Zener

Distributed Software Defined Firewalls

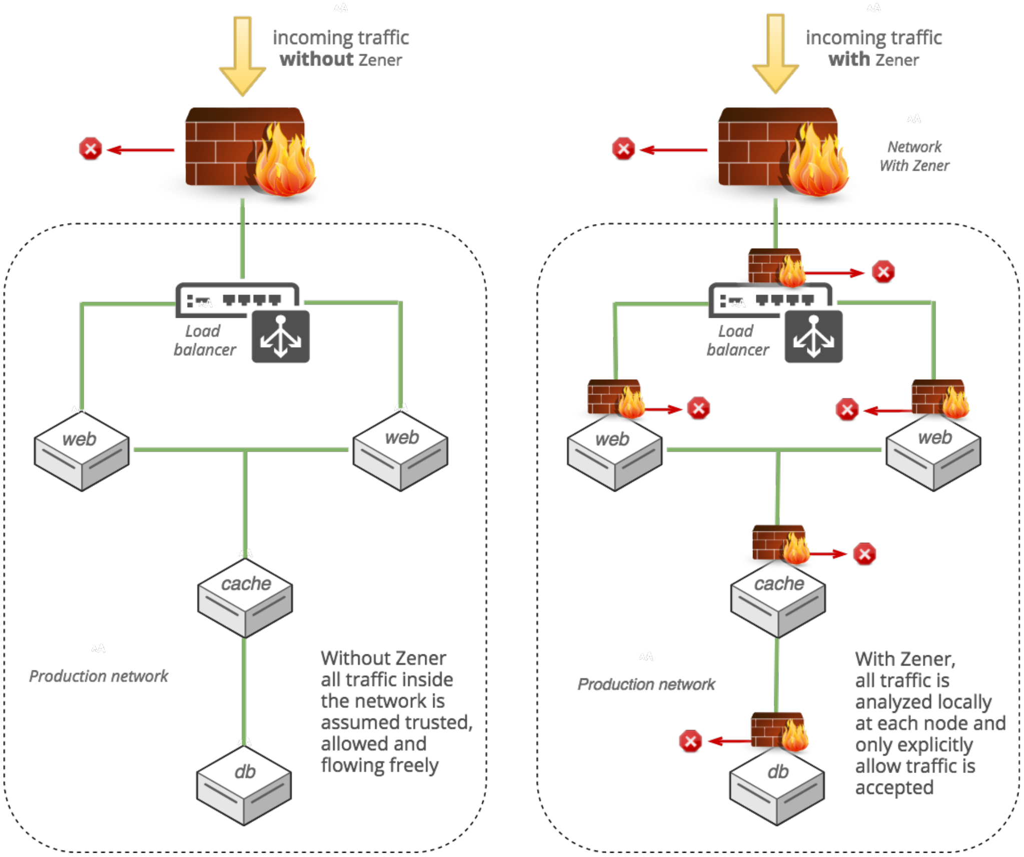
A TECHNICAL WHITE PAPER

Summary

Zener is a distributed Software Defined Firewall (SDF) solution. Zener protects global IT infrastructure by managing local host-based firewall technology. Zener ensures that centrally defined model-based firewall rules are correctly generated, distributed and enforced throughout the network. Zener makes it easy and fast to dynamically segment the network and apply firewall rules based on a wide range of options, from static values such as subnet to fully dynamic attributes.

This paper covers what Zener is and for whom it is intended before explaining the architecture and suggesting best-practices when designing a distributed firewall solution using Zener.

Last part of the paper gives a glimpse into the ideas for the future and how to get involved to impact the further development.



What is Zener

Zener is a distributed Software Defined Firewall (SDF) solution. It uses the state of the art local host-based, software-based firewall technologies, like nftables, iptables, Berkeley Packet Filter, etc. to control traffic on individual nodes.

Zener protects assets (data and applications) locally by distributing and enforcing firewall rules on each host. These centrally defined, simple, dynamic and unified rules can be applied to any layer 3-7 of the OSI-model.

At low cost, Zener adds an extra layer of security. Being cross-platform (Linux, Windows, AIX, HP-UX, etc.), Zener is well suited for complex and hybrid IT-infrastructures.

Zener allows for rapid growth without increase in Firewall complexity as one only need to think about local firewall rules. Automation takes human errors, and scaling pains out of the loop.

Zener is completely API driven and offers both CLI and a GUI interface in addition to API access.

Zener supports highly granular and dynamic definitions of groups, where firewall rules can be applied on for example hosts running a specific application, hosts belonging to a subset, etc.

Zener offers complete flexibility in designing a fully distributed SDF solution. The core of this technology has been running in production at large scale installations with more than 100,000 physical hosts and holds a reputation for being very scalable and secure.

Who is Zener for

Zener seeks to address the need of *mid- to large-scale IT-infrastructures* looking to enhance their security measures by utilizing automation and dynamic network segmentation.

Zener offers a new alternative to IT-organizations looking for ways to continue to grow while reducing the rising hardware cost and inherent inflexibility of existing network topology solutions. Zener supports the shift towards the flattened datacenter, and thereby reduction in the need for expensive proprietary network hardware. A flattened network topology further moves the need for firewalls from large and complex hierarchical models to distributed local and easy to understand firewalls.

For organizations aspiring to use the latest collaborative DevOps tools, Zener, with its open APIs is a good fit. Integrating Zener with CI/CD processes and thereby making security part of the release pipeline offers great benefits in terms of security and operational efficiency.

Zener provides a simple and transparent way for security teams to collaborate with DevOps, NetOps and SysAdmin. Managers at all levels can access granular traffic, application, asset and compliance reports.

Large-scale Firewall management made easy

Zener allows IT-Infrastructures to grow without adding complexity to the firewall rules. For traditional firewall solutions, growth typically implies increased firewall rule complexity. With Zener unlimited growth is supported, while the complexity remains unchanged. Whether a firewall rule applies to 1 or 100,000 web servers doesn't matter. The humans only need to deal with one rule (eg. allow port 443 on web servers). Zener automation ensures that this one rule is correctly applied to all the nodes regardless of their attributes and what subnet they belong to. Traditional large-scale IT-organizations can expect and reduction in the number of firewall rules by an order of magnitude.

Zener supports both IPv4 and IPv6 and can provide a simple way for many organizations to move from IPv4 to enable full IPv6 support.

Zener is cross-platform. It runs equally well on-premise as in the public cloud. Zener's decentralized architecture easily supports globally distributed datacenters and multi-cloud architectures.

A big pain with scale relates to cost of network hardware. With Zener, the firewalls often are part of the native operating system. This reduces the need to buy new expensive hardware firewalls, or expensive switches and the burden of dealing with complex vendor-specific TCAM rules.

Zener allows security to easily grow with IT-infrastructures, at no additional hardware costs, while reducing complexity.

Architecture

Zener is a client-server architecture, with decentralized decision-making.

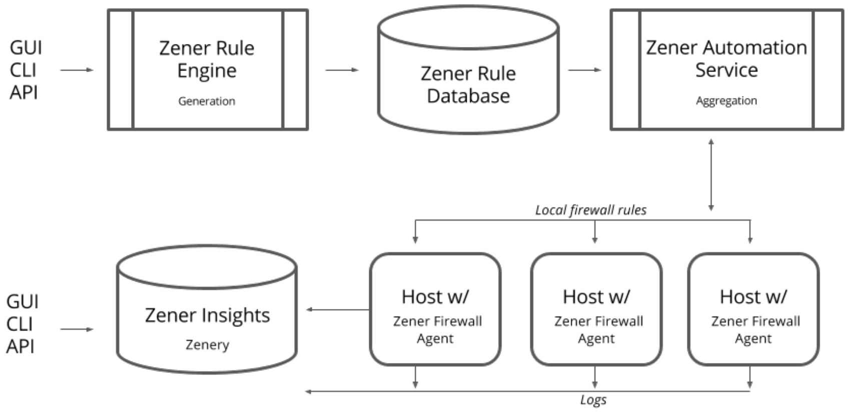
The Zener Rule Engine (ZRE) is a service where global firewall rules, and network segmentation is defined. This information is stored in the Zener Rule Database (ZRD).

The Zener Automation Service (ZAS) ensures the distribution of correct firewalls rules to the correct local nodes.

Zener Firewall Agent (ZFA) re-applies/evaluates the local firewall ruleset at each run (by default every 5 minutes). This way the local firewall will always conform with the definitions from the Zener Rule Engine.

Incoming data traffic is monitored locally and frequently sent to the Zener Insights Service (ZIS) backend analytics engine for reporting and debugging.

Architecture: From user defined rules to granular Insights



The architecture of Zener supports all kinds of network topologies. It works well in a traditional hierarchical network topology but is optimized for the next generation horizontally scalable (flattened) datacenter. Based on research done by the largest data consumers in the business. Facebook has written in depth about their new data center architecture [1], and LinkedIn is dramatically changing how they build data centers, too [2].

Zener is at the forefront providing a distributed SDF as the flattened datacenter starts to proliferate.

Components

Below follows a brief explanation of the key components in Zener.

Zener Rule Engine (ZRE)

The Zener Rule Engine (ZRE) is a service for defining firewall rules. The default setup of Zener implements a policy of “Deny All Inbound Traffic”. This means that all traffic needs to be explicitly defined as Rules, and added to the specific Group or Node (see below).

A Rule would be a typical firewall rule consisting of ports to be open, protocols that are used, but would only have a receiver, not a recipient. This is because all rules are processed at the end Node, not some random place in the middle that is unaware of the context. This makes the rules simpler to manage and maintain, and faster to process. Finally, this approach supports the Zero Trust Model. ZRE is written in Python.

Zener Rule Database (ZRD)

The Zener Rule Database (ZRE) is a Postgres database service where all user-defined firewall rules and network segmentation definitions are stored.

Zener Automation Service (ZAS)

The Zener Automation Service (ZAS) translates globally defined firewalls rules and network segmentations into locally applicable firewall rule sets. It ensures that the correct firewall syntax is being used depending on what firewall technology will be used at the local Node.

Zener Firewall Agent (ZFA)

The Zener Firewall Agent is a light-weight agent that runs on each host in the network. ZFA enable each host to maintain an up to date set of rules and definitions, and both check for updates and provide the alive signal to the central hub at specified intervals. The Zener Automation Service also collects information for use in Zenery Insights directly from each node.

The agent software is written in C and Python, and it is the most mature, secure and scalable agent technology in the industry, known to automate the largest and most complex IT-infrastructures in the world.

Zenery Insight Service (ZIS)

Distributed SDF increases knowledge and understanding a system has about the topology and data flow. With a “Deny by default” on inbound traffic and allowing all outbound traffic, debugging becomes feasible. The reason for a packet drop or rejection only stems from a node. One can inspect the individual node’s information and look up the firewall rules to understand the reason for packet rejections.

The Zener Insights Service (ZIS) provides detailed insights to each component of the network: the rules, groups, and individual nodes and their interdependencies and relations.

This is based on the powerful reporting and analytics features of the Zener Automation Service. It is written in PHP and JavaScript.

Network segmentation

Zener allows for micro-segmentation based on static and or dynamic rules and attributes. It does this by combining Nodes and Groups.

Node

The Node is the autonomous unit of computation. This can be a traditional physical server, but it can also be a Virtual Machine, a container or Unikernel. The node can reside on a public cloud infrastructure or on-premises. Each Node manages its own local firewall, with rules defined in the Zener Rules Engine and implemented by the Zener Firewall Agent.

Node Attributes

Node Attributes allows defining Firewall rules based on attributes of the Node. Whereas traditional firewall rules definitions normally are constrained to ip-addresses and subnet, or other network specific properties, Zener offers a flexible, truly Software Defined way of defining and managing firewall rules with Nodes Attributes.

Group

A Rule will be assigned to a Group of Nodes. A Group can consist of one single Node, thousands of Nodes, or Groups of Groups. The Group definition is the fundamental unit to create security zones and bundle nodes together. All nodes belonging to a Group will receive rules designated to this Group, and thereby themselves.

Zener supports creating Groups based on a number of different criteria, such as IP addresses, subnet and powerful Node Attributes, such as hostname, the presence of a file, content of a specific file on the node, or a command output.

Best practices - Zero Trust

Applying a distributed SDF challenges traditional firewall thinking and encourages moving security from a centralized complex to a decentralized simpler problem. To get started, below follows some getting started principles to help understand the potential and how Zener can be applied.

- “Deny all inbound traffic” as default. All new nodes should, by default, deny all inbound traffic. Only explicitly defined traffic should be allowed.

- “Allow all outbound traffic” as default. To keep complexity down, all outbound traffic can be allowed

Use Groups as the basis for Security Zones like Production, Test, Corporate, and more granular network segmentation. Groups are collections of nodes or node attributes.

Global Rules

All the nodes in the network should comply with a basic set of baseline security and global operations rules. One such rule could be to allow for SSH, so that even though all traffic is blocked one always has the option to login to the node.

In Group Rules

To implement a concept such as a security zone, you can define a set of In Group Rules. All Nodes within this one Group are allowed to communicate, while traffic from the outside become constrained. Building Groups that are limited in complexity makes it simpler to have a good overview of which applications talk to each other and makes the rules that cover the majority – often around 95% – of traffic simpler to manage and monitor. These rules are standard, and most nodes will be part of these.

Out of Group Rules

Define strict and limited ways applications or micro services can communicate with each other across groups. These rules should be an exception and will constitute a minimal amount of network traffic. These rules are mainly “human” made and maintained.

Finally, when designing a distributed SDF, remember to layer the network and all rules according to a holistic idea and architecture. Groups should represent efficient, and small units that covers your network. Groups should have minimal overlap of Nodes. And Rules that allow communication outside of a Group should be the exception and kept at a minimum.

Current Status

Zener is not yet officially released. The product is currently in a pilot phase. The current version of Zener, although limited in functionality and reporting capabilities, is fully functional and able to work at scale across hybrid IT-infrastructures.

A hosted demo is available. Users interested in Zener are welcome to test it out.

Get Involved

Zener has been developed in close collaboration with System Administrators, and DevOps at some of the largest IT-infrastructures. Our goal is to include more members for close collaboration. If you are interested, please drop us an email: contact@zener.io.

Resources

1. https://code.facebook.com/posts/360346274145943/introducing-data-center-fabric-the-next-generation-facebook-data-center-network/

2. https://engineering.linkedin.com/blog/2016/03/project-altair--the-evolution-of-linkedins-data-center-network