Research on SDP Software Defined Perimeter Initiating Host Protocol Configuration Algorithm

Xinmeng Xie, Gang Gan and Yong Chen
School of Cyberspace Security, Chengdu University of Information Technology, Chengdu 610225, China
Email: 452617485@qq.com

Abstract. With the popularity of digital and cloud services, the demand for network security products and solutions is increasing day by day. At the same time, the network security problems are becoming more and more serious. The network security architecture and platform are far from meeting the challenges brought by the current situation. As a new generation network security solution concept, the Software Definition Perimeter was first proposed by the Cloud Security Alliance in 2013. The whole central idea is to build a virtual enterprise boundary in the mobile + cloud era through software. The use of identity-based access control is to cope with the problem of coarse control and poor validity caused by boundary fuzziness, so as to achieve the purpose of protecting data security. This paper mainly changes the infrastructure of the Software Defined Perimeter SDP, and adds the message queue, which mainly stores the control information in the message queue to solve the peak congestion and network congestion problem.

1. Introduction
The usual network architecture consists of a fixed boundary that isolates the internal network from the external network. This boundary includes a series of firewalls to withstand the entry of external users, but allows internal users to access externally. Traditional fixed boundaries increase the internal server's resistance to external threats by disconnecting external visibility and accessibility to internal applications and devices [1]. But now because mobile smart device terminals and attack software provide untrusted access to the internal network, the fixed boundary model in the network architecture becomes less powerful.

Therefore, it is imperative to create a new type of security model that needs to be able to understand contextual information, determine the user's location, devices, and connections established, and when devices are connected. This information can be integrated into a context. In the access rules, authentication checks based on context parameters and access to resources can provide better protection against threats inside and outside the boundary. Using the SDP architecture to protect the server from being invisible to unauthorized users can eliminate some attacks, including brute force attacks, network traffic attacks, and so on[2-3]. The SDP architecture is a great way to achieve the zero-trust principle. SDP uses pre-authentication and pre-authorization as its two basic pillars. By authenticating and authorizing users and devices before a single-packet authorization reaches the target server, SDP can be in the network. The principle of least privilege is enforced on the layer to encrypt network traffic, so that all resources can be securely accessed regardless of the user's location.
With the diversification of certification and authorization and the increase in the number, the development will inevitably affect the basic processing performance of the SDP security architecture. In order to improve its processing performance, this paper introduces the security challenges faced in contemporary networks, and SDP as a potential and effective solution, the concept and framework structure are basically described. The basic principle of queuing is also used to mathematically model the allocation process of SDP security architecture protocol, and then theoretical analysis is carried out. The Prim algorithm is used to simulate the allocation of the protocol, and the protocol processing performance of the SDP security architecture is discussed.

2. Software Defined Perimeter SDP

2.1. Summary

The Software Defined Perimeter (SDP) was proposed by the Cloud Security Alliance (CSA) [4] in 2013. The application owner controls the logical components to replace the physical devices. Only after device authentication and identity authentication, SDP provides the application basis and access to the facility. SDP enables application owner-deployed boundaries to maintain invisibility and inaccessibility to external users in traditional models, which can be deployed anywhere, such as on the network, in the cloud, in a hosting center, on a private enterprise network, or also deployed in these locations. The SDP authenticates the device, authenticates and authorizes the user, ensures two-way encrypted communication, and dynamically configures the service.

2.2. Architecture

The system architecture of Software Defined Perimeter SDP in this paper is shown in figure 1.

![Architecture Diagram of Software Defined Perimeter SDP](image)

**Figure 1.** Architecture Diagram of Software Defined Perimeter SDP

The SDP architecture consists of three parts: the SDP controller, the SDP Initiating host, and the SDP accepting host. The SDP controller communicates with all SDP components and acts as an administrator. The SDP initiates communication with the controller and requests a list of available connections. The controller can request other information from the originator, such as authentication and status, and receive a list of accessible gateways or hosts based on this response. The SDP accepts the host to reject all connection requests except the controller. It provides a connection to the user and application that the controller has identified. The controller can be linked to various authentication methods (eg RADIUS, PKI, OpenID, OAuth, and LDAP) and can request other status information. The initiating host connected to the architecture is connected to the SDP controller and authenticated [5-7]. After the initiating host is authenticated, the SDP controller determines a list of accepting hosts that can authorize the initiating host based on the policy, and the SDP controller provides the receiving host with the host. The host list is initiated, and the initiating host sends a single packet data to the authorized receiving host to perform a two-way encrypted connection.

3. SDP Deployment Model

This article describes the client-to-gateway model as shown in Figure 2.
The protected server is behind the host and accepts the host as a gateway. In this model, the SDP controller and host use a single package of authorization to protect against unauthorized users. This model allows the protected server to be included in the SDP without any changes. It also preserves the ability of existing network security components such as IDS / IPS to operate unchanged, each device establishing a mutual transport layer security (MTLS) secure connection with the controller for authentication and control messages. In this model, the client connects directly to the gateway through the MTLS tunnel, and regardless of the underlying network topology, traffic is directly and securely performed between the two.

4. SDP Architecture Analysis and Modeling

4.1. Initiates Host Protocol Request Corresponding Protocol Configuration Analysis

In the SDP architecture, control information and data information are separated from each other and are not interfered with each other to prevent potential network-based attacks. In the SDP security architecture, according to different startup hosts having their own request services, the controller in the SDP architecture selects the corresponding protocol processing module and processing protocol for the startup host, then in the SDP architecture, the startup host port needs to be established. This paper quotes the basic knowledge of queuing theory to mathematically model the allocation process of the protocol processing module in the SDP architecture, analyzes and evaluates it according to the mathematical model, and then gives the relevant algorithm for the configuration of the protocol module. Under the time limit of the processing protocol, the allocation time of the protocol processing module required by the service can be optimized by the module allocation method of the queuing principle in the protocol processing mode, thereby saving related resources and improving the security performance of the SDP.

In the physical network, the processing of the protocol is divided into different network layers, and the data is layer-by-layer wrapped in the sending information field to form a complete information packet. However, for the SDP architecture, the request for data from different business requirements is different, and the hierarchically integrated protocol stack architecture makes the data processing very complicated. Therefore, the SDP architecture has a data package for the protocol stack and dynamic programming controllable operation process. Figure 3 visually shows the SDP architecture to the virtual protocol development process, the protocol is divided into multiple protocol function blocks, and then the virtual protocol stack is built according to the request of the host, and then the protocol block is allocated to the client corresponding to the SDP. Therefore, the SDP controller configures different protocol function blocks according to different initiating host requirements to obtain corresponding protocol function modules from multiple protocol stacks. Then, when the user logs in to the client in the SDP architecture, the user implements the configuration of the personalized virtual
protocol stack, completes the identity authentication, authorization, and other services that need to confirm the identity of the client.

**Figure 3. SDP Protocol Processing Diagram**

Compared to the past, users start the host through SDP to access resources in the entire system. The SDP boot host utilizes proven standards-based protocol components such as mutual TLS, SAML and X.509 certificates, IAM, PKI services, setup credentials, geolocation, SAML, OpenID, LDAP, multi-factor authentication, etc. Protocol, authentication and authorization.

### 4.2. Queuing Model for Protocol Processing

From the above analysis, we can see that to improve the allocation efficiency of the protocol processing module, this paper uses the queuing model [8-9], which is the mathematical theory and method to study the stochastic service process. Its wide range of applications can be applied to all systems with random service, covering all industries, from field production to factory processing, as well as daily travel or commonly used electronic communication equipment. In this paper, the queuing system is studied and summarized and classified according to several main features of the queuing system, as shown in Figure 4.

**Figure 4. Main Features of the Queuing System**

The most basic feature of the input process is the probability distribution of arrival time, which reflects the rule of the customer arriving at the queue. It obeys the exponential distribution and the Poisson distribution. The characteristics of the organization mainly cover the number of services. In multiple services, the services are arranged in parallel or serial. Whether the customer is a service or a batch service and a certain distribution to be followed by the service. Such as the exponential distribution.
The queuing rule is a question indicating the status of the guest when they are queuing, and the rules to be followed, such as the structure of the queue and the order in which the customer handles the service. Classify the three categories when the customer enters the queue, waiting system, loss system and mixed system. Waiting system is when the customer arrives at the service window, if all the service windows are fully occupied, and then the customer will queue up to wait for the service window. Loss system is when the customer arrives, if all the service windows are occupied, the customer will leave immediately instead of waiting in line for the service. Mixed system is it is divided into long and time limits. If the queued customers exceed the long time, then the customers who arrive later will automatically leave; if the waiting time is too long, the customer will leave the service system.

In the queuing system mentioned above, its characteristics can be combined in various ways to form a diverse queuing system. Then use the Kendall symbol \((A/B/C)-(d/e/f)\) to indicate a way of queuing. \(A\) represents the time when the customer arrives at the interval, \(B\) represents the time distribution of the service time of the service window, \(C\) represents the number of service windows, \(d\) represents the maximum number of customers in the queuing system, \(e\) represents the total number of customer sources and \(f\) stands for service rules, such as first-come, first-served, post-first, priority, etc. For example \((M/M/1)-(\infty/\infty/LCFS)\) means that the customer's arrival time and service are subject to exponential distribution, a service window, customer source and system capacity are infinite, and the service rules are post-first service.

In addition, several main performance indicator parameters of a queuing system. The system queuing service is divided into service pair length \(L_s\) and queue length \(L_q\). The service pair length indicates the number of clients being served, and the queue length indicates the number of clients waiting in the queue. The entire team leader is \(L=L_s+L_q\). The customer's stay time \(W_s\) indicates the time the customer was served and the waiting time \(W_q\) indicates the time spent waiting in the queue [10]. The total waiting time is \(W=W_s+W_q\).

5. Queuing Theory Applies to Protocol Processing of the SDP Architecture

In the SDP architecture, the SDP controller performs different configurations on the protocol function modules according to service requests of different IH clients. We assign a corresponding queuing model to the protocol function module according to the queuing principle. The relevant parameters used are shown in Table 1.

| Parameter | Significance |
|-----------|--------------|
| \(N\)     | The total number of SDP initiated host requests |
| \(M\)     | Total number of SDP protocol function blocks |
| \(L\)     | Total number of SDP architecture protocol service points |
| \(i\)     | Initiate host request number |
| \(j\)     | Protocol module number |
| \(s\)     | Protocol service site |
| \(X_i\)   | Request sample for initiating host |
| \(Y_j\)   | Protocol module sample |
| \(\lambda_j\) | Requested arrival rate |
| \(\mu_j\) | Processing efficiency of the j-th block protocol |
| \(L_S\)   | Protocol service site sample |
| \(T\)     | Time delay maximum |

In the input process, in the SDP security architecture, the SDP controller corresponds to the protocol service site sample \(L_s\), the total number is \(L\); the SDP architecture can be configured with a
total of M class protocol processing modules, the sample is represented by \( Y_j \); SDP The request sample on the IH side of the architecture is \( X_i \), and there are \( N \). Then the input part of the data protocol in the SDP security architecture is represented by an \( R \) (Request): \( I = R_{nm} \), where \( R_{nm} \) is the request matrix, the matrix of which is as follows:

\[
R_{nm} = R_{ij} = \begin{bmatrix}
    r_{11} & \cdots & r_{rm} \\
    \vdots & \ddots & \vdots \\
    r_{n1} & \cdots & r_{nm}
\end{bmatrix}
\]

The expression \( R_{ij} \) represents the initiating host request matrix, and represents the \( j \) type protocol function module corresponding to the \( i \)-th initiating host.

\[ R_{ij} = \begin{cases}
    1, & \text{The } j \text{th protocol module corresponding to the } i \text{th request of the initiating host} \\
    0, & \text{else}
\end{cases} \]

In the input part of multiple protocol type requests, the types of protocol function module requests corresponding to the initiating host of different requirements are different, then the \( R \) request matrix can be expressed as an input part in the SDP architecture, that is, multiple initiating hosts continuously The service request is sent to the controller, and the queue model appears on the service time window model, then the number of protocol modules of each class can be allocated according to the \( R \) matrix, thereby providing corresponding services. Through the above analysis, the process of protocol function allocation, the process can be simulated by \( M/M/C/\infty/\infty/FCFS \) in the queue model. The state transition analysis of the protocol queue is as follows: the letter \( i \) can be used to represent IH and \( j \) represents the protocol function module. Then the request arrival rate of IH is \( \lambda_i \), the processing rate of the protocol function module \( j \) is \( \mu_j \), and the number of the \( j \)-th protocol provided to the initiating host \( i \) is. When the \( j \) type protocol processes the \( i \) th request data, it can be expressed as the state transition of the queuing model as shown in Figure 5.

![Figure 5. Protocol Queue State Transition Diagram of Simulated Queuing Model](image)

If the queue system is to meet the steady-state requirements, then its protocol processing capability is greater than the ability of the data to arrive, otherwise it is a divergent queue. Then ask for \( \rho_{ij} = (\lambda_i/(C_j \mu_j)) \) \( < 1 \). Then, by the queue model \( (M/M/C)-(\infty/\infty/FCFS) \), it can be inferred that the staying time expression of the initiating host \( i \) is:

\[
W_{ij} = \frac{c_{ij}^{c_{ij}}}{c_{ij}! \mu_j (1-\rho_{ij})} \rho_{ij} \sum_{m=0}^{c_{ij}} \frac{c_{ij}^m}{m!} \rho_{ij}^m + \frac{c_{ij}^{c_{ij}+1}}{c_{ij}! (1-\rho_{ij})^2} \rho_{ij}^{c_{ij}+1}
\]

Therefore, its model can be classified as a cascade of multi-level \( (M/M/C)-(\infty/\infty/FCFS) \) queuing models. So the total length of stay of \( i \) in the queue \( W_{ij} \) is:

\[
W_{ij} = \sum_{j=1}^{M} W_{ij}, \; \forall i \in A_i
\]

The data stays in the network for less than a certain upper limit. We can set it to T, then there is:

\[
W_i \leq T, \; \forall i \in A_i
\]

In the output section, the SDP fabric controller responds to the protocol request and configures the protocol module into different request protocols, so its output process can be expressed \( O_{\text{out}} \) in a matrix. In the SDP architecture, the configuration of the protocol module that initiates host \( i \) can be represented by \( O_{\text{out}} = (O_{i1}, O_{i1}, \ldots O_{i\lambda}) \). The meaning of \( O_{ij} \) is the number of the \( j \)th protocol
modules allocated by the i-th host, that is, the optimal solution. This is the queuing model analysis process of protocol module allocation.

6. Protocol Allocation Policy

6.1. Allocation Algorithm
Through the above analysis, in the SDP architecture, the protocol request of the SDP originating host and the allocation of the protocol function processing block can be analyzed by the queuing model theory, and then the W staying time and network of the originating host according to the parameter Delay value comparison to optimize protocol allocation. In this task, first assign a protocol processing module to each request protocol, then calculate the total stay time of this time, and then compare with the system time limit. If it is less than the system time limit, then generate an allocation plan, if it is greater than the system time limit, then add a protocol processing module, then recalculate the total stay time, and then compare until the requirements are met. The essence can be attributed to the knapsack problem in the classic problem, and the local optimal solution is found to achieve the overall optimal solution. Then you can use a variety of algorithms, such as prim algorithm [11], dynamic programming algorithm, shortest path algorithm, etc. to find the local optimal solution.

6.2. Analysis and Implementation of Distribution Algorithm
Through analysis, the configuration system of the whole protocol is a knapsack problem, that is, the process of achieving the overall optimal solution by pursuing the local optimal solution. This idea has a variety of algorithms. This paper will first use the prim algorithm as an allocation strategy for protocol matching, thereby improving the performance of its allocation and making a time analysis graph of the queued queue. The principle of the prim algorithm is that G= (V, E) is an undirected Unicom weighted graph, that is, a network. The weight of each edge (v, w) in E is c[v] [w]; if a subgraph G1 of G is a tree containing all the vertices in G, G1 is called a spanning tree of G. The sum of the edge rights on the spanning tree is called the cost of the spanning tree. Among all the spanning trees in G, the least expensive spanning tree is called the minimum spanning tree [12]. In this paper, the number of Cij protocols is the weight of each edge. The least weight is the least time-consuming. Finding the minimum weight value of each IH is the shortest protocol processing module.
Algorithm 1 PrimeAA allocation algorithm

Input: $R_{NM}$
Output: $O_{NM}$

1. for $i=1$ to $N$
   2. Assign a protocol processing module to each type of protocol in the request
   3. $C_{ij}^{c_{ij}} \frac{c_{ij}^{m} m!}{\sum_{m=0}^{c_{ij}^{m} m!}} \frac{c_{ij}^{m} m!}{\sum_{m=0}^{c_{ij}^{m} m!}} \frac{\lambda_{i}}{(1-\rho_{ij})^{2}} \rho_{ij} c_{ij}^{m} + \frac{C_{ij}^{c_{ij}}}{C_{ij}^{c_{ij}}(1-\rho_{ij})^{2}} - 1$

4. $W_{i} = \sum_{j=1}^{M} W_{ij}$
   5. if ($W_{i} \leq T$) 
      6. Generate allocation strategy
   7. else
      8. for $j = 1$ to $M$
         9. Add 1 protocol processing module $j$ to the allocation policy
      10. $C_{ij}^{c_{ij}} \frac{c_{ij}^{m} m!}{\sum_{m=0}^{c_{ij}^{m} m!}} \frac{c_{ij}^{m} m!}{\sum_{m=0}^{c_{ij}^{m} m!}} \frac{\lambda_{i}}{(1-\rho_{ij})^{2}} \rho_{ij} c_{ij}^{m} + \frac{C_{ij}^{c_{ij}}}{C_{ij}^{c_{ij}}(1-\rho_{ij})^{2}} - 1$
      11. $W_{i} = \sum_{j=1}^{M} W_{ij}$
      12. Find the smallest value of VecW in the array, and write down its serial number $num$
      13. Add 1 protocol processing block $num$
      14. Update the system and go back to calculating the stay time of $W_{i}$
      15. End if
   16. End for

6.3. Simulation

System parameter setting, there are many types of protocol types to be requested at the IH side, but here the protocol type is set to 10. The rate of protocol module processing is several hundred thousand times per second, and the system delay is set to $\mu$. As shown in Table 2.

| Simulation parameter | Simulated range value |
|----------------------|-----------------------|
| Initiating host request maximum value $N$ | 1000 |
| Protocol processing type $M$ | 10 |
| Requested arrival rate $\lambda$ | 1000000–1000000 |
| Protocol module processing rate $\mu$ | 1000000 1500000 2000000 |
| System delay $T$ | 1.0$\mu$s, 2.0$\mu$s |

This paper mainly compares greedy and random algorithms. Compare their anti-real results. Other parameters are fixed, only in the case of initiating host-side request changes, the impact on the processing module samples, which is set $\lambda = 500000$, as shown in Figure 6.
In the figure, we can find that when I initiates the request sample and protocol module of the host, the red line of the primeAA algorithm is the minimum spanning tree result and the blue is the randomly generated connection graph superior. Simultaneously, this paper also compares the arrival and departure times of the protocol in the queuing principle, as shown in Figure 7.

As shown in Figure 7, the service time performance of the queuing system using the primeAA algorithm allocation protocol is simulated. The blue curve in the figure represents the time when the protocol module arrives in the queue, and the red curve represents the end time of leaving the queue after the protocol is processed. By comparing the two curves, the more coincident the two curves are, the better the protocol allocation performance of the SDP security architecture [13].

7. Conclusion
In this paper, the theoretical concept of the software definition boundary SDP of the new security architecture concept, the framework composition, the workflow and the deployment model are studied and analyzed. At the same time, through the basic analysis of the SDP security architecture, the basic principle of the SDP architecture is used to launch the host. The mathematical model analysis and simulation are carried out in order to better optimize the regulation and configuration of the protocol, so as to better optimize and demonstrate the SDP architecture, and improve the resource utilization and security performance of the system.
8. References

[1] Huang Chao. Application of Network Firewall Technology in Campus Network Security [J]. Heilongjiang Science, 2018, 1(9):132-133.

[2] Chen Fei. Research on DDoS Attack Defense Method in SDN [J]. Information and Computer, 2018, 12: 216-217.

[3] Simon Parkinson, Mauro Vallati, Andrew Crampton, Shirin Sohrabi. GraphBAD: A general technique for anomaly detection in Security information and event management [J]. Concurrency and Computation: Practice and Experience, 2018, 16(30)

[4] CLOUD SECURITY ALLIANCE(April 2014), Software Defined Perimeter Working Group, SDP Specification 1.0

[5] Kerner, Sean Michael. Cryptzone AppGate Update Secures the Software Defined Perimeter [J]. eWeek, 2017:1-1.

[6] Vizard, Michel. What Software-Defined Security Could Mean for the Channel [J]. Channel Insider, 2013:1-2.

[7] A. Mubayed, A. Refaey, and A. Shami, “Software-Defined Perimeter (SDP): State of the Art Secure Solution for Modern Networks,” IEEE Network Magazine, 2019.

[8] Zhou Shijie. Research on Queuing Theory Model of Message Middleware [J]. Computer Science, 2002, 29(11): 29-32.

[9] Li Wei, Qiang Yan, Chen Junjie. Performance Evaluation of Distributed System Based on Hierarchical Queuing Network Model [J]. Computer Development & Application, 2009, 22(3): 56-58.

[10] Zheng Guangjie. A message processing method and device based on message queue system: CN107197015A [P]. 2017.

[11] Dairina, Arif S, Munzir S, et al. Modification of Prim’s algorithm on complete broadcasting graph[C]. Journal of Physics Conference Series. 2017:012012.

[12] Hong-Mei G, Chao X, Ben-Cheng Y. Design and Analysis of Minimum Spanning Tree in Euclidean Plane[C]. International Conference on Computational & Information Sciences. 2017.

[13] M. Rash, “Single Packet Authorization: A Comprehensive Guide to Strong Service Concealment with fwknop”. 1–30, 2016.