virtual private networks. In: 2017 International Conference on Networking, Systems and Security (NSysS), pp. 39–44 (2017). https://doi.org/10.1109/NSysS.2017.7885799
11. A.K. Singh, S.G. Samaddar, A.K. Misra, Enhancing vpn security through security policy management. In: 2012 1st International Conference on Recent Advances in Information Technology (RAIT), pp. 137–142 (2012). https://doi.org/10.1109/RAIT.2012.6194494
12. B. Lipp, B. Blanchet, K. Bhargavan, A mechanised cryptographic proof of the wireguard virtual private network protocol. In: 2019 IEEE European Symposium on Security and Privacy (EuroS &P), pp. 231–246 (2019). https://doi.org/10.1109/EuroSP.2019.00026
13. M. Campbell, Beyond zero trust: Trust is a vulnerability. Computer **53**(10), 110–113 (2020). https://doi.org/10.1109/MC.2020.3011081
14. Cloudflare: What is a software-defined perimeter? | SDP vs. VPN (2023). https://www.cloudflare.com/learning/access-management/software-defined-perimeter/. Accessed Accessed 04 Feb 2023
15. A. Sallam, A. Refaey, A. Shami, On the security of sdn: a completed secure and scalable framework using the software-defined perimeter. IEEE Access **7**, 146577–146587 (2019). https://doi.org/10.1109/ACCESS.2019.2939780

Publisher's Note