|  |  |  |
| --- | --- | --- |
| Cisco-logo-picture | Document Number | EDCS-23036895 |
| Based on Template | [EDCS-19601156](https://docs.cisco.com/share/proxy/alfresco/url?docnum=EDCS-19601156) Rev 2.5 |
| Created By | ZTNA Team |

Clientless ZTNA on FTD  
Software Functional Specification

Clientless Zero Trust Network Access Support on FTD Platforms.

Reviewers

|  |  |
| --- | --- |
| Department | Name/Title |
| Development Engineering |  |
| Development Manager | Rekha Puvvada |
| QA Engineering |  |
| QA Manager |  |
| Product Marketing | Ashley Pilbeam, Andrew Ossipov |
| Customer Advocacy |  |

Modification History

|  |  |  |  |
| --- | --- | --- | --- |
| Revision | Date | Originator | Comments |
| 1 | 15-Mar-22 | George Koikara | Initial Draft |
| 2 | 12-Oct-22 | Naveen Gujje & Pruthvi Panyam Nataraj | Updated software architecture and end user sections |
| 3 | 29-Oct-22 | George Koikara | Updated overview and architecture |
| 4 | 4th-Nov-22 | Pruthvi Panyam Nataraj | CLI for app groups and other updates |
| 5 | 5-Nov-22 | George Koikara | Addition of ZT Application Policy |
| 6 | 7-Nov-22 | Vikramaditya Singh | Updated the FMC section |
| 7 | 7-Nov-22 | George Koikara | Corrected EDCS number |
| 8 | 8-Aug-23 | Punit Shukla | Updated REST endpoints |

Table of Contents

[1 Problem Definition 4](#_Toc142917796)

[1.1 ZTNA Modes of Access 5](#_Toc142917797)

[1.2 Use Cases 5](#_Toc142917798)

[1.3 Deployment Models/Scenarios 5](#_Toc142917799)

[1.3.1 Remote Access Deployments 5](#_Toc142917800)

[1.3.2 Internal Access Deployments 6](#_Toc142917801)

[1.3.3 High Availability deployments 7](#_Toc142917802)

[1.3.4 Cluster Deployments 7](#_Toc142917803)

[2 Software Architecture 8](#_Toc142917804)

[2.1 Design Considerations 8](#_Toc142917805)

[2.1.1 High level deployment model 8](#_Toc142917806)

[2.1.2 High Level Workflow 8](#_Toc142917807)

[2.1.3 Components of clientless ZTNA on FTD 11](#_Toc142917808)

[2.1.4 ZTNA Policy 12](#_Toc142917810)

[2.1.5 Zero Trust Application Policy 13](#_Toc142917811)

[2.1.6 Integration with DUO SSO 13](#_Toc142917812)

[2.2 Publishing Applications 14](#_Toc142917813)

[2.3 DNS Resolution 15](#_Toc142917814)

[2.4 Authenticating Users 15](#_Toc142917815)

[2.4.1 TLS Termination 16](#_Toc142917816)

[2.4.2 SAML Redirection 16](#_Toc142917817)

[2.4.3 SAML Assertion Consumption 16](#_Toc142917818)

[2.4.4 Authentication Metadata Population 16](#_Toc142917819)

[2.5 Authorizing User to App Access 16](#_Toc142917820)

[2.5.1 Auth Classification Rules 16](#_Toc142917821)

[2.5.2 Auth Database Population 17](#_Toc142917822)

[2.5.3 Auth Database Validation 17](#_Toc142917823)

[2.5.4 Cookie Validation 17](#_Toc142917824)

[2.5.5 Re-authenticating Users 18](#_Toc142917825)

[2.6 Application groups 18](#_Toc142917826)

[2.7 HTTP2 protocol support 19](#_Toc142917827)

[2.8 Failover Support 19](#_Toc142917828)

[2.9 Cluster Support 19](#_Toc142917829)

[2.10 WebVPN Migration Support 20](#_Toc142917830)

[2.11 ZTNA Firewall Plugin 20](#_Toc142917831)

[2.11.1 ztna.rules 20](#_Toc142917832)

[2.11.2 zero\_trust.lua 21](#_Toc142917833)

[2.11.3 ZTNA Event 22](#_Toc142917834)

[2.12 Source NAT for clientless ZTNA 22](#_Toc142917835)

[2.13 ZTNA Diagnostics tool 24](#_Toc142917836)

[3 Software Requirements 27](#_Toc142917837)

[4 Memory and Performance Impact 28](#_Toc142917838)

[4.1 Memory 28](#_Toc142917839)

[4.2 Performance 28](#_Toc142917840)

[5 Platform Specific Considerations 29](#_Toc142917841)

[6 Packaging Considerations 29](#_Toc142917842)

[7 End User Interface/User Experience 29](#_Toc142917843)

[7.1 Configuring ZTNA feature 31](#_Toc142917844)

[7.1.1 Enabling/Disabling ZTNA 31](#_Toc142917845)

[7.1.2 Base Zero-trust URL for  assertion consumption 32](#_Toc142917846)

[7.1.3 Session timeout for zero-trust sessions 33](#_Toc142917847)

[7.2 Configuring Application Groups 34](#_Toc142917848)

[7.2.1 Configuring an application group 34](#_Toc142917849)

[7.2.2 Enable zero-trust protection for the application group 35](#_Toc142917850)

[7.2.3 IdP entity ID for application group 36](#_Toc142917851)

[7.2.4 SAML IdP sign-in URL for application group 37](#_Toc142917852)

[7.2.5 IdP Trustpoint for an application group 37](#_Toc142917853)

[7.2.6 Service Provider trustpoint for an application group 38](#_Toc142917854)

[7.2.7 Configure signature for SAML authentication request for an application group 39](#_Toc142917855)

[7.3 Configuring Applications 40](#_Toc142917856)

[7.3.1 Configuring an application 40](#_Toc142917857)

[7.3.2 Adding application to application group 41](#_Toc142917858)

[7.3.3 Enable zero-trust protection for the application 42](#_Toc142917859)

[7.3.4 Exposing application on an interface 42](#_Toc142917860)

[7.3.5 Configure IdP for SAML based authentication 43](#_Toc142917861)

[7.3.6 Configure SAML identity provider sign-in URL 44](#_Toc142917862)

[7.3.7 Configure external URL for zero-trust application 45](#_Toc142917863)

[7.3.8 Configure internal URL for zero-trust application 46](#_Toc142917864)

[7.3.9 Configure mapped port associated with the application 47](#_Toc142917865)

[7.3.10 Configure IdP Trustpoint for an application 48](#_Toc142917866)

[7.3.11 Configure Service Provider (SP) trustpoint for an application 49](#_Toc142917867)

[7.3.12 Configure signature for SAML authentication request for an application 50](#_Toc142917868)

[7.3.13 Enable/disable force-reauthentication for an application 51](#_Toc142917869)

[7.3.14 Configure Source NAT for an application 52](#_Toc142917870)

[7.4 Show Commands 52](#_Toc142917871)

[7.4.1 Show Config 53](#_Toc142917872)

[7.4.2 Show Run-time State 55](#_Toc142917873)

[7.4.3 Show Cluster-wide Run-time State 56](#_Toc142917874)

[7.4.4 Show conn ZTNA flow tagging: 57](#_Toc142917875)

[7.5 Clear Commands 58](#_Toc142917876)

[7.5.1 Clear Config 58](#_Toc142917877)

[7.5.2 Clear Run-time State 59](#_Toc142917878)

[7.5.3 Clear Cluster-wide Run-time State 60](#_Toc142917879)

[7.5.4 Show Snort Statistics 61](#_Toc142917880)

[7.5.5 Show Snort Counters 62](#_Toc142917881)

[7.5.6 Show Counters 63](#_Toc142917882)

[7.6 Debug Commands 63](#_Toc142917883)

[7.6.1 Connection Based Debugs 64](#_Toc142917884)

[7.6.2 Non-connection/Config Based Debugs 65](#_Toc142917885)

[7.6.3 Debug Menu Commands 65](#_Toc142917886)

[7.7 Syslog Messages 68](#_Toc142917887)

[7.8 Events 68](#_Toc142917888)

[7.8.1 Connection Events 68](#_Toc142917889)

[7.8.2 Dashboards: 68](#_Toc142917890)

[7.9 Configuring through CLISH 69](#_Toc142917891)

[7.10 Configuring through FMC 69](#_Toc142917892)

[7.10.1 ZTNA Policy 69](#_Toc142917893)

[7.10.2 HW Modeling 72](#_Toc142917894)

[7.10.3 DUO API Integration 73](#_Toc142917895)

[7.11 FMC Deployment 73](#_Toc142917896)

[7.12 Configuring through ASDM/FDM 74](#_Toc142917897)

[7.13 Rest API Support 74](#_Toc142917898)

[8 Caveats 74](#_Toc142917899)

[9 Testing Considerations 75](#_Toc142917900)

[10 Troubleshooting 75](#_Toc142917901)

[10.1 Troubleshooting on FTD 75](#_Toc142917902)

[10.2 ZTNA Diagnostics 75](#_Toc142917903)

[10.3 SRC NAT troubleshooting 77](#_Toc142917904)

[11 Telemetry 78](#_Toc142917905)

[11.1 FTD Telemetry 78](#_Toc142917906)

[11.1.1 LINA Telemetry 78](#_Toc142917907)

[11.1.2 Snort Telemetry 79](#_Toc142917908)

[12 Initiative, Legal, & Regulatory 81](#_Toc142917909)

[13 Requirements Traceability Considerations 81](#_Toc142917910)

[14 SReferences 82](#_Toc142917911)

[15 Glossary 82](#_Toc142917912)

[16 Attachments 82](#_Toc142917913)

[16.1 Review Action Items 82](#_Toc142917914)

# 1 Problem Definition

ZTNA is a modern way of enabling remote and on-prem users to securely access internal resources and applications. It hides the network from users and allows users to only access applications they are authorized for without assuming that authorization for one application in the network gives an implicit authorization for other applications on the network, thereby reducing the attack surface significantly. In other words, every access to an application has to be explicitly authorized (though explicit authentication may not be essential for every access).

With firewalls, traditional access control mechanisms have relied on network elements and rules typically allowed or denied access based on source and destination of the traffic. However, with ZTNA the access model changes to per app access rather than network based access. Given the install base of Cisco’s FTD devices, adding ZTNA functionality to the FTD which provides a per app access would allow for easy adoption and migration to a more secure access model without having to install/manage yet another device in the network. The ZTNA model described here leverages our firewalls as enforcement points (gatekeepers to app access).

Since a firewall already does access control and inspection, adding a directed access to the application for the user is logical extension to cater to the ZTNA principles.

The added advantage of leveraging the data path is that the entire resources of the data path are available for providing the access and so the performance would be much better with negligible costs overhead.

Some of our competitors like Fortinet have already adopted this model successfully to create user↔application micro segmentation using object based firewall rules.

## ZTNA Modes of Access

There are 2 major access modes considered under ZTNA

1. Clientless access:

Here the access to the protected resource is done over a web browser. This mode can support only HTTP/HTTPS content and no client/agent is installed on the endpoint for secure access. As there is no client, non-HTTP(s) applications cannot be supported. Also this mode requires access to the DNS entries of the protected services which implies that the internal services are published on the public DNS servers.

1. Client based / Agent based model:

In this model, an agent or client is needed on the endpoint for primarily 3 purposes:

1. Device posture: The posture of the device is evaluated and based on policy access to an application is granted.
2. Support for non-HTTP(s) application traffic.
3. Prevent exposure of internal resources on the public DNS.

Only the clientless mode of ZTNA would be covered as part of this initiative for FTD 7.4 release and only handle HTTPs traffic. Supporting HTTP traffic requires active termination of HTTPs traffic on the client facing side and creating a HTTP traffic on the server facing side. This capability requires enhancements to the proxy stack and will be taken up in subsequent releases.

## Use Cases

* HTTPs application support
  + Authentication of Web Applications, which are accessed using browser
  + Verifying if an incoming connection from user is authenticated. Redirect user to authenticate if needed.
* Multiple Application Support
* Multiple IdP Support
* Support Application groups with single IdP authentication

## Deployment Models/Scenarios

Customers deploy FTD devices both at the edge as well as at internal network boundaries. The FTD masquerades as the application and intercepts the connection until the user has been authenticated, so the FTD should be deployed only in routed mode. The possible scenarios of deployment are discussed below.

### Remote Access Deployments

For access enabling access to internal applications to remote users, the application server should be at the DMZ. The IP of the external interface of the FTD should be published as the application server’s IP on the public DNS. This can be done by adding a CNAME record pointing to the FTD’s external interface IP. Each FTD could be protecting multiple applications and there may be multiple sites protecting multiple applications and customers would like to access all allowed applications at the same time without having to rely on site-to-site connectivity between the sites hosting the applications.

Figure 1: Remote Access Deployment

Diagram

Description automatically generated

### Internal Access Deployments

A few major customers of FTD also require users on premise to access applications using ZTNA. In such scenarios, multiple on premise FTDs could be protecting multiple applications and all such protected applications should be accessible simultaneously. When any DNS query is made for the application server, the FTD will intercept the DNS queries and rewrite the response with the corresponding FTD IP as the application server’s IP.

Figure 2: Internal Access Deployment

Diagram

Description automatically generated

### High Availability deployments

FTD devices protecting applications can be hosted in High Availability mode. In such cases the DNS entries of the applications protected by the HA setup should point to the Active IP of the HA setup.

### Cluster Deployments

FTD devices protecting applications can also be hosted in a cluster setup. In such cases the DNS entries of the applications protected by should point to the Virtual IP of the cluster. In a cluster setup, the node that services the initial AuthN request for a connection for SAML initiation may not be the node that receives the SAML response. So appropriate state sharing would be done within the nodes in the cluster for proper authentication. This is discussed in more detail in later sections.

# Software Architecture

## Design Considerations

### High level deployment model

Using a browser, a user attempts to connect to an application. This connection is intercepted by the firewall that protects the application. Firewall redirects the user to application’s configured IdP for authentication and once authenticated, the firewall allows user to access the application. This model does not require a site-to-site VPN. Each firewall protects a set of web applications and user can directly authenticate and access apps behind respective firewall.

Figure 3: FTD deployment for Clientless ZTNA

Diagram

Description automatically generated

.

### High Level Workflow

At a high level the workflow consists of the following steps

1. Configuration
   1. Port Assignment
   2. DNS Publishing
2. Pre-Authentication
3. Authentication
4. Post-Authentication and Application Access

A brief overview of the above is given in this section along with the sequence diagram. More granular details are listed in the latter sections of this document.

#### Configuration

Each application (in essence an URL) that needs to be exposed as a ZTNA application, needs to be assigned a port on the firewall which will used to allow the browser to reach the application. This application also needs to be published on the DNS with the destination IP of the FTD protecting it. Essentially, the FTD masquerades as the application so that when the browser attempts to reach the application via its FQDN, the request comes to the FTD protecting that application.

**Note**: For the FTD to masquerade for all the applications that it protects, the TLS certificate presented during the initial connection should either be a wildcard domain certificate or the Subject Alternate Name in the certificate should have the list of applications.

#### Pre-Authentication

An Authentication Proxy service on the FTD listening on port 443 intercepts the flow based on the Host header in the HTTP traffic and determines if the application is one of its protected applications. If it is, a SAML Request Assertion is created for the application’s Identity Provider (IdP) and the traffic is redirected for Authentication.

#### Authentication

The IdP (or the IdP Broker) is the place where the user and application and posture policy are associated. In this step the IdP authenticates the user for the application and depending on policy may do a two-factor and posture assessment before generating an SAML Response Assertion for the FTD.

#### Post-Authentication and Application Access

The post authentication is a two-part flow. In the first part, the SAML Response generated by the IdP is validated and cookie is generated for the Application Host domain. The FTD then issues a HTTP redirect for the application to access the service on the port assigned 2.1.2.1

In the second part, when the traffic from the browser hits the application specific port on the FTD, the IP of the client (4-tuple) is validated in Lina and the cookie is validated in Snort. If either of these check fails, the traffic is redirected to the port 443 where the Authentication Proxy will start the authentication process. If the checks pass, then the traffic is NATed to the application and traffic inspected as per the application’s policy.

Figure 4: End-to-End Workflow

Diagram, schematic

Description automatically generated

### Components of clientless ZTNA on FTD

Figure 5: Components Diagram

Diagram

Description automatically generated

#### FTD

* FTD is an enforcer, which will actually protect applications hosted behind it.
* It acts as authentication gateway and redirects users to authenticate with configured IdP.
* On successful auth, it allows the user to access application and inspects the traffic according to policy configured.

#### SAML Auth Service

WebVPN already supports SAML authentication and the same is being used by AnyConnect VPN for authentication. This service is part of Lina and handles the user authentication in the control plane.

* It listens on To the Box traffic on 443 and terminates TLS connections from user
* It redirects user to IdP for SAML authentication and verifies SAML assertion coming from IdP
* After successful SAML authentication, it updates the Auth DB and redirects the user to service configured on a mapped port (PAT).

#### DUO / IdP

DUO or any other IdP is used as a MFA and policy engine. Duo will provide configuration such as which user can access which application.  It also provides multi factor authentication service by integrating with IdP such as Microsoft.

Additionally, Duo provides the client device posture, which helps to assess client's health.

Customers who already have a different IdP and MFA (such as Okta), can continue to use same for authentication.

#### Snort

After successful authentication, a cookie is generated in stored in Auth DB. Same cookie is shared with user browser on HTTP redirect to mapped port. Snort gets a copy of cookie from Auth DB and verifies cookie user is sending on every HTTP request. If cookie validation fails, Snort drops the connection and redirects user to authentication again. Only Snort 3 will be enhanced for this validation and devices running Snort 2 will not be able to support this feature.

#### Auth checks and Auth DB

Auth DB resides in Lina and keeps track of authenticated clients (for each app). After successful client authentication, Auth DB will be updated with client IP and app it has authenticated for. When client connects to app, connection will be validated against this DB. If Auth check fails, client will be redirected to authenticate to the app. Auth DB also contains information like Cookies and Redirect-URL if access fails.

This Auth check in Lina will serve as first level of verification, which prevents un-authenticated clients from accessing the app. Second level Cookie verification will be done by Snort.

#### Policy and configuration

FMC configures applications protected, with details such as internal FQDN for application and IdP to be used for authentication.

User access and posture polices will be outside FMC. This basically defines which user/group has access to which application, MFA policies, posture requirements of client device etc. If Duo is used for authentication, then these policies are configured and managed at Duo policy engine.

### ZTNA Policy

A ZTNA policy typically has 3 components

1. User and Groups definition
2. User’s device posture policy
3. Application definition

The policy intent can be expressed as user or group of users having Discretionary Access to an application wherein the access is based on the posture of user’s device being used to access the application.

In the FTD, identity of the users are captured either by mechanisms like Captive Portal or Cisco’s ISE. In either of these approaches the identity is tied to the IP of user’s device. When a user switches networks or the IP of the device changes, there is a period of interval, though brief, where the authorized IP could be reused by another user.

Since the FTD protects the network where the private applications exist, the FTD/FMC is the logical place to define Applications that need to be exposed for Zero Trust Access.

Existing AC policies are more network centric and hence are not directly applied on ZTNA traffic. ZTNA being more application centric, a new ZTNA policy will be created, where the policies like Threat and File can be configured to inspect the application traffic. The traffic intended for ZT application will be evaluated against the ZT rules and not fall through the regular AC rules.

In the policy envisaged for the FTD in the clientless model, the Users & Groups definition, the device’s posture and the association of the users/groups with the applications are to be defined and evaluated outside the FTD. Only the Zero Trust Application & it’s threat policy would be defined in the FMC.

### Zero Trust Application Policy

The Zero Trust Application Policy is the method for assigning threat policies to application definitions. An administrator will have the ability to assign an existing (or create a new) Inpection and File/Malware Policy to a ZTNA enabled application.

This policy is different from the traditional Access Control Policy typically defined on the firewall.

Typical AC policies have source and destination network ranges among other criteria which are evaluated for the application of the policy. In the case of ZTNA, the policy is more application specific where the administrator decides the inspection levels based on the threat perception for that application. It has been observed that,over a period of years, AC policies bloat with a lot of rules as networks and organisations evolve and some of these rules become redundant when applications are decommised or moved but the rules persist since the effect of removal is not fully understood or considered risky. Modification of the rules also has to be done carefully since a change would impact all applications hosted in that network and applying a policy for a specific set of applications using such constructs is cumbersome.

ZTNA policies are supposed to be more manageable as they are application focussed and the network elements do not have a strong role in this model. Irrespective of which network segment hosts the application, the policy is applied for the application and even if the application were to be moved around network segments, the policy applied to the application is always the same. When an application is decommissioned, the application entry can be removed without impacting other applications.

### Integration with DUO SSO

Note: This is a stretch goal for 7.4 release and not part of MVP.

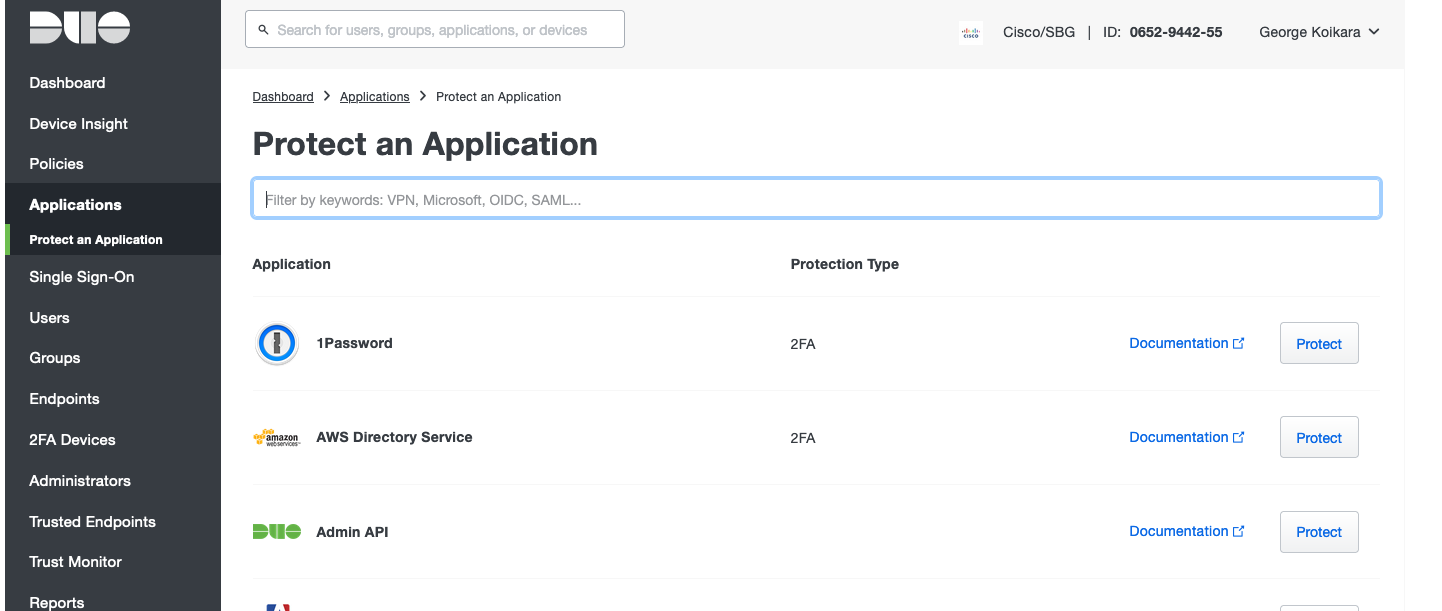
DUO provides a full Zero Trust (ZT) policy in terms of associating the users, applications and device posture. One of the stretch goals for 7.4 release would be to allow FTD customers to configure the ZT policy from the FMC. This would remove the need to switch browsers/tabs for accessing different management consoles to create a full ZT policy especially for the SAML configuration which require exchange of SAML metadata between the Service Provider (FTD), and the IdP (DUO).

The objective here would be to publish the applications defined on the FMC to the DUO policy page.

Some of the activities that would be done as part of this workflow would be the following:

* Onboarding of FMC admin account as DUO tenant
* Associating or adding a DUO API key to the FMC as a configuration element
* Push the application definitions from FMC to DUO using APIs so that private applications are listed in the ‘Protect an Application’ page in DUO screen below or any other appropriate screen as needed so that the admin can associate users and device policies with this private application on DUO to build the Zero Trust Policy.

Figure 6: DUO screen to add application



## Publishing Applications

Each application definition will have these key configuration attributes:

* Interface: FTD interface on which the application is configured. There can be multiple apps configured on same interface. The external URL used by clients, should resolve to the IP configured in the interface.
  + Same application can be configured on multiple interfaces
* External URL: End users/clients specify this in their browser address bar to access the application.
  + Example: https://wiki.acme.com
* Internal URL: The URL for application in the inside network. This URL will be resolved to server IP and Port and used along with the mapped port to configure a dynamic NAT rule. The rule basically translates traffic coming on mapped port to server IP and port.
  + Example: https://wiki.acme.com:8080
* Mapped Port: Port which identifies the application. This will be used to NAT (PAT) the traffic coming from user to internal server’s IP where application is running. This must be outside range of standard ports.
  + Example: 9000
  + Instead of asking admin to define a port, FMC can automatically pick an unused port and assign while admin is configuring the application. The range of ports to be used for this purpose can be configured by admin (example: 9000-9100).
* IdP: This is IdP which authenticates the user for accessing the application.
  + Example: <http://www.okta.com/exk5460t24OenZZcI5d7>

## DNS Resolution

In scenarios where protected applications are exposed to public domains, the public DNS servers should be configured to map application’s external URLs to the FTD devices public interface IP behind which those applications are hosted.

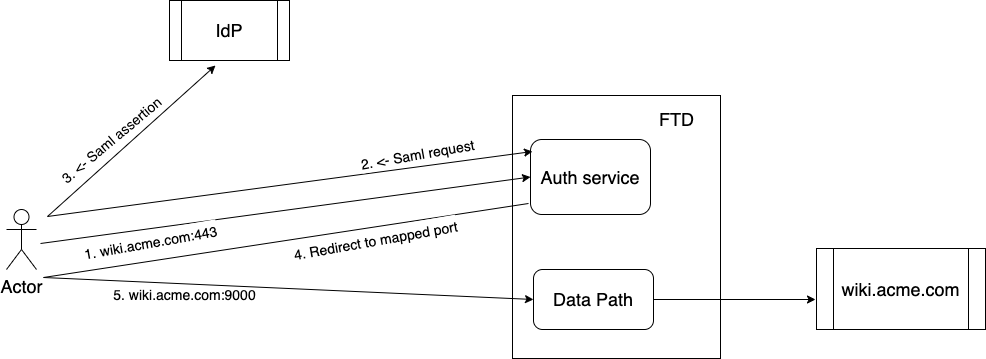
In scenarios where protected applications are exposed to internal domains, the internal DNS servers should be configured to map application’s external URLs to the FTD devices internal interface IP through which the users will access protected applications.

FTD’s DNS server configuration will be used to resolve the application’s internal URL.

## Authenticating Users

Existing WebVPN authentication framework will be used for SAML based authentication of users. This is currently used by AnyConnect VPN for SAML authentication.

Figure 7: Authentication Sequence



* Today, SAML service (WebVPN) performs redirects to logon.html asking for selection of a tunnel group. Tunnel groups have no significance in ZTNA. Hence, tunnel groups should be removed or completely hidden from user. A default ZTNA tunnel group can be defined and used for all applications.

### TLS Termination

ZTNA will be configured on specific interface on FTD. The application hostnames (ex: wiki.acme.com or bitbucket.acme.com) will resolve to this FTD IP. SAML auth service in lina will listen on 443 port on this IP and front-end any access to applications. Auth service will terminate the TLS and parses the HTTP headers to get the hostname (application that user is trying to access).

### SAML Redirection

Auth service fetches the hostname of the application from HTTP headers and looks up in the configured ZTNA applications. It fetches the IdP details from the configuration and constructs a SAML authentication request. This request will be added as part of the “Location” in HTTP response and sent as redirection message to user. User will land on the IdP login page and provide the credentials. Once IdP successfully authenticates the user, it redirects user back to FTD with SAML assertion.

* If SAML auth service finds no ZTNA application with hostname, it drops the connection and logs an event

### SAML Assertion Consumption

SAML assertion will be verified by the SAML auth service. On success, it updates the AuthDB with IP, User, Cookie details for the connection. Then user will be redirected to configured mapped port to access the application. During HTTP redirect, the generated cookie will be set on the domain being accessed.

On successful assertion verification, corresponding counters are incremented

If SAML assertion verification fails, then the connection is dropped, and an event is logged with reason.

### Authentication Metadata Population

Once SAML assertion is verified successfully, a cookie will be generated and client is redirected to mapped-port with cookie set on the domain. Cookie will be a hash of expiry time, user email and random bytes. Total size of cookie should not exceed 32 bytes.

## Authorizing User to App Access

Post authentication, each application access needs to be authorized by FTD. This section describes how that authorization happens.

### Auth Classification Rules

On configuration of an application profile, a classification rule is installed in datapath to match all traffic destined to that application. For example, let us say a user configured an application profile for “confluence” exposing it on port 9000 and interface “outside”. The classification rule will be installed to match any traffic coming from any source and destined to “outside” interface IP and port 9000.

The auth classification rules can be viewed through “show asp table classify domain zero-trust” CLI and a typical classification rule would look like:

Input Table

in id=0x15196d876970, priority=13, domain=zero-trust, deny=false

hits=0, user\_data=0x0, cs\_id=0x0, reverse, flags=0x0, protocol=6

src ip/id=0.0.0.0, mask=0.0.0.0, port=0, tag=any

dst ip/id=172.16.77.0, mask=255.255.255.0, port=9000, tag=any

src nsg\_id=none, dst nsg\_id=none

dscp=0x0, input\_ifc=Outside(vrfid:0), output\_ifc=any

### Auth Database Population

As part of successful authentication of each user to application request, the SAML assertion will be consumed and validated by SAML auth service. On successful assertion consumption, the metadata will be populated as part of which the authentication database is populated with details such as Client IP, User, Cookie, Application. The validity of this metadata is timebound and removed from authentication database on expiration.

### Auth Database Validation

The first packet of each new connection will be subjected to datapath classification rules. When the traffic matches ZTNA authorization classification rule, we lookup for a matching entry in authentication database. If we have a matching entry, then we allow the connection to progress further. If we do not find a matching entry, we will intercept the connection and redirect client to the application’s external URL for re-authentication.

### Cookie Validation

The validation of connections against authentication database for authorization is a first step towards ensuring zero trust. But that alone may not be sufficient in scenarios where there are subsequent connections from the same authenticated user. In such cases, we need to authorize requests through cookie validation.

For each HTTP(s) request from a user to ZTNA protected application, we need to extract the cookie from request headers and validate it against the metadata populated for that user<->app session.

The cookie validation is handled in Snort and only Snort 3 inspectors are being modified to handle this.

Thus, this feature can be used only if Snort 3 is enabled on the FTD device.

#### Cookie Propagation from Lina to Snort

Lina will perform SAML assertion validation and populate the metadata. Lina will also populate cookies that are shared with client so that client and produce them in the actual application requests. These cookies will be made available to Snort so that it can validate the cookies in HTTP(s) requests as part of connection authorization.

As soon as the first packet of a new connection is validated against ZTNA auth classification rule and sent to Snort, lina will extract the related cookies and application URL of the respective session from authentication database, pack them into TLV formatted structure and send it across to Snort via PDTS. Lina will also set a ZTNA flow flag so that Snort can identify such flows.

#### Snort Cookie Validation

DAQ on the Snort side will receive and process the message containing cookies and application URL. It will unpack the TLVs into a local data structure and store it as part of daq headers.

As part of HTTP(s) request processing, once the cookie header is processed and cookie values are extracted, Snort will invoke a DAQ IOCTL API to fetch the lina shared cookies. Snort will match the cookie in request header with locally stored cookie values. The packet as well as connection will be allowed if there is a match. If not, the packet will be dropped, and a 307 redirect message is built with location set to application URL and injected towards client and the connection will be terminated.

### Re-authenticating Users

Users are re-authenticated against application access requests whenever there are authorization failures. This authorization happens locally to FTD and is done on two levels.

#### Auth DB Validation Failures

Whenever FTD receives a connection destined to protected application without a matching authentication database entry, the connection will be intercepted, and the client will be redirected to application’s external URL for re-authentication.

This invalidation can also happen due to expiry of the authentication database entry or forceful removal of an entry due to posture invalidation.

#### Snort Cookie Validation Failures

Whenever FTD receives a connection destined to protected application with a matching authentication database entry but does not contain a valid cookie in the HTTP(s) request header, the client will be redirected to applications external URL for re-authentication and the connection will get terminated.

#### Applying ZT Application Policy

Every ZTNA flow would be subject to the ZT Application Policy. In the post authentication phase when the traffic hits the application specific port, LINA would notify Snort and SSL modules using a ‘ZTNA flow’ flag. The SSL engine will use this flag to compare the SSL rules specific to the ZTNA applications. On the Snort side, this ZTNA flag will be used by the firewall module to associate the right threat policy to the flow.

In case the administrator has chosen not to associate a threat policy with an application, the firewall module, after cookie validation, will whitelist the flow to bypass snort inspection.

## Application groups

Need for supporting application groups:

* Each application needs configuration on IdP. And IdP details for each application must be configured on FMC. It becomes difficult to manage it in both places.
* Redirecting end user to IdP for each application access is not good experience. It also increases load on IdP, which might have cap on number of authentications per second.

With application groups, same IdP configuration can be used for multiple applications. Application which is part of a group will inherit the group’s SAML configuration.

When end user tries to access an application which is part of group, user will be authenticated to the group the first time. Later, when user tries to access other apps in same group, will be allowed without redirecting to IdP.

## HTTP2 protocol support

Currently, webVPN authentication framework only supports HTTP/1.1 version. However, Application traffic is expected to be in HTTP/2 as well. During the negotiation phase of HTTP, HTTP/1.1 is used before upgrading the connection to HTTP/2. This allows using the existing WebVPN HTTP/1.1 framework for authentication before switching to the application port for data access which an still happen in HTTP/2. Providing HTTP/2 native support in the authentication phase is not warranted at the moment.

## Failover Support

All pre-auth connections destined to device IP and service port (eg: FTD outside IP and 443) for authentication are not replicated to standby. If a failover role change happens before the authentication is complete, the new active unit will drop packets belonging to ongoing sessions and respond with reset. Client will have to re-initiate the connections for authentication to progress further.

On successful authentication of a user request towards a protected application and as part of SAML assertion consumption, we populate an authentication database entry to validate and authorize the connections. This database entries will be replicated to standby. We will respond to client with metadata only after replicating the entry to standby.

If a failover role change happens in the middle of assertion validation, the new active will reject any packets belonging to old connection as it will not have any flow state. In such cases, the client may re-initiate connection and retry sending SAML assertion.

Post authentication, irrespective of where the authenticated session lands, we will validate it against the local authentication database and allow it to progress further only if there is a matching entry and the cookie is validated successfully.

Post authentication connections are handled as regular through-the-box connections. Continuation of such connections across failover role changes are subject to existing support & limitations. For example, a TLS decrypted HTTPS connection will fail on role changes as we do not replicate the state of the connection across units.

## Cluster Support

All pre-auth connections destined to device IP and service port for authentication will be handled by the control node. This is because we need to handle both the SAML AuthN request and SAML ACS on the same node. If the control node fails before the authentication is complete, the new control node will drop packets belonging to ongoing sessions and respond with reset. Client will have to re-initiate the connections for authentication to progress further.

On successful authentication of a user request towards a protected application and as part of SAML assertion consumption, we populate an authentication database entry to validate and authorize the connections. This database entries will be replicated across all nodes of cluster. We will respond to the client with metadata only after replicating the entry across cluster.

Post authentication, irrespective of where the authenticated session lands, we will validate it against the local authentication database and allow it to progress further only if there is a matching entry and the cookie is validated successfully.

Post authentication connections are handled as regular through-the-box connections. Continuation of such connections across cluster node failures are subject to existing support & limitations.

In the case of L3 cluster (which we started supporting in FTDv), we need to ensure that the external URLs resolve to the device IP of control node.

## WebVPN Migration Support

<is this section needed?>

## ZTNA Firewall Plugin

As every ZTNA flow is subject to ZT Application Policy, the Snort3 ZTNA firewall plugin validates if the flow belongs to the application as listed in the ZT Application Policy. Once the ZT Application Policy is configured and deployed snort creates a ztna.rules file with zero\_trust.lua containing the path to the corresponding file.

### ztna.rules

The ztna.rules file contains the ZT Application Policy details and the list of the applications part of the ZT Application Policy along with the application specific configuration of application name, url, file policy and ips policy.

**ztna.rules Format:**

*global\_policy*

*{*

*ztna\_policy\_id <policy\_uuid>;*

*ztna\_policy\_name <policy\_name>;*

*}*

*ztna\_rules*

*{*

*<ztna\_app\_id>*

*{*

*ztna\_app\_id <id>;*

*ztna\_app\_name <app\_name>;*

*url <url>;*

*ips\_policy <policy\_id>;*

*app\_id <appId>;*

*file\_policy <policy\_id>;*

*}*

*}*

**Example:**

*global\_policy*

*{*

*ztna\_policy\_id 9999;*

*ztna\_policy\_name ztna\_testing;*

*}*

*ztna\_rules*

*{*

*12345*

*{*

*ztna\_app\_id 12345;*

*ztna\_app\_name wiki.abhicorps.com;*

*url* [*https://172.16.77.1*](https://172.16.77.1/)

*ips\_policy 1;*

*app\_id 1122;*

*}*

*268435459*

*{*

*ztna\_app\_id 268435459;*

*ztna\_app\_name extapp.ssummang.com;*

*app\_id 676;*

*url* [*https://extapp.ssummang.com*](https://extapp.ssummang.com/)

*file\_policy 1;*

*ips\_policy 2;*

*}*

*268435460*

*{*

*ztna\_app\_id 268435460;*

*ztna\_app\_name extapp.ssummang.com;*

*app\_id 676;*

*url* [*https://extapp.ssummang.com*](https://extapp.ssummang.com/)

*ips\_policy 2;*

*}*

*}*

### zero\_trust.lua

The ***zero\_trust.lua*** file contains the path to the ***ztna.rules*** file.

**zero\_trust.lua Format:**

*custom\_zero\_trust =*

*{*

*rule\_path = '/var/sf/detection\_engines/<uuid>/ztna.rules'*

*}*

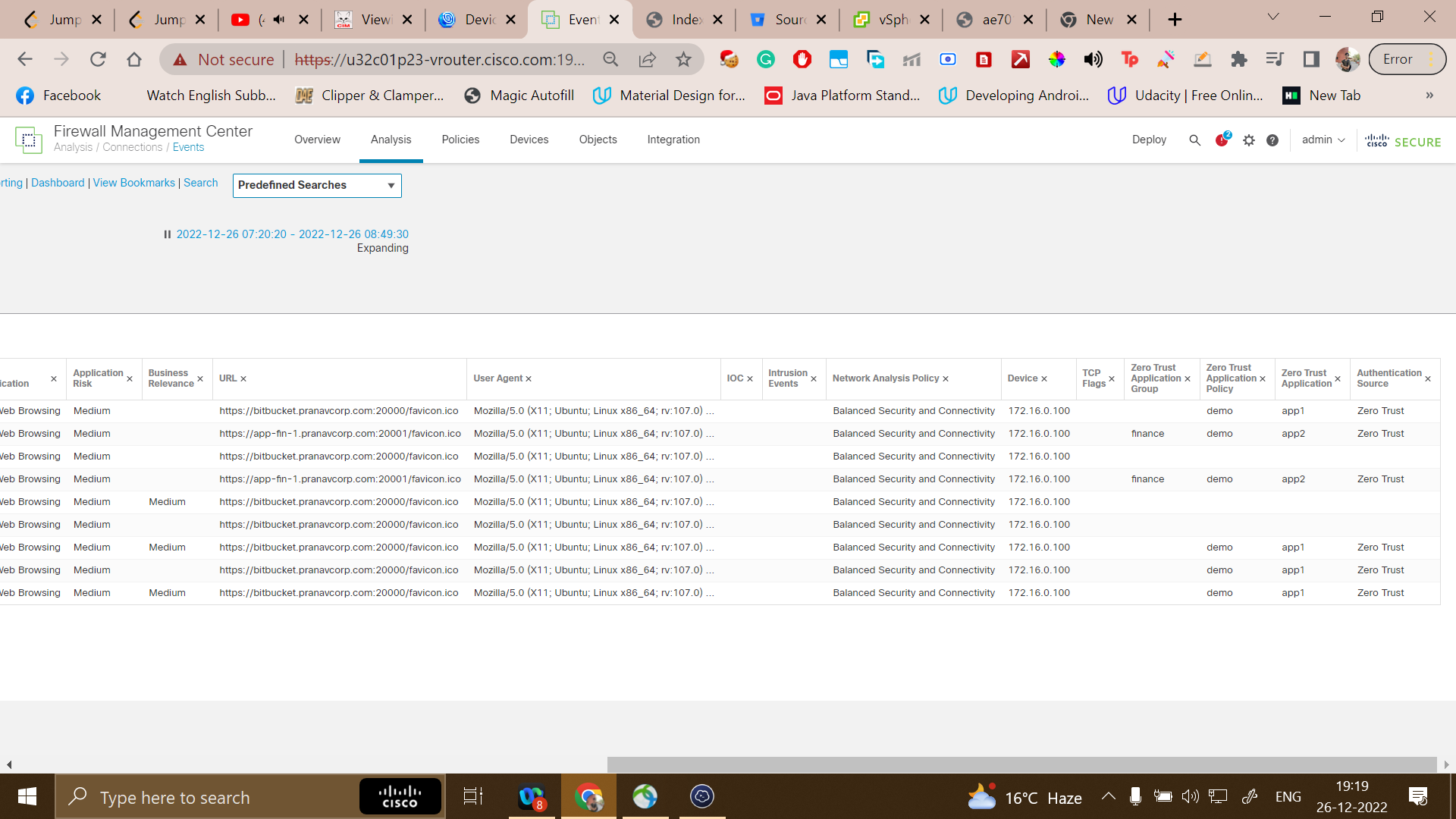
*zero\_trust  = util.merge(zero\_trust, custom\_zero\_trust)*

Post policy deployment, snort reads the ztna.rules file path from zero\_trust.lua, it then loads ztna.rules and stores all the application configuration. For every ZT flow received by snort is matched against the ztna application id (ztna\_app\_id) from the list of applications added in the ZT Application Policy. In case ZTNA flow destined to a protected application matches any of the applications listed in the ZT Application Policy, the ZTNA flow will be allowed access to the protected application. On the other hand, the ZTNA flows will be blocked if they do not find a match in the application list configured as part of the ZT Application Policy.

Based on the configuration of the ZT Application Policy, the ZTNA flows can further be subjected to ips policy and file policy.

### ZTNA Event

For every ZT flow which is allowed access to the protected application a ZTNA event is generated triggered by an EOF from LINA. The ZTNA event will contain the following details about the zero-trust flow app group name, ztna\_policy name, app name, authentication source, and url.



## Source NAT for clientless ZTNA

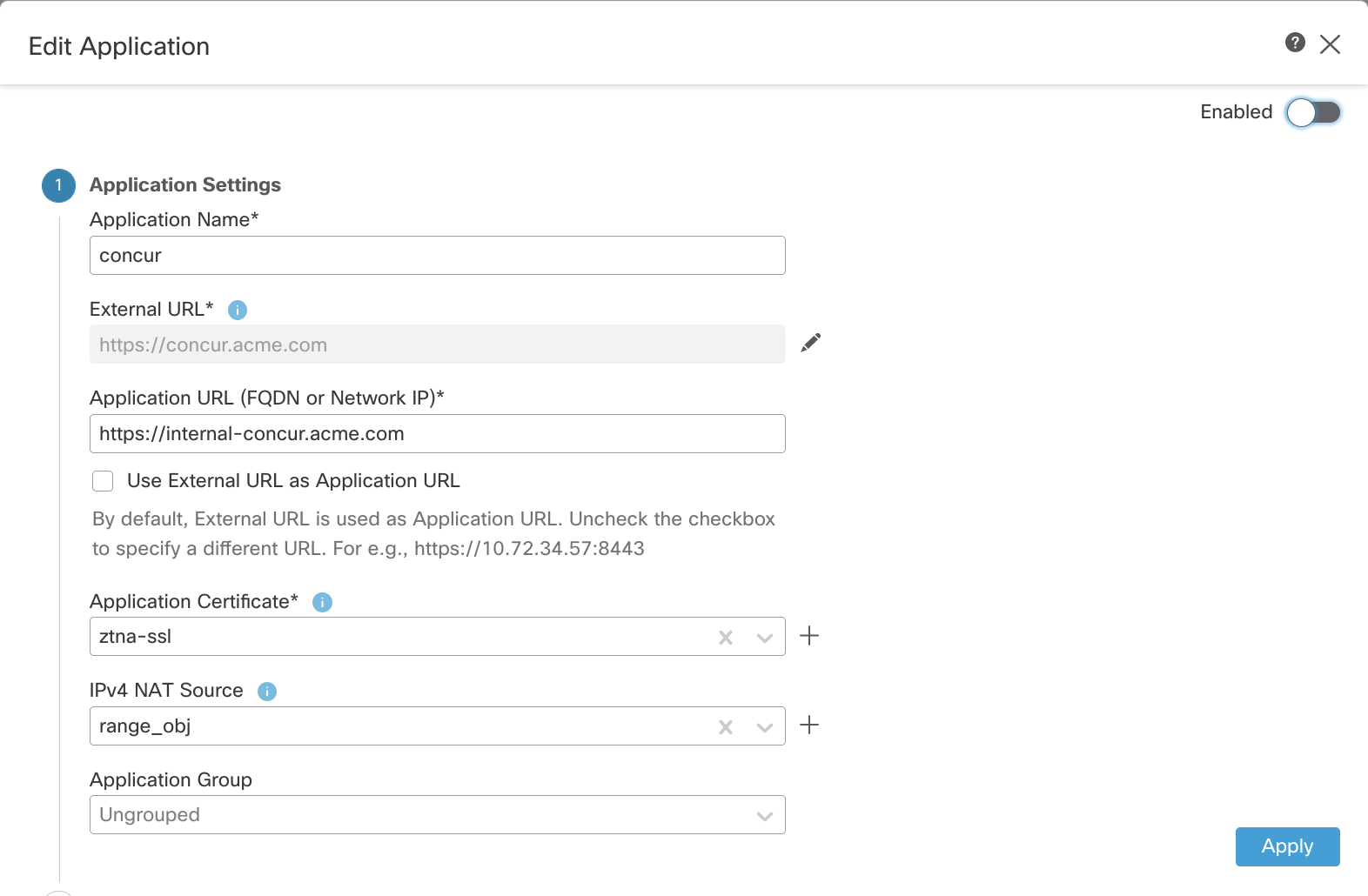
An incoming request from an end user may contain a source IP address from a public network. Inside the corporate Network, the original source IP may not be routable. to make this source IP address routable, an admin may want to translate it to a corporate network IP. Source NAT feature, introduced in the 7.4.1 release, allows an admin to configure such IP address range, which is used to translate the source IP address of incoming requests to an IP address in the Source NAT address range.

**High-Level Design:**

**2.12.1 UX/UI:**

Add "IPV4 NAT Source" configuration option in the Application configuration screen which will take a host or range-based network object/object-group as input.

Source NAT is per Application optional configuration. Below are the steps to push the configuration to FTD.



**2.12.2 FMC:**

When "IPV4 NAT Source" is configured, FMC will push this configuration using the new CLI described in the below section.

When a user tries to delete or modify the obj/obj-group to an unsupported type which is configured as mapped-source-v4 for a ZTNA application, FMC should not allow to delete/modify the obj/obj-group to unsupported type and should display an error as an object already in use/cannot be modified in ZTNA NAT rule.

**2.12.3 CLI:**

A new CLI “mapped-source-v4” is introduced to configure a mapped source for an application.

When this CLI is configured, the NAT rule for an application will be installed as shown below:

**nat (outside, any) source dynamic any pat-pool <src\_map> destination static interface <app\_real\_ip> service <real\_port> <mapped\_port>**

This NAT rule will translate any source with IP and port from <src\_map> pool when going to interface ip:<mapped\_port>.

**2.12.4 NOTES:**

* Since we use mapped interface as "any" in the NAT rule and identify the interface to reach the application server through route lookup, the admin must ensure to configure the mapped-source-v4 object to contain IPs belonging to the application server's network or the ones which are routable back to FTD from the application server.
* ZTNA application access failures due to mapped-source-v4 pool exhaustion must be notified so that corrective action (like increasing pool size) can be taken.

## 2.13 ZTNA Diagnostics tool

Clientless ZTNA involves a lot of touch points such as certificates, DNS, internal configurations such as NAT, classification rules, etc. When an issue occurs, it is tedious to troubleshoot manually. This Diagnostics feature will ease the troubleshooting process for the users by collecting CLI logs and configuration file outputs and analysing them to detect issues.

This provides an overall analysis and the logs collected can be used by engineers/admins to troubleshoot further. Currently, troubleshooting for only one application and device/node is supported. The tool analyses the current state of the machine by collecting logs, it will not be able to detect issues outside this context e.g., application reachability issues such as route not present, cannot be detected.

The troubleshooting can be broadly classified into 2 categories, application-specific and generic.

**Application-specific:**

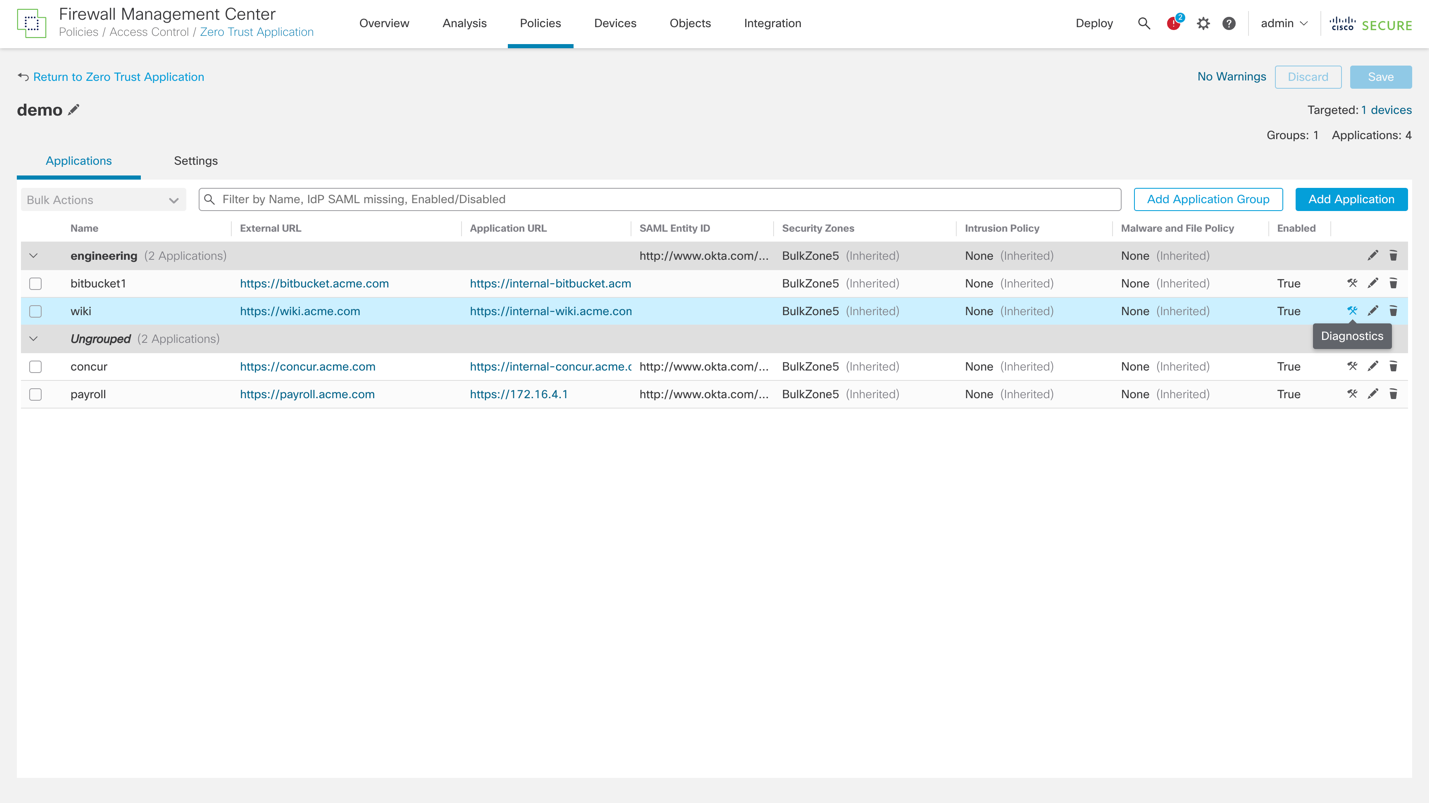
* detect DNS-related issues.
* detect misconfiguration, e.g., socket not opened, and NAT rules related issues.
* detect issues in the deployment of ZTNA and SSL rules.
* detect interface-related issues. (No IP configured or is down or is not connected)
* detect source NAT-related issues in NAT and asp drop.

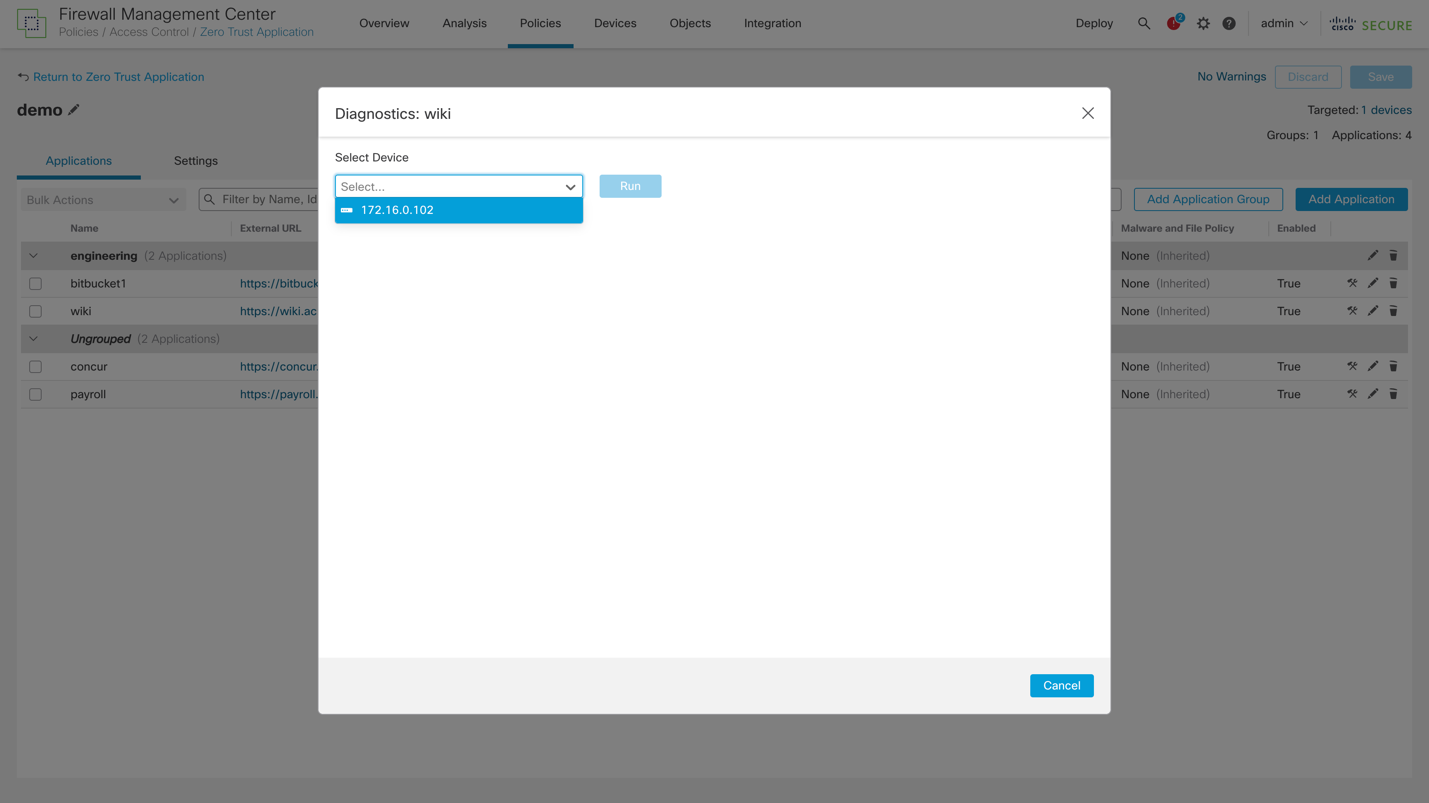
**Generic:**

* detect if a strong cipher license is not enabled.
* detect if the application certificate is not valid.
* detect if authn is not saml in tunnel group.
* detect SAML-related issues.
* detect HA and cluster bulk sync issues.
* insights from snort counters like token-related issues, decryption related, etc

**2.13.1 UX/UI:**

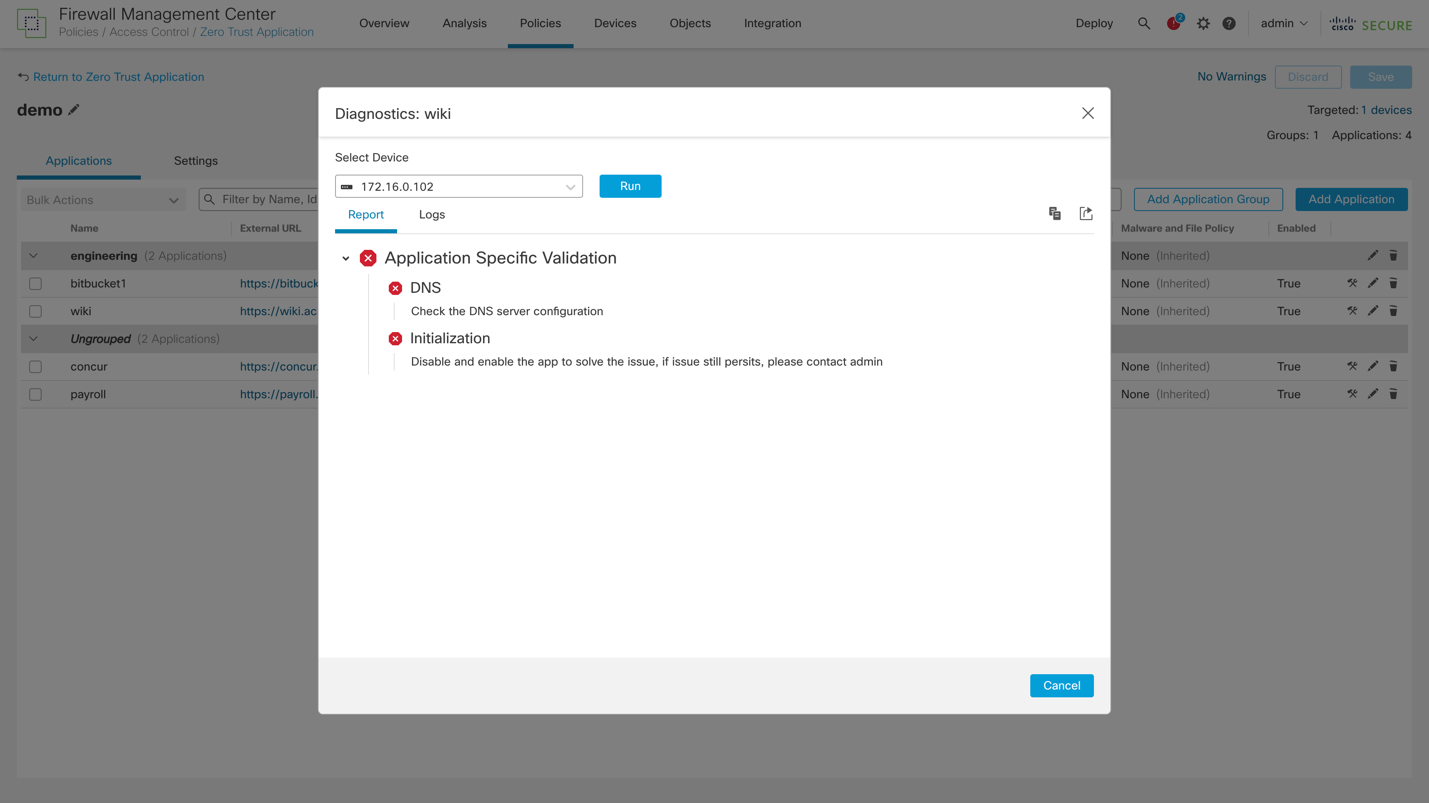
A Diagnose icon will be present inline against each app. Clicking it will open a dialog box providing a drop-down to choose the device. Once the Run button is clicked, the results are shown within this dialog box context.





A screenshot of a computer

Description automatically generated



* + 1. **CLI**

A hidden CLI is introduced to collect logs in the same format as tech-support.

show zero-trust diagnostics

The "ZT" option has been introduced in the system troubleshooting feature to collect ZTNA specific logs.   
These TS logs can be collected as part of Device Troubleshooting Logs from the FMC.

# Software Requirements

Target Process Entry: <https://targetprocess.cisco.com/entity/724028-ztna-platform>

The 7.4 release should focus on building the core workings of the ZTNA platform as well as delivering an MVP solution to customers meanwhile keeping all of the key requirements listed below in mind during the process.

1. **Core ZTNA Components -** This requirement covers the underlying infrastructure required to handle the ZTNA traffic including but not limited to:

* Intercepting of traffic destined for an application or service that is protected by the ZTNA Policy.
* Authentication redirection for an unauthenticated or "authentication expired" flow.
* Dynamically applying the relevant configuration in FTD to allow through an authenticated flow.
* Passing of traffic to Snort for handling of AVC, IPS, Malware and EVE processing where applicable.
* Status, Monitoring and failure handling for Fault Tolerance in the event of a failure.
* Detailed logging capabilities for troubleshooting.

2. **Telemetry for ZTNA -** In order to further develop the solution, we need to have telemetry showing the usage of the solution. This should include the ability to track the number of customers with ZTNA configured, how many applications they have configured, how many concurrent users have been authenticated and what platform the customer is using for their configuration.

3. **Configurable ZTNA Policy -** The Policy engine will provide administrators of the solution to configure which applications should be protected by ZTNA, which Identity source should be used (Azure AD SAML as a priority), Reauth timers, level of inspection desired etc.

4. **ZTNA for Web-Based Apps -** Should be able to protect HTTPS browser-based apps using SAML as an Authentication Source with Azure AD as an Identity Source as a minimum with Cisco Duo second (Duo does not handle primary Auth, the Primary Auth may need to be limited to AD or Azure AD for the initial use-case).

5. **ZTNA for Non-Web Apps -** Likely a stretch goal for 7.4 but this should be the next priority, in order to really address ZTNA use-cases we need to be able to protect all apps on all ports/protocols. Whilst some will require further investigation, the next focus after HTTPS should be SSH and RDP ideally using the Native OS Client Applications.

At the delivery of the 7.4 release, as a minimum customers should be able to configure ZTNA policy to protect Web-based applications for outside-in and inside-in traffic flows and authenticate users based on Azure AD with configurable policy to determine who has access.

# Memory and Performance Impact

## Memory

This feature will be allocating memory for following:

* New zero-trust apps and groups configuration:
* This will be negligible. Number of applications supported by an individual FTD is limited (perhaps in hundreds max).
* Auth table in data path
* This is a table in data path. For each client to app authentication, there will be an entry in this DB. Entry contains user email, app, cookie and other details. Estimated size of each entry will be maximum 1K bytes.

Each ZTNA application requires an SSL certificate and key pair to be provided so that the FTD can act as a MITM to inspect the traffic and validate the cookies. The memory consumption is directly proportional to the number of applications and their certificates. If all applications use the same certificate and key (wildcard certificates) then the memory used would be far lesser as compared to all applications using separate certificate-key pairs.

This memory consumption puts limits on the number of applications that can be configured on a single FTD device and is platform dependent. The recommended number of applications that can be serviced by a platform would be known after performing tests midway into the development and will be updated later.

## Performance

We will have to consider following performance aspects:

* Number of authentications per second by control plane (CP)
* CP terminates initial connection and redirects to IdP. It processes SAML assertion and redirects to mapped port.
* There will be a limit on how many simultaneous authentications CP can support.
* Total number of applications/groups that can be supported
* Total number of connections that can be supported
* Each data path connection will need to be decrypted for Cookie inspection. This depeneds on TLS performance (certificate lookups and decryption).

We will need to determine maximum numbers for each platform and accordingly set the limits.

The FTD clientless approach for ZTNA requires decryption of the flow for validation of the cookie. So for this purpose every application added to ZTNA, a new SSL rule needs to be added to an existing SSL policy or a new SSL policy is created if not already present. For scenarios where the application traffic earlier was subjected to SSL policy and inspection, there would not be any performance impact for that application.

However, for traffic that was not subjected to SSL policy and inspection earlier, such traffic would now be put through the SSL interception and IPS inspection for cookie validation which is likely to bring down the performance of the flow to that of SSL processing which would be typically 70% to 80 % degrade when compared to when no SSL policy is applied.

# Platform Specific Considerations

1. This feature is supported only on FTD platforms (not on ASA). Reason for this is we need cookie validations, which is part of Snort.
2. This feature is only supported with Snort3 deployments (not supported in Snort2).

# Packaging Considerations

There are no specific packaging considerations. The feature implementation that is part of Lina will be available only on FTD platforms.

# End User Interface/User Experience

This section describes configuration commands, show commands, debug commands, clear commands, syslog messages (if any) and events that are introduced to configure and debug clientless ZTNA functionality on FTD platforms.

Here is a snapshot of CLI configuration on FTD. We define a new CLI hierarchy under “zero-trust” and all the application/IdP/SP configuration can be added here.

Sample configuration:

zero-trust

base-url <https://ztna.acme.com>

authentication-timeout <mins>

port-range 20000-22000

log enable

enable

application-group finance

application-group-id 268434452

idp-entity-id http://www.okta.com/exk5460t24OenZZcI5d7

idp-sign-in https://dev-13121569.okta.com/app/dev-13121569\_appsaml\_1/exk5460t24OenZZcI5d7/sso/saml

trustpoint idp okta

trustpoint sp asa\_saml\_sp

signature rsa-sha256

sp-entity-id https://ztna.acme.com/finance/saml/sp/metadata

sp-acs-url <https://ztna.acme.com/finance/+CSCOE+/saml/sp/acs0x3Ftgname=DefaultZeroTrustGroup>

authentication-timeout 230

log enable

enable

application benefits.acme.com

application-id 268434452

application-interface BulkPhysicalNo1

internal-url https://wiki.acme.com:8080

external-url https://wiki.acme.com

mapped-port 9001

application-group-name finance <<- Optional, if its configured, it inherits group attributes

authentication-timeout 1440

log enable

enable

application bitbucket.acme.com <<- Standalone app, not part of group

application-id 268434450

application-interface BulkPhysicalNo1

internal-url http://bitbucket.acme.com:8080

external-url https://bitbucket.acme.com

idp-entity-id http://www.okta.com/exk6wek63kPB2J69u5d7

idp-sign-in https://dev-13121569.okta.com/app/dev-13121569\_appgroup\_1/exk6wek63kPB2J69u5d7/sso/saml

trustpoint idp okta

trustpoint sp asa\_saml\_sp

mapped-port 9002

signature rsa-sha256

sp-entity-id https:// bitbucket.acme.com/saml/sp/metadata/ bitbucket.acme.com

sp-acs-url https:// bitbucket.acme.com/+CSCOE+/saml/sp/acs0x3Ftgname=DefaultZeroTrustGroup

authentication-timeout 1440

log enable

enable

***This feature configuration is entirely driven from the FMC and none of the config CLIs discussed below will be available to users in CLISH for direct configuration of the feature from device. Only non config CLIs (such as show and clear commands) will be available to users in CLISH for displaying or troubleshooting the feature functionality.***

## Configuring ZTNA feature

To configure ZTNA feature new command mode named “zero-trust” is introduced which will contain all the ZTNA specific configuration.

**Syntax Description**

**zero-trust** Configure Zero Trust Network Access feature

zero-trust ?

   <cr>

**Command Default**

      None.

**ASA Context Mode**

      This command is specific to FTD deployments and won’t be available for configuration on ASA platforms.

**FTD Deployment Mode**

      Routed mode in Standalone, Failover, Cluster and MI deployments.

**Usage Guidelines**

To enter ZTNA configuration mode:

zero-trust

**Example**

(config)# zero-trust

(config-zero-trust)#

### Enabling/Disabling ZTNA

ZTNA configuration is a global config and can be enabled or disabled at system level using the following command.

**Syntax Description**

**enable** Enable zero-trust feature

   enable ?

      <cr>

**Command Default**

      None.

**ASA Context Mode**

     This command is specific to FTD deployments and won’t be available for configuration on ASA platforms.

**FTD Deployment Mode**

      Routed mode in Standalone, Failover, Cluster and MI deployments.

**Usage Guidelines**

       To enable ZTNA configuration:

zero-trust

        enable

**Example**

(config)# zero-trust

(config-zero-trust)# enable

(config-zero-trust)#

### Base Zero-trust URL for  assertion consumption

This base-url will be used by ACS (assertion consumption service) on FTD. It receives and processes the SAML assertions sent by IdP. Note that the URL is common for all the applications and groups. On IdP, while configuring an application, admin has to use this base URL along with the application or group tag.

Example: If base-url is “<https://ztna.acme.com>” and application-group is “finance”, then on Okta IdP SAML configuration for the application:

Single Sign On URL: https://ztna.acme.com/+CSCOE+/saml/sp/acs?tgname=DefaultZeroTrustGroup

Audience Restriction: https://ztna.acme.com/saml/sp/metadata/ztna.acme.com

Note: Above URLs will be generated by FMC while configuring application-group (or application). Admin has to copy it and edit the application definition on Okta.

**Syntax Description**

   [no] base-url <Zero-trust base URL>

   base-url ?

      <cr>

**Command Default**

      None.

**ASA Context Mode**

     This command is specific to FTD deployments and won’t be available for configuration on ASA platforms.

**FTD Deployment Mode**

      Routed mode in Standalone, Failover, Cluster and MI deployments.

**Usage Guidelines**

       To configure base-url:

zero-trust

        enable

base-url <>

**Example**

(config)# zero-trust

(config-zero-trust)# enable

(config-zero-trust)# base-url https://ztna.acme.com

### Session timeout for zero-trust sessions

Zero-trust authentications are valid for a period of time (e.g.: 24 \* 60 minutes). This time can be configured using authentication-timeout. After this period, authentication will be timed out and if user accesses application, will be redirected to IdP.

Default value of this will be 1440 minutes (24 hours).

**Syntax Description**

   [no] authentication-timeout <mins>

   authentication-timeout ?

      <30 – 10080> Authentication time out value in minutes, default is 1440.

**Command Default**

      None.

**ASA Context Mode**

     This command is specific to FTD deployments and won’t be available for configuration on ASA platforms.

**FTD Deployment Mode**

      Routed mode in Standalone, Failover, Cluster and MI deployments.

**Usage Guidelines**

       To configure base-url:

zero-trust

        enable

authentication-timeout

**Example**

(config)# zero-trust

(config-zero-trust)# enable

(config-zero-trust)# authentication-timeout 60

## Configuring Application Groups

Application group is a configuration construct which facilitates grouping of related applications so that when a user is authenticated against one of the applications in the group, he can have access to other applications in the group without a need for authenticating him against each of those applications. It just simplifies the authentication part. All user to application access requests will still be authorized before allowing access.

Application group and their corresponding configuration parameters such as certificates, IdP, SP, etc can be configured as explained below. Once application group and their corresponding parameters are configured, applications can be linked to the group.

### Configuring an application group

A new application can be configured using the CLI described below.

**Syntax Description**

**[no] application-group <app\_group\_name>** Configure application group which require zero-trust protection

application-group ?

WORD < 64 char Name of application group that requires zero-trust protection

**Command Default**

None.

**ASA Context Mode**

This command is specific to FTD deployments and won’t be available for configuration on ASA platforms.

**FTD Deployment Mode**

Routed mode in Standalone, Failover, Cluster and MI deployments.

**Usage Guidelines**

To configure an application group for zero-trust protection:

zero-trust

enable

application-group <app\_group\_name>

**Example**

(config)# zero-trust

(config-zero-trust)# enable

(config-zero-trust)# application-group finance

(config-zero-trust-application-group)#

### Enable zero-trust protection for the application group

Enabling an application group requires all mandatory configuration to be in place. If an application group is enabled without mandatory configuration in place, an error will be displayed indicating the missing mandatory configuration and the application group will not be enabled.

If an application group is not enabled, then applications that are part of this group are also not enabled. Enable at group overrides the enable on individual application.

**Syntax Description**

**enable** Enables application group for zero-trust protection

enable ?

<cr>

**Command Default**

None.

**ASA Context Mode**

This command is specific to FTD deployments and won’t be available for configuration on ASA platforms.

**FTD Deployment Mode**

Routed mode in Standalone, Failover, Cluster and MI deployments.

**Usage Guidelines**

To enable application group for zero-trust service:

zero-trust

enable

application-group <app\_group\_name>

enable

**Example**

(config)# zero-trust

(config-zero-trust)# enable

(config-zero-trust)# application-group finance

(config-zero-trust-application)# enable

### IdP entity ID for application group

All zero-trust protected application under the group needs to be configured with an identity provider URL for SAML based user authentication. Identity provider will authenticate the user to allow access to the zero-trust protected application.

Once Identity provider is configured for application group, the same will be inherited to all the applications under the group.

**Syntax Description**

**[no] idp-entity-id <url>** Configure SAML identity provider

idp-entity-id ?

WORD < 256 char This is a SAML IdP entityID

**Command Default**

None.

**ASA Context Mode**

This command is specific to FTD deployments and won’t be available for configuration on ASA platforms.

**FTD Deployment Mode**

Routed mode in Standalone, Failover, Cluster and MI deployments.

**Usage Guidelines**

To configure identity provider for an application group:

zero-trust

enable

application-group <app\_group\_name>

idp-entity-id <URL>

**Example**

(config)# zero-trust

(config-zero-trust)# enable

(config-zero-trust)# application-group finanacee

(config-zero-trust-application-group)# idp-entity-id <http://www.okta.com/exk5460t24OenZZcI5d7>

### SAML IdP sign-in URL for application group

All zero-trust protected applications under the group needs to be configured with an identity provider sign-in URL for SAML based user authentication. Identity provider will authenticate the user to allow access to the zero-trust protected application.

Identity provider sign-in URL configured for application group will be inherited to all the applications under that group

**Syntax Description**

**[no] idp-sign-in <url>** Configure the SAML IdP sign-in URL

idp-sign-in ?

WORD < 256 char Enter the URL (must begin with “http://" or “https://”)

**Command Default**

None.

**ASA Context Mode**

This command is specific to FTD deployments and won’t be available for configuration on ASA platforms.

**FTD Deployment Mode**

Routed mode in Standalone, Failover, Cluster and MI deployments.

**Usage Guidelines**

To configure identity provider sign-in URL:

zero-trust

enable

application-group <app\_group\_name>

idp-sign-in <URL>

**Example**

(config)# zero-trust

(config-zero-trust)# enable

(config-zero-trust)# application-group finance

(config-zero-trust-application-group)# idp-sign-in <https://dev-13121569.okta.com/app/dev-13121569_appsaml_1/exk5460t24OenZZcI5d7/sso/saml>

### IdP Trustpoint for an application group

The trust relationship between the FTD and the SAML Identity Provider is established through configured certificates (trustpoints). “trustpoint” is an existing ASA/FTD configuration entity which is being reused in zero-trust feature.

**Syntax Description**

**[no] trustpoint idp <name>** Configure trustpoint which contains the IdP certificate for FTD to verify SAML assertions

trustpoint idp ?

Available configured trustpoints:

[List of configured trustpoints]

**Command Default**

None.

**ASA Context Mode**

This command is specific to FTD deployments and won’t be available for configuration on ASA platforms.

**FTD Deployment Mode**

Routed mode in Standalone, Failover, Cluster and MI deployments.

**Usage Guidelines**

To configure IdP trustpoint for an application group:

zero-trust

enable

application-group <app\_group\_name>

trustpoint idp <name>

**Example**

(config)# zero-trust

(config-zero-trust)# enable

(config-zero-trust)# application-group finance

(config-zero-trust-application-group)# trustpoint idp okta\_idp

### Service Provider trustpoint for an application group

The trust relationship between the users/IdP and FTD (SP) is established through configured certificates (trustpoints). “trustpoint” is an existing ASA/FTD configuration entity which is being reused in zero-trust feature.

**Syntax Description**

**[no] trustpoint sp <name>** Configure trustpoint which contains the SP (FTDs) certificate for IdP to verify device’s signature or encrypt SAML assertion.

trustpoint sp ?

Available configured trustpoints:

[List of configured trustpoints]

**Command Default**

None.

**ASA Context Mode**

This command is specific to FTD deployments and won’t be available for configuration on ASA platforms.

**FTD Deployment Mode**

Routed mode in Standalone, Failover, Cluster and MI deployments.

**Usage Guidelines**

To configure service provider (FTD) trustpoint for an application group:

zero-trust

enable

application-group <app\_group\_name>

trustpoint sp <name>

**Example**

(config)# zero-trust

(config-zero-trust)# enable

(config-zero-trust)# application-group finance

(config-zero-trust-application-group)# trustpoint sp asa\_saml\_sp

### Configure signature for SAML authentication request for an application group

A signature can be configured for SAML authentication if required using below CLI. It is optional and disabled by default.

**Syntax Description**

**[no] signature <name>** Enable/disable signature in SAML request for an application. By default, signature is disabled.

signature ?

rsa-sha1 User rsa-sha1 for signing SAML authentication request

rsa-sha256 User rsa-sha256 for signing SAML authentication request (default)

rsa-sha384 User rsa-sha384 for signing SAML authentication request

rsa-sha512 User rsa-sha512 for signing SAML authentication request

**Command Default**

None.

**ASA Context Mode**

This command is specific to FTD deployments and won’t be available for configuration on ASA platforms.

**FTD Deployment Mode**

Routed mode in Standalone, Failover, Cluster and MI deployments.

**Usage Guidelines**

To configure signature for SAML auth request:

zero-trust

enable

application-group <application name>

signature rsa-sha256

**Example**

(config)# zero-trust

(config-zero-trust)# enable

(config-zero-trust)# application-group finance

(config-zero-trust-application-group)# signature rsa-sha512

## Configuring Applications

Applications that need zero trust protection and their corresponding configuration parameters such as certificates, ports, URLs, IdP, SP, etc can be configured as explained below.

### Configuring an application

A new application can be configured using the CLI described below.

**Syntax Description**

**[no] application <app\_name>** Configure applications which require zero-trust protection

      application ?

      WORD < 256 char Name of application that required zero-trust protection

**Command Default**

      None.

**ASA Context Mode**

     This command is specific to FTD deployments and won’t be available for configuration on ASA platforms.

**FTD Deployment Mode**

     Routed mode in Standalone, Failover, Cluster and MI deployments.

**Usage Guidelines**

       To configure an application with zero-trust protection

               zero-trust

                    enable

                application <app\_name>

**Example**

       (config)# zero-trust

       (config-zero-trust)# enable

       (config-zero-trust)# application wiki.acme.com

       (config-zero-trust-application)#

### Adding application to application group

By default, applications are not part of any group. To leverage the benefits of a group, an application can be linked to application group so that some of the parameters like certificates, idp will be inherited from the group. User need not configure these parameters separately for each of the application.

**Syntax Description**

**enable** Enables application for zero-trust protection

enable ?

<cr>

**Command Default**

None.

**ASA Context Mode**

This command is specific to FTD deployments and won’t be available for configuration on ASA platforms.

**FTD Deployment Mode**

Routed mode in Standalone, Failover, Cluster and MI deployments.

**Usage Guidelines**

To add application to previously configured application group:

zero-trust

enable

application <app\_name>

application-group <app\_group\_name>

**Example**

(config)# zero-trust

(config-zero-trust)# enable

(config-zero-trust)# application finance-audit-server

(config-zero-trust-application)# application-group finance

(config-zero-trust-application)#

### Enable zero-trust protection for the application

Enabling an application requires all mandatory configuration to be in place. If an application is enabled without mandatory configuration in place, an error will be displayed indicating the missing mandatory configuration and the application will not be enabled.

**Syntax Description**

**enable** Enables application for zero-trust protection

      enable ?

      <cr>

**Command Default**

      None.

**ASA Context Mode**

      This command is specific to FTD deployments and won’t be available for configuration on ASA platforms.

**FTD Deployment Mode**

      Routed mode in Standalone, Failover, Cluster and MI deployments.

**Usage Guidelines**

      To enable application for zero-trust service

                zero-trust

                    enable

                application <app\_name>

                    enable

**Example**

       (config)# zero-trust

       (config-zero-trust)# enable

       (config-zero-trust)# application wiki.acme.com

       (config-zero-trust-application)#enable

### Exposing application on an interface

Applications need to be exposed on interface through which they will be accessed. As an example, an internal application needs to be exposed on public interface of FTD if it is intended to be accessed by public internet hosts.

**Syntax Description**

**[no] interface <interface\_name>** Configure the interface on which to expose the application

    interface ?

    Current available interface(s):

    [List of routed interfaces]

**Command Default**

    None.

**ASA Context Mode**

    This command is specific to FTD deployments and won’t be available for configuration on ASA platforms.

**FTD Deployment Mode**

    Routed mode in Standalone, Failover, Cluster and MI deployments.

**Usage Guidelines**

       To enable application on an interface(s):

                zero-trust

                    enable

                application <app\_name>

                    interface <interface\_name>

**Example**

       (config)# zero-trust

       (config-zero-trust)# enable

       (config-zero-trust)# application wiki.acme.com

       (config-zero-trust-application)# interface Outside

**Note:** “interface <interface\_name>” CLI can be executed multiple times to expose same application on multiple interfaces.

### Configure IdP for SAML based authentication

Each zero-trust protected application needs to be configured with an identity provider URL for SAML based user authentication. Identity provider will authenticate the user to allow access to the zero-trust protected application.

**Syntax Description**

**[no] idp-entity-id <url>** Configure SAML identity provider

      idp-entity-id ?

      WORD < 256 char This is a SAML IdP entityID

**Command Default**

      None.

**ASA Context Mode**

     This command is specific to FTD deployments and won’t be available for configuration on ASA platforms.

**FTD Deployment Mode**

      Routed mode in Standalone, Failover, Cluster and MI deployments.

**Usage Guidelines**

       To configure identity provider for an application:

                zero-trust

                    enable

                application <app\_name>

                    idp-entity-id <URL>

**Example**

(config)# zero-trust

(config-zero-trust)# enable

(config-zero-trust)# application wiki.acme.com

(config-zero-trust-application)# idp-entity-id <http://www.okta.com/exk5460t24OenZZcI5d7>

### Configure SAML identity provider sign-in URL

Each zero-trust protected application needs to be configured with an identity provider sign-in URL for SAML based user authentication. Identity provider will authenticate the user to allow access to the zero-trust protected application.

**Syntax Description**

**[no] idp-sign-in <url>** Configure the SAML IdP sign-in URL

      idp-sign-in ?

      WORD < 500 char Enter the URL (must begin with “http://" or “https://”)

**Command Default**

      None.

**ASA Context Mode**

      This command is specific to FTD deployments and won’t be available for configuration on ASA platforms.

**FTD Deployment Mode**

      Routed mode in Standalone, Failover, Cluster and MI deployments.

**Usage Guidelines**

     To configure identity provider sign-in URL:

                zero-trust

                    enable

                application <app\_name>

                    idp-sign-in <URL>

**Example**

       (config)# zero-trust

       (config-zero-trust)# enable

       (config-zero-trust)# application wiki.acme.com

       (config-zero-trust-application)# idp-sign-in <https://dev-13121569.okta.com/app/dev-13121569_appsaml_1/exk5460t24OenZZcI5d7/sso/saml>

### Configure external URL for zero-trust application

Each application will have an external URL, which resolves to a configured FTD interface IP address. End user access the application using this URL.

**Syntax Description**

**[no] external-url <URL>** Configure external URL for accessing the application

      external-url ?

      WORD < 256 char This is external URL of application (must begin with “http://” or “https://”)

**Command Default**

      None.

**ASA Context Mode**

      This command is specific to FTD deployments and won’t be available for configuration on ASA platforms.

**FTD Deployment Mode**

      Routed mode in Standalone, Failover, Cluster and MI deployments.

**Usage Guidelines**

       To configure external URL for an application:

                zero-trust

                    enable

                application <app\_name>

                    external-url <URL>

**Example**

       (config)# zero-trust

       (config-zero-trust)# enable

       (config-zero-trust)# application wiki.acme.com

       (config-zero-trust-application)# external-url [https://wiki.acme.com](https://wiki.acme.com/)

### Configure internal URL for zero-trust application

Each application needs an internal URL. This URL resolves to private application hosted behind FTD. This URL is used to configure an implicit NAT rules to translate <FTD-IP>:<mapped port> to <internal application server IP>:<port>.

**Syntax Description**

**[no] internal-url <URL>** Configure external URL for accessing the application

     internal-url ?

     WORD < 256 char This is external URL of application (must begin with “http://” or “https://”)

**Command Default**

     None.

**ASA Context Mode**

     This command is specific to FTD deployments and won’t be available for configuration on ASA platforms.

**FTD Deployment Mode**

      Routed mode in Standalone, Failover, Cluster and MI deployments.

**Usage Guidelines**

       To configure internal URL for an application:

                zero-trust

                    enable

                application <app\_name>

                    internal-url <URL>

**Example**

       (config)# zero-trust

       (config-zero-trust)# enable

       (config-zero-trust)# application wiki.acme.com

       (config-zero-trust-application)# internal-url  [https://wiki-internal.acme.com:8080](https://wiki-internal.acme.com:8080/)

### Configure mapped port associated with the application

Each application needs a mapped port. The user will be redirected to application’s mapped port post successful authentication. FTD will be configured with an implicit NAT rule to translate all connections destined to <FTD-IP>:<mapped port> to <internal application server IP>:<port>.

**Syntax Description**

     [no] mapped-port <port\_number> Configure mapped port of the application

     mapped-port ?

     <1-65535> Enter mapped port number

**Command Default**

      None.

**ASA Context Mode**

      This command is specific to FTD deployments and won’t be available for configuration on ASA platforms.

**FTD Deployment Mode**

      Routed mode in Standalone, Failover, Cluster and MI deployments.

**Usage Guidelines**

       To configure mapped port for an application:

                zero-trust

                    enable

                application <app\_name>

                    mapped-port <port\_number>

**Example**

       (config)# zero-trust

       (config-zero-trust)# enable

       (config-zero-trust)# application wiki.acme.com

       (config-zero-trust-application)# mapped-port 9000

### Configure IdP Trustpoint for an application

The trust relationship between the FTD and the SAML Identity Provider is established through configured certificates (trustpoints). “trustpoint” is an existing ASA/FTD configuration entity which is being reused in zero-trust feature.

**Syntax Description**

**[no] trustpoint idp <name>** Configure trustpoint which contains the IdP certificate for FTD to verify SAML assertions

      trustpoint idp ?

      Available configured trustpoints:

      [List of configured trustpoints]

**Command Default**

      None.

**ASA Context Mode**

      This command is specific to FTD deployments and won’t be available for configuration on ASA platforms.

**FTD Deployment Mode**

      Routed mode in Standalone, Failover, Cluster and MI deployments.

**Usage Guidelines**

       To configure IdP trustpoint for an application:

                zero-trust

                    enable

                application <app\_name>

                    trustpoint idp <name>

**Example**

       (config)# zero-trust

       (config-zero-trust)# enable

       (config-zero-trust)# application wiki.acme.com

       (config-zero-trust-application)# trustpoint idp okta\_idp

### Configure Service Provider (SP) trustpoint for an application

The trust relationship between the users/IdP and FTD (SP) is established through configured certificates (trustpoints). “trustpoint” is an existing ASA/FTD configuration entity which is being reused in zero-trust feature.

**Syntax Description**

**[no] trustpoint sp <name>** Configure trustpoint which contains the SP (FTDs) certificate for IdP to verify device’s signature or encrypt SAML assertion.

       trustpoint sp ?

      Available configured trustpoints:

      [List of configured trustpoints]

**Command Default**

     None.

**ASA Context Mode**

     This command is specific to FTD deployments and won’t be available for configuration on ASA platforms.

**FTD Deployment Mode**

     Routed mode in Standalone, Failover, Cluster and MI deployments.

**Usage Guidelines**

       To configure service provider (FTD) trustpoint for an application:

                zero-trust

                    enable

                application <app\_name>

                    trustpoint sp <name>

**Example**

       (config)# zero-trust

       (config-zero-trust)# enable

       (config-zero-trust)# application wiki.acme.com

       (config-zero-trust-application)# trustpoint sp asa\_saml\_sp

### Configure signature for SAML authentication request for an application

A signature can be configured for SAML authentication if required using below CLI. It is optional and disabled by default.

**Syntax Description**

**[no] signature <name>** Enable/disable signature in SAML request for an

application. By default, signature is disabled.

     signature ?

     rsa-sha1 User rsa-sha1 for signing SAML authentication request

     rsa-sha256 User rsa-sha256 for signing SAML authentication request (default)

     rsa-sha384 User rsa-sha384 for signing SAML authentication request

     rsa-sha512 User rsa-sha512 for signing SAML authentication request

**Command Default**

     None.

**ASA Context Mode**

     This command is specific to FTD deployments and won’t be available for configuration on ASA platforms.

**FTD Deployment Mode**

     Routed mode in Standalone, Failover, Cluster and MI deployments.

**Usage Guidelines**

       To configure signature for SAML auth request:

                zero-trust

                    enable

                application <application name>

                     signature rsa-sha256

**Example**

       (config)# zero-trust

       (config-zero-trust)# enable

       (config-zero-trust)# application wiki.acme.com

       (config-zero-trust-application)# signature rsa-sha512

### Enable/disable force-reauthentication for an application

“force re-authentication" will cause the identity provider to authenticate directly rather than relying on a previous security context when a SAML authentication request occurs. This setting is enabled by default; therefore, to disable, use “no force re-authentication”.

**Syntax Description**

**[no] force re-authentication** Enable/disable force re-authentication of an application.

By default, re-authentication is enabled.

       force re-authentication ?

       <cr>

**Command Default**

      None.

**ASA Context Mode**

      This command is specific to FTD deployments and won’t be available for configuration on ASA platforms.

**FTD Deployment Mode**

      Routed mode in Standalone, Failover, Cluster and MI deployments.

**Usage Guidelines**

       To enable/disable force re-authentication for an application

    zero-trust

        enable

    application <app\_name>

        force re-authentication

**Example**

       (config)# zero-trust

       (config-zero-trust)# enable

       (config-zero-trust)# application wiki.acme.com

       (config-zero-trust-application)# force re-authentication

### Configure Source NAT for an application

Source NAT can be configured by “mapped-source-v4” keyword under zero-trust application app\_name CLI. It is optional and disabled by default.

**Syntax Description**

**firepower(config-zero-trust-application)# mapped-source-v4 ?**

config-zero-trust-application mode commands/options:

WORD < 129 char Specify object or object-group name for source translation

of traffic destined to zero-trust application.

**Command Default**

     None.

**ASA Context Mode**

     This command is specific to FTD deployments and won’t be available for configuration on ASA platforms.

**FTD Deployment Mode**

     Routed mode in Standalone, Failover, Cluster and MI deployments.

**Usage Guidelines**

       To configure signature for SAML auth request:

                zero-trust

                    enable

                application <application name>

                     mapped-source-v4 <obj\_name>

**Example**

       (config)# zero-trust

       (config-zero-trust)# enable

       (config-zero-trust)# application wiki.acme.com

       (config-zero-trust-application)# mapped-source-v4 host\_obj

## 7.4 Show Commands

The following show run commands are provided to display the user configured values as well as the run-time state of the feature.

### 7.4.1 Show Config

Zero-trust configuration can be displayed using the following show command.

**Syntax Description**

      show running-config zero-trust [application <app\_name>] [application-group <group\_name>]

**Command Default**

      None.

**ASA Context Mode**

      This command is specific to FTD deployments and won’t be available for configuration on ASA platforms.

**FTD Deployment Mode**

      Routed mode in Standalone, Failover, Cluster and MI deployments.

**Usage Guidelines**

To display all configurations under zero-trust:

    show running-config zero-trust

To display configurations of all applications:

    show running-config zero-trust application

      To display configuration of a particular application:

    show running-config zero-trust application <app\_name>

To display configurations of all application-groups

    show running-config zero-trust application-group

      To display configuration of a particular application-group

    show running-config zero-trust application-group <group\_name>

**Example**

(config)# show running-config zero-trust

zero-trust

    enable

    application wiki.acme.com

       enable

       interface Inside

       idp-entity-id http://www.okta.com/exk5460t24OenZZcI5d7

       idp-sign-in https://dev-13121569.okta.com/app/dev-13121569\_appsaml\_1/exk5460t24OenZZcI5d7/sso/saml

       external-url https://wiki.acme.com

       internal-url  https://wiki-internal.acme.com:8080

       mapped-port 9000

        trustpoint idp okta

        trustpoint sp asa\_saml\_sp

        signature rsa-sha256

        force re-authentication

(config)# show running-config zero-trust application

application wiki.acme.com

       enable

       interface Inside

       idp-entity-id http://www.okta.com/exk5460t24OenZZcI5d7

       idp-sign-in https://dev-13121569.okta.com/app/dev-13121569\_appsaml\_1/exk5460t24OenZZcI5d7/sso/saml

       external-url https://wiki.acme.com

       internal-url  https://wiki-internal.acme.com:8080

       mapped-port 9000

        trustpoint idp okta

        trustpoint sp asa\_saml\_sp

        signature rsa-sha256

        force re-authentication

application wiki.zascorp.com

       enable

       interface In

       idp-entity-id http://www.okta.com/exk5460t24OenZZcI5d7

       idp-sign-in https://dev-13121569.okta.com/app/dev-13121569\_appsaml\_1/exk5460t24OenZZcI5d7/sso/saml

       external-url https://wiki.zascorp.com

       internal-url  https://wiki-internal.zascorp.com:8080

       mapped-port 9002

        trustpoint idp okta

        trustpoint sp asa\_saml\_sp

        signature rsa-sha256

(config)# show running-config zero-trust application wiki.acme.com

application wiki.acme.com

       enable

       interface Inside

       idp-entity-id http://www.okta.com/exk5460t24OenZZcI5d7

       idp-sign-in https://dev-13121569.okta.com/app/dev-13121569\_appsaml\_1/exk5460t24OenZZcI5d7/sso/saml

       external-url https://wiki.acme.com

       internal-url  https://wiki-internal.acme.com:8080

       mapped-port 9000

        trustpoint idp okta

        trustpoint sp asa\_saml\_sp

        signature rsa-sha256

        force re-authentication

### 7.4.2 Show Run-time State

The following show commands are provided to display the runtime state and statistics of zero-trust feature.

**Syntax Description**

      show zero-trust ?

         sessions [user <username>] [application <app\_name>] [detail]

         statistics

**Command Default**

      None

**ASA Context Mode**

      This command is specific to FTD deployments and won’t be available for configuration on ASA platforms.

**FTD Deployment Mode**

      Routed mode in Standalone, Failover, Cluster and MI deployments.

**Usage Guidelines**

To display the runtime zero-trust sessions for all users and applications:

    show zero-trust sessions

To display the runtime zero-trust sessions count

    show zero-trust sessions count

To display the runtime zero-trust sessions of a specific user:

    show zero-trust sessions user <username>

To display the runtime zero-trust sessions of a specific application:

    show zero-trust sessions application <app\_name>

To display the detailed runtime zero-trust information for any of the above options:

    show zero-trust sessions [user <username>] [application <app\_name>] detail

       To display the runtime zero-trust statistics:

    show zero-trust statistics

**Example**

(config)# show zero-trust sessions

Sessions display order: User, Application, Application-Group, Src Ip, Sessions test@cisco.com, wiki.ztna.com, parent, 172.16.77.1, 1  
ironman@cisco.com, wiki.bitbucket.com, bitbucket\_grp, 172.16.77.1, 1 test@cisco.com, wiki.outlook.com, None, 172.16.77.1, 1  
test@cisco.com, wiki.confluence.com, parent, 172.16.77.1, 1

>show zero-trust sessions count

5 in use, 156 most used

(config)show zero-trust sessions detail

Sessions display order: User, Application, Application-Group, Src Ip, Cookie, Expiry Time test@cisco.com, wiki.ztna.com, None, 172.16.77.1, E194C7F0..., 23:54:53  
test@cisco.com, wiki.confluence.com, None, 172.16.77.1, F9E330A4..., 23:55:05

(config)# show zero-trust statistics

Zero-trust sessions (total/active/failed) 7/0/0

Users (total/active) 3/2

Applications (total/enabled/active) 4/4/0

Authentications (total/in-progress) 4/6

SAML requests (total/passed/failed) 10/10/0

SAML responses (total/passed/failed) 4/4/0

Authentication latency (min/max/avg) 254/384/1062 ms

Total bytes in 16242 Bytes

Total bytes out 81092 Bytes

Last clearing of stats: @ 06:42:20 UTC Aug 3 2023

### 7.4.3 Show Cluster-wide Run-time State

The following show commands are provided to display the runtime values in cluster mode.

**Syntax Description**

      show cluster zero-trust ?

      statistics

**Command Default**

      None.

**ASA Context Mode**

      This command is specific to FTD deployments and won’t be available for configuration on ASA platforms.

**FTD Deployment Mode**

      Routed mode in Standalone, Failover, Cluster and MI deployments.

**Usage Guidelines**

       To display the cluster wide runtime zero-trust statistics:

    show cluster zero-trust statistics

**Example**

(config)# show cluster zero-trust statistics

Usage Summary In Cluster:\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Zero-trust sessions (total/active/failed) 7/0/0

Users (total/active) 3/2

Applications (total/enabled/active) 4/4/0

Authentications (total/in-progress) 4/6

SAML requests (total/passed/failed) 10/10/0

SAML responses (total/passed/failed) 4/4/0

Authentication latency (min/max/avg) 254/384/1062 ms

Total bytes in 16242 Bytes

Total bytes out 81092 Bytes

Last clearing of stats: @ 06:42:20 UTC Aug 3 2023

unit-1-1(LOCAL):\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Zero-trust sessions (total/active/failed) 7/0/0

Users (total/active) 3/2

Applications (total/enabled/active) 4/4/0

Authentications (total/in-progress) 4/6

SAML requests (total/passed/failed) 10/10/0

SAML responses (total/passed/failed) 4/4/0

Authentication latency (min/max/avg) 254/384/1062 ms

Total bytes in 16242 Bytes

Total bytes out 81092 Bytes

Last clearing of stats: @ 06:42:20 UTC Aug 3 2023

### 7.4.4 Show conn ZTNA flow tagging:

In "show conn all detail" the new flag Z1 will represent the ZTNA PostAuth flow.

>show conn detail

0 in use, 19 most used

Inspect Snort:

preserve-connection: 0 enabled, 0 in effect, 1 most enabled, 0 most in effect

Flags: A - awaiting responder ACK to SYN, a - awaiting initiator ACK to SYN,

B - TCP probe for server certificate,

b - TCP state-bypass or nailed,

C - CTIQBE media, c - cluster centralized,

D - DNS, d - dump, E - outside back connection, e - semi-distributed,

F - initiator FIN, f - responder FIN,

G - group, g - MGCP, H - H.323, h - H.225.0, I - initiator data,

i - incomplete, J - GTP, j - GTP data, K - GTP t3-response

k - Skinny media, L - decap tunnel, M - SMTP data, m - SIP media

N - inspected by Snort (1 - preserve-connection enabled, 2 - preserve-connection in effect,

3 - elephant-flow, 4 - elephant-flow bypassed, 5 - elephant-flow throttled, 6 - elephant-flow exempted )

n - GUP, O - responder data, o - offloaded,

P - inside back connection, p - passenger flow,

Q - QUIC, q - SQL\*Net data, R - initiator acknowledged FIN,

R - UDP SUNRPC, r - responder acknowledged FIN,

T - SIP, t - SIP transient, U - up,

V - VPN orphan, v - M3UA W - WAAS,

w - secondary domain backup,

X - inspected by service module,

x - per session, Y - director stub flow, y - backup stub flow,

Z - Scansafe redirection, Z1 - zero-trust flow, z - forwarding stub flow

TCP outside: 11.77.2.197/58931 inside: 11.78.2.90/443,

flags UIOiNZ1, idle 1s, uptime 2s, timeout 1h0m, bytes 858, xlate id 0x151dc6e4d1c0, Rx-RingNum invalid, Internal-Data invalid

Initiator: 11.77.2.197, Responder: 11.78.2.90

Connection lookup keyid: 286232462

Debug enabled flows: Forward, Reverse

## Clear Commands

Zero-trust runtime sessions and statistics can be cleared using the following clear commands.

### Clear Config

“zero-trust” configuration can be removed using the following CLIs.

**Syntax Description**

 clear configure [zero-trust [application <app\_name>]]

**Command Default**

      None.

**ASA Context Mode**

      This command is specific to FTD deployments and won’t be available for configuration on ASA platforms.

**FTD Deployment Mode**

      Routed mode in Standalone, Failover, Cluster and MI deployments.

**Usage Guidelines**

To remove the zero-trust configuration of a specific application:

    clear configure zero-trust application <app\_name>

To remove the zero-trust configuration of all applications:

    clear configure zero-trust application

To remove all zero-trust configuration:

    clear configure zero-trust

**Example**

(config)# clear configure zero-trust application wiki.cisco.com

(config)#

(config)# clear configure zero-trust application

(config)#

(config)# clear configure zero-trust

### Clear Run-time State

The run-time state and statistics of zero-trust feature can be cleared using the following CLIs.

**Syntax Description**

      clear zero-trust ?

      sessions [user <username>] [application <app\_name>]

      statistics

**Command Default**

      None.

**ASA Context Mode**

      This command is specific to FTD deployments and won’t be available for configuration on ASA platforms.

**FTD Deployment Mode**

      Routed mode in Standalone, Failover, Cluster and MI deployments.

**Usage Guidelines**

To clear the zero-trust sessions of all users and applications:

    clear zero-trust sessions

To clear the zero-trust sessions of a specific user:

    clear zero-trust sessions user <username>

To clear the zero-trust sessions of a specific application:

    clear zero-trust sessions application <app\_name>

To clear the zero-trust sessions of a specific user and application:

    clear zero-trust sessions user <user\_name> application <app\_name>

       To clear the runtime zero-trust statistics:

    clear zero-trust statistics

**Example**

(config)# clear zero-trust sessions

16 zero-trust authorization sessions removed.

(config)# clear zero-trust sessions user ladygaga@cisco.com

2 zero-trust authorization sessions removed.

(config)# clear zero-trust sessions application wiki.cisco.com

1 zero-trust authorization session removed.

(config)# clear zero-trust statistics

### Clear Cluster-wide Run-time State

Zero-trust cluster wide runtime sessions and statistics can be cleared using the following clear commands.

**Syntax Description**

      cluster exec clear zero-trust ?

          sessions [user <username>] [application <app\_name>]

          statistics

**Command Default**

      None.

**ASA Context Mode**

      This command is specific to FTD deployments and won’t be available for configuration on ASA platforms.

**FTD Deployment Mode**

      Routed mode in Standalone, Failover, Cluster and MI deployments.

**Usage Guidelines**

To clear the runtime zero-trust sessions across cluster nodes for all users and applications:

    cluster exec clear zero-trust sessions

To clear the runtime zero-trust sessions across cluster nodes for a specific user:

    cluster exec clear zero-trust sessions user <username>

To clear the runtime zero-trust sessions across cluster nodes for a specific application:

    cluster exec clear zero-trust sessions application <app\_name>

To clear the runtime zero-trust sessions across cluster nodes for a specific user and application:

    cluster exec clear zero-trust sessions user <username> application <app\_name>

       To clear the runtime zero-trust statistics across cluster nodes:

                   cluster exec clear zero-trust statistics

**Example**

(config)# cluster exec clear zero-trust sessions

16 zero-trust authorization sessions removed.

(config)# cluster exec clear zero-trust sessions user ladygaga@cisco.com

2 zero-trust authorization sessions removed.

(config)# cluster exec clear zero-trust sessions application wiki.cisco.com

1 zero-trust authorization session removed.

(config)# cluster exec clear zero-trust statistics

### Show Snort Statistics

Existing “show snort statistics” command is enhanced to display Snort related zero-trust specific counters on the Lina side.

**Syntax Description**

show snort statistics

**Command Default**

None.

**ASA Context Mode**

These commands are specific to FTD deployments and won’t be available for configuration on ASA platforms.

**FTD Deployment Mode**

Standalone, Failover, Cluster and MI modes.

**Usage Guidelines**

To display zero-trust specific counters in Lina:

show snort statistics

**Example**

> show snort statistics

Packet Counters:

  Passed Packets                                                  79434

  Blocked Packets                                                  2208

  Injected Packets                                                         0

  Packets bypassed (Snort Down)                                0

  Packets bypassed (Snort Busy)                                  0

Flow Counters:

  Fast-Forwarded Flows                                              86

  Blacklisted Flows                                                     35

Miscellaneous Counters:

  Start-of-Flow events                                                 68

  End-of-Flow events                                                  68

  Denied flow events                                                     0

  Frames forwarded to Snort before drop                      0

  Inject packets dropped                                                0

  TCP Ack bypass Packets                                            0

  TCP Meta-Ack Packets                                               0

  Portscan Events                                                           0

  Zero-trust token messages                                     27

  Packet decode optimized                                             0

  Packet decode legacy                                           79424

### Show Snort Counters

Existing “show snort counters” command is enhanced to display zero-trust specific counters on the Snort side.

**Syntax Description**

show snort counters

**Command Default**

None.

**ASA Context Mode**

These commands are specific to FTD deployments and won’t be available for configuration on ASA platforms.

**FTD Deployment Mode**

Standalone, Failover, Cluster and MI modes.

**Usage Guidelines**

To display zero-trust specific counters in Snort:

show snort counters

**Example**

> show snort counters

<snip>

--------------------------------------------------

Module Statistics

--------------------------------------------------

<snip>

zero-trust

           http\_header\_events: 79413

                         ztna\_flows: 79413

               cookies\_matched: 74018

        cookies\_not\_matched: 3

        http\_sessions\_blocked: 456

                         ztna\_events: 3

                    invalid\_cookie: 0

                           no\_cookie: 0

--------------------------------------------------

         <snip>

### Show Counters

Existing “show counters” command is enhanced to display zero-trust specific counters on the Lina side.

[TBD]

## Debug Commands

Following debug commands can be used to troubleshoot issues related to zero-trust configuration, application authentication/authorization, and application traffic inspection functionality.

### Connection Based Debugs

A new “zero-trust” module will be added to the “connection based debugging” infrastructure to support debugging and troubleshooting of zero trust functionality. This can be used to troubleshoot issue related to run time authentication, authorization, and inspection of application traffic and zero trust functionality.

**Syntax Description**

[no] debug packet-module

zero-trust Zero-trust information

**Command Default**

None.

**ASA Context Mode**

These commands are specific to FTD deployments and won’t be available for configuration on ASA platforms.

**FTD Deployment Mode**

Standalone, Failover, Cluster and MI modes.

**Usage Guidelines**

To enable “zero-trust” module debugs at informational level:

debug packet-module zero-trust 6

To filter “zero-trust” module debugs in output:

show packet debugs module zero-trust

**Example**

> debug packet-condition match tcp host 1.1.1.1 host 2.1.1.1

> debug packet-module zero-trust 6

>

> show packet-config

Debugging is OFF

Conditional debug filters:

debug packet-condition position 1

match tcp host 1.1.1.1 host 2.1.1.1

Conditional debug features:

debug packet-module zero-trust 6

### Non-connection/Config Based Debugs

Following debug command can be used to troubleshoot issues related to zero-trust configuration handling.

**Syntax Description**

[no] debug zero-trust <1-255>

**Command Default**

None.

**ASA Context Mode**

These commands are specific to FTD deployments and won’t be available for configuration on ASA platforms.

**FTD Deployment Mode**

Standalone, Failover, Cluster and MI modes.

**Usage Guidelines**

To enable “zero-trust” module debugs at level 5:

debug zero-trust 5

To disable “zero-trust” module debugs at level 5:

no debug zero-trust

**Example**

(config)# debug zero-trust 5

debug zero-trust enabled at level 5

(config)#

(config)# show debug

debug zero-trust enabled at level 5

(config)#

(config)# no debug zero-trust

### Debug Menu Commands

Following debug menu command can be used to modify/fine-tune zero-trust functionality.

**Syntax Description**

[no] debug menu zero-trust

**Command Default**

None.

**ASA Context Mode**

These commands are specific to FTD deployments and won’t be available for configuration on ASA platforms.

**FTD Deployment Mode**

Standalone, Failover, Cluster and MI modes.

**Usage Guidelines**

To enable “zero-trust” feature:

debug menu zero-trust 1

To disable “zero-trust” feature:

debug menu zero-trust 2

To view “zero-trust” feature status:

debug menu zero-trust 3

**Example**

(config)# debug menu zero-trust 1

(config)#

(config)# debug menu zero-trust 3

Zero-trust feature is enabled

(config)#

(config)# debug menu zero-trust 2

(config)#

(config)# debug menu zero-trust 3

Zero-trust feature is not enabled

#### 7.5.4 Packet Tracer Enhancement for zero-trust flow

A new snort zero-trust phase will be present in the packet tracer and packet capture which will help to debug zero-trust flows.

----------------------------------------------------------

For SYN Packet

----------------------------------------------------------

Phase: 18

Type: EXTERNAL-INSPECT

Subtype:

Result: ALLOW

Elapsed time: 57405 ns

Config:

Additional Information:

Application: 'SNORT Inspect'

Phase: 19

Type: SNORT

Subtype: zero-trust

Result: ALLOW

Elapsed time: 71107 ns

Config:

Rule Id : 268434437

Additional Information:

Enforcing Rule : Rule matched; allowing flow

Phase: 20

Type: SNORT

Subtype: appid

Result: ALLOW

Elapsed time: 46529 ns

Config:

Additional Information:

service: (0), client: (0), payload: (0), misc: (0)

------------------------------------------------------------

For Application Packet

------------------------------------------------------------

Phase: 4

Type: EXTERNAL-INSPECT

Subtype:

Result: ALLOW

Elapsed time: 75205 ns

Config:

Additional Information:

Application: 'SNORT Inspect'

Phase: 5

Type: SNORT

Subtype: zero-trust

Result: ALLOW

Elapsed time: 836825 ns

Config:

Rule Id : 268434437

Additional Information:

flow authorized successfully

Phase: 6

Type: SNORT

Subtype: appid

Result: ALLOW

Elapsed time: 245014 ns

Config:

Additional Information:

service: HTTP(676), client: Firefox(638), payload: (0), misc: HTTPS(1122)

## Syslog Messages

This Feature does not add any new syslog. The existing syslogs will continue to work as they are.

## Events

### Connection Events

* Once the Zero-trust application is deployed, connection events will be generated. These events will have the following new fields populated with the Zero Trust policy names: Zero Trust Application, Zero Trust Application Group, and Zero Trust Application Policy.
* The User field will be populated with the identified username from the identity provider​.
* The new Authentication Source field will be populated with “Zero Trust"​

### Dashboards:

* A new "Zero Trust" tab is added to the default Summary Dashboard.
* This new tab will contain 2 widgets

                1. Top Zero Trust Applications (by total traffic)

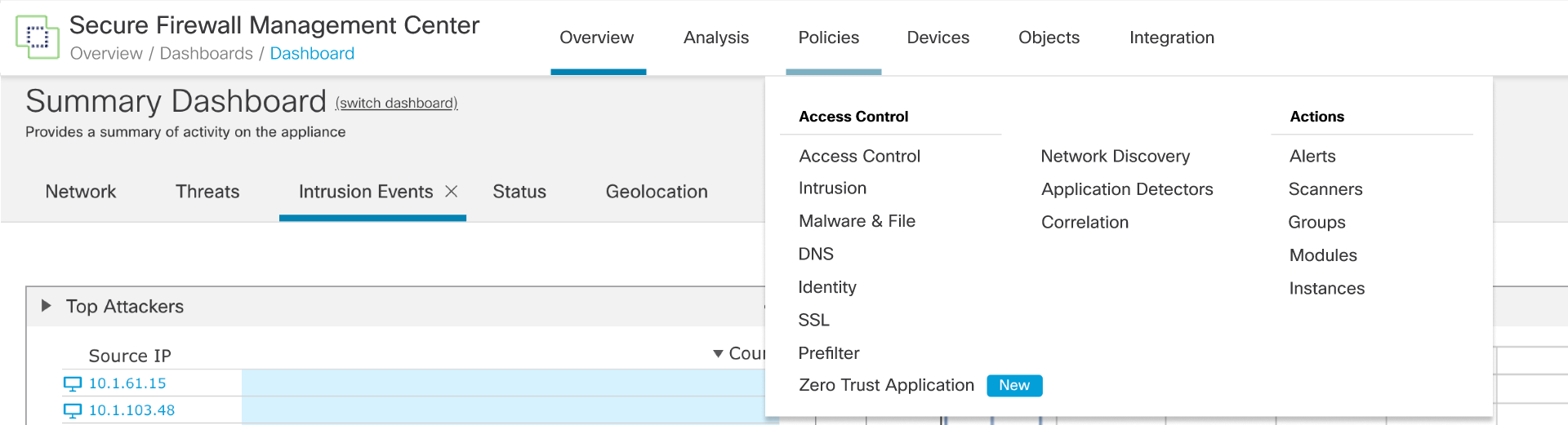
                2. Top Zero Trust Users (by total traffic)

## Configuring through CLISH

Only non-exec mode CLIs are exposed by default in CLISH implying that “show”, “debug” and “clear” command will be available at CLISH prompt.

## Configuring through FMC

ZTNA is modelled as a new policy group on the FMC accessible through the main menu



### ZTNA Policy

The specific configuration can be set for the policy group on creation

Graphical user interface, text, application, email

Description automatically generated

The applications & application group can be created after policy creation

Graphical user interface, text

Description automatically generated

And after adding the application’s and application’s group they will be listed as shown inside the policy

Graphical user interface, text, application, email

Description automatically generated

#### Application Group

An application group can be created as shown below

A picture containing application

Description automatically generated

and the attributes can configured for the group which will be used by the application belonging to the group by default

Graphical user interface

Description automatically generated

#### Application

An application can be added similarly

Graphical user interface, application

Description automatically generated

The IDP & SAML for the application can be configured as shown

Graphical user interface, application

Description automatically generated Graphical user interface, application

Description automatically generated

### HW Modeling

The ZTNA policy is modelled as a new policy group and the application & application groups are modeled as policy to ensure proper integration with the existing framework & its functionality.

#### Database Model

**HIERARCHY\_NODE**

It stores the common data of a policy group. This is an existing table and there is no schema change in it. A new entry for a policy group type PG.FIREWALL.ZTNAPolicy is made here

**POLICY\_MAIN**

It stores the common data of a policy. This is an existing table and there is no schema change in it. A new entry of policy type ZTNAPolicy is made here on creating a new ZTNA Policy. Similarly new entries of type ZTNAApplicationPolicy are made when new applications are inserted into the policy.

**ZTNA\_POLICY\_SETTING**

It stores the data for a ZTNA policy, there would be one entry per policy.

**ZTNA\_APPLICATION**

It stores the data for both application and group. There could be multiple entries per policy.

### DUO API Integration

The DUO API is not available for the integration. Once the DUO api is available the plan is to push the SAML data to the DUO and get the IDP information from it without any manual copy pasting.

## FMC Deployment

The FMC deploys the configuration to the FTD using the lina CLI.

zero-trust

base-url https://ztna.acme.com

enable

log enable

max-applications <number>

cookie-validity <mins>

application-group finance

enable

idp-entity-id < >

idp-sign-in < >

trustpoint idp okta

trustpoint sp asa\_saml\_sp

signature rsa-sha256

log enable

application benefits.acme.com

application-group finance <<- Optional, if its configured, it inherits group attrs

interface BulkPhysicalNo1

internal-url http://172.16.4.1

external-url https://wiki.acme.com

mapped-port 9001

force re-authentication

idp-entity-id < > << same as config of group

idp-sign-in < > << same as config of group

trustpoint idp okta << same as config of group

trustpoint sp asa\_saml\_sp << same as config of group

signature rsa-sha256 << same as config of group

log enable

enable

## Configuring through ASDM/FDM

Clientless zero trust configuration will not be supported in either ASDM or FDM.

## Rest API Support

The ZTNA policy can be configured through the rest API with the help of API’s listed under policy section.

**GET​/api​/fmc\_config​/v1​/domain​/{domainUUID}​/policy​/zerotrustpolicies​/{objectId}**

**PUT​/api​/fmc\_config​/v1​/domain​/{domainUUID}​/policy​/zerotrustpolicies/{objectId}**

**DELETE​/api​/fmc\_config​/v1​/domain​/{domainUUID}​/policy​/zerotrustpolicies/{objectId}**

**GET​/api​/fmc\_config​/v1​/domain​/{domainUUID}​/policy​/zerotrustpolicies**

**POST​/api​/fmc\_config​/v1​/domain​/{domainUUID}​/policy​/zerotrustpolicies**

**GET​/api​/fmc\_config​/v1​/domain​/{domainUUID}​/policy​/zerotrustpolicies/applications​/{objectId}**

**PUT​api​/fmc\_config​/v1​/domain​/{domainUUID}​/policy​/zerotrustpolicies​/applications​/{objectId}**

**DELETE​/api​/fmc\_config​/v1​/domain​/{domainUUID}​/policy​/zerotrustpolicies/applications​/{objectId}**

**GET​/api​/fmc\_config​/v1​/domain​/{domainUUID}​/policy​/zerotrustpolicies/applications**

**POST​/api​/fmc\_config​/v1​/domain​/{domainUUID}​/policy​/zerotrustpolicies /applications**

**GET​/api​/fmc\_config​/v1​/domain​/{domainUUID}​/policy​/zerotrustpolicies /applicationgroups​/{objectId}**

**PUT​/api​/fmc\_config​/v1​/domain​/{domainUUID}​/policy​/zerotrustpolicies/applicationgroups​/{objectId}**

**DELETE​/api​/fmc\_config​/v1​/domain​/{domainUUID}​/policy​/zerotrustpolicies/applicationgroups​/{objectId}**

**GET​/api​/fmc\_config​/v1​/domain​/{domainUUID}​/policy​/zerotrustpolicies/applicationgroups**

**POST​/api​/fmc\_config​/v1​/domain​/{domainUUID}​/policy​/zerotrustpolicies​/applicationgroups**

There is API available for GET,POST,PUT & DELETE for the policy, application & application groups with various settings & filters.

# Caveats

* Only IPv4 applications are supported for Zero trust protection. IPv6 applications are not supported.
* Zero-trust Applications with strict HTTP Host Header check do not work with NAT Port.
* The maximum number of users can access the zero-trust protected application per client IP is 100.
* Application exposed within another zero-trust protected application will not be supported unless both the applications have the same IdP and are part of the same application group.
* The application origin server may receive the initial TLS handshake packets from an unauthenticated user behind a NAT device, if another user behind the same (same source IP) has an Authenticated session because for FTD both request (authenticated and unauthenticated) comes from the same Source IP address (NAT device). However, once snort validates the HTTP cookies, unauthenticated connections will be blocked, and actual HTTP traffic will not be allowed to reach the application origin server.
* Interface PAT is not supported for the SRC NAT feature for ZTNA clientless.
* The following modification to unsupported types will not be prevented in FTD. These validations will be done on FMC and any unsupported configs will be rejected.

> Addition of new obj/objgrp to nested objgrp when parent objgrp is in use by ZTNA.  
> Modification of objects configured under objgrp/nested objgrp when parent objgrp is in use by ZTNA.

# Testing Considerations

Clientless zero trust functionality testing considerations include:

[TBD]

# 10 Troubleshooting

If the configuration is proper and has been deployed to the device successfully but zero trust authentication is not working as expected, below are the CLIs and debugs which can be used to troubleshoot.

## 10.1 Troubleshooting on FTD

* "show snort counters" will have a zero\_trust section which shows errors related to cookie validation. (Cookie mismatch/invalid/empty)
* “show snort statistics” has been enhanced with the additional counter for the number of zero-trust cookie messages sent to Snort for validation and username messages sent to Snort for eventing.
* Connection Based Debugging (CBD) infra has been enhanced to emit debug logs from both Lina and Snort modules.​
* “show counters protocol zero\_trust” will show errors related to authentication/authorization for zero-trust applications.
* Below debugs can be used to obtain logs to troubleshoot:
  + debug webvpn request
  + debug webvpn response
  + debug webvpn saml
  + debug zero-trust
* To differentiate among different sessions and among webvpn and zero-trust feature, a unique authentication ID, an 8-char long random hex string, will be generated and used for each authentication session, which will help to tie together multiple requests sent for an application access authentication.
* Debugs will be prepended with "zero-trust:" to identify the debugs emitted by zero-trust feature/module.

## 10.2 ZTNA Diagnostics

**> system generate-troubleshoot**

One or more subset options required.  Displaying list of options:

ALL - Run ALL Of The Following Options

SNT - Snort Performance and Configuration (SNT is not to be used unless approved by Snort team)

PER - Hardware Performance and Logs

SYS - System Configuration, Policy, and Logs

DES - Detection Configuration, Policy, and Logs

NET - Interface and Network Related Data

VDB - Discovery, Awareness, VDB Data, and Logs

UPG - Upgrade Data and Logs

DBO - All Database Data

LOG - All Log Data

NMP - Network Map Information

DMT - Deployment Logs

HA - HA Troubleshoot Logs

CLS - Cluster Troubleshoot Logs

**ZT - Zero-Trust Diagnostics Logs**

> **system generate-troubleshoot ZT**

The troubleshoot option code specified is ZT.

Starting /usr/local/sf/bin/sf\_troubleshoot.pl...

Please, be patient. This may take several minutes.

getting filenames from [/ngfw/usr/local/sf/etc/db\_updates/index]

getting filenames from [/ngfw/usr/local/sf/etc/db\_updates/base-7.4.1]

\*\*\*\*\*\*\*\*\*\*\*\* Applying dynamic update files \*\*\*\*\*\*\*\*\*\*\*\*

Dynamic update files directory: /usr/local/sf/etc/dynamic\_db\_updates

Applying file rule-comments.yaml.

               Status: Success.

\*\*\*\*\*\*\*\*\*\*\*\* Applying dynamic update files finished \*\*\*\*\*\*\*\*\*\*\*\*

Script will collect show tech & find device status with its logs

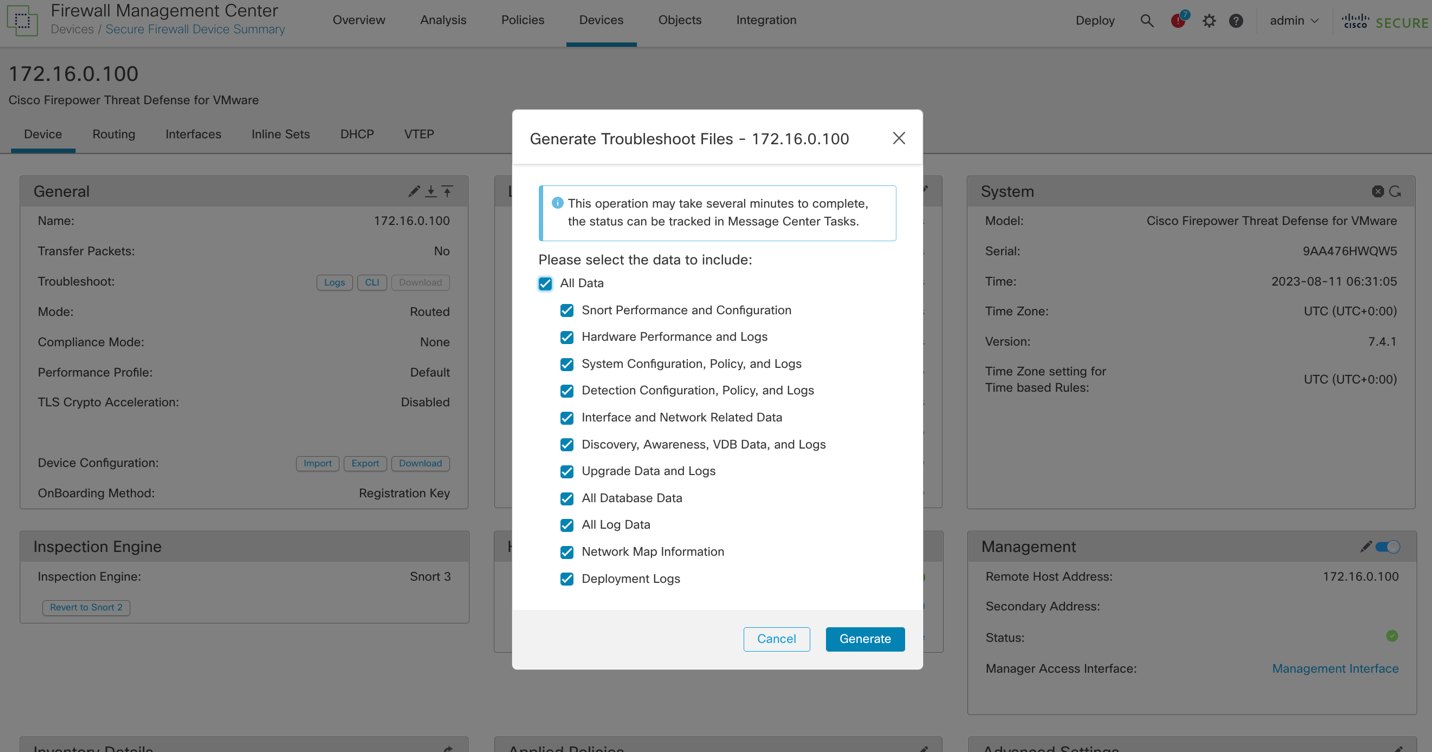
Device Status:- STANDALONE

SSL HW Acceleration:- Enabled

Please wait for few minutes as show tech logs are collected

/ngfw/var/common/results-08-11-2023--50131/command-outputs/usr-bin-perl -MFlyLoader -e "print SF::ZTNA::Diagnostics::collect\_logs()".output

 Will have the TS logs collected in the FTD.



## 10.3 SRC NAT troubleshooting

* "show running-config zero-trust" will display the new field "mapped-source-v4" with obj/objgrp name configured for src nat.
* "show nat detail" will have an implicit SRC NAT dynamic rule with a pat-pool object installed.\*\*
* "show running-config object/object-group" will display the object content and type in detail from the CLI.
* "show xlate" will display all the dynamic xlates created by ZTNA implicit NAT rules.
* "show nat pool ip <ip\_address>" will display the number of addresses allocated/in use in the pat pool.
* "show asp drop". A new drop counter "ztna-src-translation-failure" will be incremented when the pat pool configured for a ZTNA app exhausts, and the app access fails due to pat pool address allocation failures. \*\*
* On cluster setup "show nat pool cluster ip <ip\_address>" will display the current assignment of a PAT address to the control and backup nodes.

\*\* Any issues with SRC NAT rules and app access failures due to pool exhaustion, can be detected using ZTNA Trouble Shooting/Diagnostics tool.

# 11 Telemetry

## 11.1 FTD Telemetry

### 11.1.1 LINA Telemetry

Below are the stats populated by the LINA module for the zero-trust flow and are exposed through telemetry in FMC.

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Metric Collected** | **Documentation** |
| 1 | applicationsActiveAvg | Average number of zero-trust applications active. Time period is 24 hours. |
| 2 | applicationsActiveMax | Max number of zero-trust applications active. Time period is 24 hours. |
| 3 | applicationsEnabledAvg | Average number of zero-trust applications enabled. Time period is 24 hours. |
| 4 | applicationsEnabledMax | Max number of zero-trust applications enabled. Time period is 24 hours. |
| 5 | applicationsTotal | Total number of zero-trust applications. |
| 6 | authenticationLatencyAvg | Average authentication latency. Average latency is calculated as a cumulative average. |
| 7 | authenticationLatencyMax | Max authentication latency. |
| 8 | authenticationLatencyMin | Min authentication latency. |
| 9 | authenticationsInProgressAvg | Average number of zero-trust authentications in progress. Time period is 24 hours. |
| 10 | authenticationsInProgressMax | Max number of zero-trust authentications in progress. Time period is 24 hours. |
| 11 | authenticationsTotal | Total number of zero-trust authentications. |
| 12 | samlRequestsFailed | Number of failed saml authentication request sent by zero-trust enabled application. |
| 13 | samlRequestsPassed | Number of passed saml authentication request sent by zero-trust enabled application. |
| 14 | samlRequestsTotal | Total number of saml authentication request sent by zero-trust enabled application. |
| 15 | samlResponsesFailed | Number of failed saml authentication responses sent by zero-trust enabled application. |
| 16 | samlResponsesPassed | Number of passed saml authentication responses sent by zero-trust enabled application. |
| 17 | samlResponsesTotal | Total number of saml authentication responses sent by zero-trust enabled application. |
| 18 | totalBytesIn | Total bytes in through-the-box post authentication. |
| 19 | totalBytesOut | Total bytes out through-the-box post authentication |
| 20 | usersActiveAvg | Average number of users with valid cookie. Time period is 24 hours. |
| 21 | usersActiveMax | Max number of users with valid cookie. Time period is 24 hours. |
| 22 | usersTotal | Total number of users with valid cookie. |
| 23 | zeroTrustSessionsFailed | Number of failed zero-trust sessions. |
| 24 | zeroTrustSessionsPassed | Number of passed zero-trust sessions. |
| 25 | zeroTrustSessionsTotal | Total number of zero-trust sessions. |

### 11.1.2 Snort Telemetry

Below are the stats populated by snort module for the zero-trust and are exposed through telemetry in FMC.

**Zero-trust snort module counters:**

| **Counter** | **Description** |
| --- | --- |
| total\_flows | Total number of zero-trust flows received. This is Sum of allowed\_flows and blocked\_flows. |
| allowed\_flows | Number of zero\_trust flows successfully authorised |
| blocked\_flows | Total number of zero\_trust flows blocked. This is sum of invalid\_app\_failures, processing\_failures and authorisation\_failures |
| invalid\_app\_failures | Number of zero-trust flows with service not set |
| processing\_failures | Total number of flows blocked due to processing failures |
| non\_tls\_flows | Number of non TLS flows blocked |
| non\_decryptable\_flows | Number of zero-trust flows blocked due to ssl dnd events |
| decrypt\_block\_flows | Number of zero-trust flows blocked due to ssl block events |
| invalid\_state | No zero-trust session state in the flow |
| authorization\_failures | Number of zero-trust flows blocked due to authorization failures |
| domain\_invalid | Number of zero-trust flows blocked due to invalid Domain |
| token\_empty | Authorization token not found |
| token\_too\_long | Authorization token length greater than acceptable value |
| token\_invalid | Authorization token is not a valid authenticated value |
| http\_events | Number of zero-trust http events received |
| uri\_too\_long\_events | Host URI length greater than acceptable value |
| redirect\_error\_events | Total number of flows blocked due to redirect failures |
| reload\_events | Number of zero-trust reload events |
| total\_messages | Total number of zero-trust messages received. This is sum of invalid\_flow\_messages, invalid\_data\_messages, token\_messages and username\_messages. |
| invalid\_flow\_messages | Total number of zero-trust messages not having flow |
| invalid\_data\_messages | Total number of zero-trust messages not having tlv |
| token\_messages | Number of zero-trust messages with cookie tlv |
| username\_messages | Number of zero-trust messages with username tlv |
| unsupported\_messages | Number of zero-trust messages with unsupported tlv |

Snapshot of SSEdata.json file from FMC telemetry.

|  |
| --- |
| "zerotrustStatistics" : {    "allowed\_flows" : 0,    "authorization\_failures" : 0,    "blocked\_flows" : 0,    "decrypt\_block\_flows" : 0,    "domain\_invalid" : 0,    "http\_events" : 0,    "invalid\_app\_failures" : 0,    "invalid\_data\_messages" : 0,    "invalid\_flow\_messages" : 0,    "invalid\_service" : 0,    "invalid\_state" : 0,    "non\_decryptable\_flows" : 0,    "non\_tls\_flows" : 0,    "processing\_failures" : 0,    "redirect\_error\_events" : 0,    "reload\_events" : 0,    "token\_empty" : 0,    "token\_invalid" : 0,    "token\_too\_long" : 0,    "total\_flows" : 1,    "total\_messages" : 0,    "unsupported\_messages" : 0,    "uri\_too\_long\_events" : 0,    "username\_messages" : **0**  } |

# 12 Initiative, Legal, & Regulatory

None.

# 13 Requirements Traceability Considerations

See Target Process PO Feature #724028 (<https://targetprocess.cisco.com/entity/724028-ztna-platform>)

# 14 SReferences

* Target Process Feature “[#735226](https://targetprocess.cisco.com/entity/735226-native-ztna-support-on-ftd-platforms) Native ZTNA support on FTD platforms - Clientless”
  + <*Name of document in italics*>, <document info in plain text>

# 15 Glossary

The following list describes acronyms and definitions for terms used throughout this document:

* + **ZTNA**: Zero Trust Network Access
  + **Term 2 <in bold>**: **<**definition in plain text**>**

# 16 Attachments

As appropriate, attach log sheets, diagrams, schematics, usability research, examples of forms, or other pieces of information used in or generated in the production of the document.

## 16.1 Review Action Items

Use this section to log meeting minutes from the review of this document and to track review action items to closure. Relevant data includes meeting attendees, issues, and action items. Action item data includes description and owner, status (Open or Closed), and closure date.

In lieu of keeping the action item log here, this section may reference external review records, which capture and track the action items to closure. Examples of these external review records include Review Minutes checked into EDCS and review data captured via Peer Review Request Queue Tool:   
<http://wwwin-tools.cisco.com/prrq/welcome.do>

<<ISO requirement>>

<Body>

For guidelines about creating documents using this template, see the [Template Instructions](http://wwwin-eng.cisco.com/cgi-bin/edcs/edcs_info?EDCS-700169). For template comments, contact the [[cpdm-input](mailto:cpdm-input@cisco.com)](mailto:cpdm-input@cisco.com) alias.

End of Document