**APIs:**

CONFIG Classes:

CiscoConfigFW:

Class parse cisco fw and return handle to each config item.

Methods:

* Initiate\_Config() : call methods to parse

ACL\_Name()

Network\_Name\_To\_IP()

Network\_Objects()

Service\_Objects()

Network\_Object\_Group()

Service\_Object\_Group()

Protocol\_Object\_Group()

* RoutingInfo() : Parse routing info , return list of routing object of type “RoutingConfig”.
* ACL\_Name() : Parse ACL access group name , return dict of access group name as key and object to the config line type “AccessGroupConfig” as value.
* Network\_Name\_To\_IP() :
* Network\_Objects()
* Service\_Objects()
* Network\_Object\_Group()
* Service\_Object\_Group()
* Protocol\_Object\_Group()
* ICMP\_TYPE\_Object\_Group()
* ACLs()

Instance Variables:

* FWName : FW Name <string>
* ObjName : Name of needed object to return <string>
* RoutingEntries : List contain routing objects of type ” RoutingConfig” <list>
* Access\_Group\_Dict : Dict {key:value} ,key = interface name , value = object of type “AccessGroupConfig”.
* Network\_Name\_IP\_Dict: Dict , key=Network name, value= object of type “NetworkName”
* Network\_Object\_Dict :
* Service\_Object\_Dict:
* Network\_Object\_Group\_Dict:
* Service\_Object\_Group\_Dict:
* Protocol\_Object\_Group\_Dict:
* ICMP\_TYPE\_Object\_Group\_Dict:
* ACLs\_Object\_List:

RoutingConfig:

Instance Variables:

* DIRECT\_CONNECTED : Boolean Value indicate if dst network is directly connected or not.
* DEFAULT\_ROUTE : Boolean Value indicate if route is the default or not.
* DestNetwork : <string> network id of the destination network.
* AD: <string> administrative distance of the route.
* Metric:<string> metric value of the route.
* NextHop: <string> network id of the next hope device.
* ExitInt : <string> name of the exit interface.
* RouteList: <list> of dictionaries contain all exist routes for certain network if many exist.

{'AD':AD,'Metric':Metric,'NextHop':NextHop,'ExitInt':ExitInt}

NetworkObject:

Instance Variables:

* Name: <string> object name
* Type: <string> subnet/host/range
* Network:<IPNetwork> in case of subnet/host and <IPRange> in case of range type

ServiceObject:

Instance Variables:

* Name: <string> object name
* Protocol: <string> protocol name
* ServiceGroup: <IP> for ip protocols or <ICMP> for icmp type, or < TCPUDP>

NetworkObjectGroup:

Instance Variables:

* Name: obj name
* Network: <list> contain <IPNetwork> / < NetworkObjectGroup> / < NetworkObject >

ProtocolObjectGroup:

Instance Variables:

* Name:Object name
* Protocol:<list> contain <string> protocol name

ServiceObjectGroup:

Instance Variables:

* Name: Object name
* Type: <sting> -> port/ service
* Protocol: <string>
* Service: <list>
  + if type == port :
    - Contain < CiscoPort >
  + If type == service:
    - Contain <TCPUDP>/< ICMP >/<IP>

ICMPTypeObjectGroup:

Instance Variables:

* Name:Object name
* Protocol: ‘icmp’
* Service: <list> contain <ICMP>/< ICMPTypeObjectGroup >

ACLs:

Instance Variables:

* Name: <string> acl name
* Protocol: <string> / < ServiceObject > / < ServiceObjectGroup > / < ProtocolObjectGroup >
* Type: <string> ‘standard’ or ‘extended’
* Action: <string> “permit” or “deny”
* SrcNet: <IPNetwork> / < NetworkObject > / < NetworkObjectGroup >
* SrcPort: < CiscoPort > / < ServiceObjectGroup > only defined if Protocol = tcp/udp/tcp-udp
* DstNet: <IPNetwork> / < NetworkObject > / < NetworkObjectGroup >
* DstPort: < CiscoPort > / < ServiceObjectGroup > only defined if Protocol = tcp/udp/tcp-udp
* TimeRange: <string> time-range name
* LineNo: <string> line number
* DstICMPType:<string> type code

FortiConfigUTM: