Chapter 1. Components of the Katzenpost mixnet

Table of Contents

Directory authorities

Configuring directory authorities

The following configuration is drawn from the reference implementation in `katzenpost/docker/voting_mixnet/auth1/authority.toml`. In a real-world mixnet, the component hosts would not be sharing a single IP address. For more information about the test mixnet, see Using the Katzenpost test network.

Server section

```
[Server]
    Identifier = "auth1"
    WireKEMScheme = "xwing"
    PKISignatureScheme = "Ed25519"
    Addresses = ["127.0.0.1:30001"]
    DataDir = "/voting_mixnet/auth1"
```

- **Identifier**

  Identifier is the human readable identifier for the node (eg: FQDN).

  Type: string

- **WireKEMScheme**

  WireKEMScheme is the wire protocol KEM scheme to use.

  Type: string

- **PKISignatureScheme**

  PKISignatureScheme specifies the cryptographic signature scheme

  Type: string

- **Addresses**

  // Addresses are the IP address/port combinations that the server will bind

  // to for incoming connections.

  Type: []string

- **DataDir**

  DataDir is the absolute path to the server's state files.

  Type: string

Authorities section

An Authorities section is configured for each peer authority.

```
[[Authorities]]
    Identifier = "auth1"
    IdentityPublicKey = "-----BEGIN ED25519 PUBLIC KEY-----\n/v3qYgh2TvV5ZqEVgwcjJ
    PKISignatureScheme = "Ed25519"
    LinkPublicKey = "-----BEGIN XWING PUBLIC KEY-----\nJeFaZoYQEOO71zPFFWjL7DyDp4g
    WireKEMScheme = "xwing"
    Addresses = ["127.0.0.1:30001"]

[[Authorities]]
    Identifier = "auth2"
    IdentityPublicKey = "-----BEGIN ED25519 PUBLIC KEY-----\n60KQRhG7njt+kLQuwWlfR
    PKISignatureScheme = "Ed25519"
    LinkPublicKey = "-----BEGIN XWING PUBLIC KEY-----\nHVR2m7i6G6cf1qxUvyEr3KC7JvA
    WireKEMScheme = "xwing"
    Addresses = ["127.0.0.1:30002"]

[[Authorities]]
    Identifier = "auth3"
    IdentityPublicKey = "-----BEGIN ED25519 PUBLIC KEY-----\naZUXqznyLO2mKDceIDs0o
    PKISignatureScheme = "Ed25519"
    LinkPublicKey = "-----BEGIN XWING PUBLIC KEY-----\nEZukXtZwHTjGj7tCI0kmUcq0QEt
    WireKEMScheme = "xwing"
    Addresses = ["127.0.0.1:30003"]
```

- **Identifier**

  Human readable identifier for the node (eg: FQDN)

  Type: string

- **IdentityPublicKey**

  String in PEM format containing the public identity key key.

  Type: sign.PublicKey

- **PKISignatureScheme**

PKISignatureScheme specifies the cryptographic signature scheme

Type: string

- **LinkPublicKey**

  LinkPublicKeyPem is string containing the PEM format of the peer's public link layer key.

  Type: kem.PublicKey

- **WireKEMScheme**

  WireKEMScheme is the wire protocol KEM scheme to use.

  Type: string

- **Addresses**

  One or more IP addresses that correspond to local network interfaces to listen for connections on. These can be specified as IPv4 or IPv6 addresses.

  // Addresses are the IP address/port combinations that the peer authority

  // uses for the Directory Authority service.

  Type: []string

Logging section

The logging configuration section controls logging.

```
[Logging]
    Disable = false
    File = "katzenpost.log"
    Level = "INFO"
```

- **Disable**

  Disables logging if set to **true**.

  Type: bool

- **File**

  Specifies the log file. If omitted, stdout is used.

  Type: string

- **Level**

  Supported values are ERROR | WARNING | NOTICE |INFO | DEBUG.

  Type: string

Warning

The DEBUG log level is unsafe for production use.

Parameters section

The Parameters section holds the network parameters, for example:

```
[Parameters]
    SendRatePerMinute = 0
    Mu = 0.005
    MuMaxDelay = 1000
    LambdaP = 0.001
    LambdaPMaxDelay = 1000
    LambdaL = 0.0005
    LambdaLMaxDelay = 1000
    LambdaD = 0.0005
    LambdaDMaxDelay = 3000
    LambdaM = 0.0005
    LambdaG = 0.0
    LambdaMMaxDelay = 100
    LambdaGMaxDelay = 100
```

- **SendRatePerMinute**

  is the rate limiter maximum allowed rate of packets per client.

  SendRatePerMinute is the rate per minute.

  Type: uint64

- **Mu**

  is the inverse of the mean of the exponential distribution that the Sphinx packet per-hop mixing delay will be sampled from.

  // Mu is the inverse of the mean of the exponential distribution

  // that is used to select the delay for each hop.

  Type: float64

- **MuMaxDelay**

  is the maximum Sphinx packet per-hop mixing delay in milliseconds.

  MuMaxDelay sets the maximum delay for Mu

  Type: uint64

- **LambdaP**

  LambdaP is the inverse of the mean of the exponential distribution that clients will sample to determine the time interval between sending messages from it's FIFO egress queue or drop decoy messages if the queue is empty.

  // LambdaP is the inverse of the mean of the exponential distribution

  // that is used to select the delay between clients sending from their egress

  // FIFO queue or drop decoy message.

Type: float64

- **LambdaPMaxDelay**

  is the maximum send interval for LambdaP in milliseconds

  LambdaPMaxDelay sets the maximum delay for LambdaP.

  Type: uint64

- **LambdaL**

  LambdaL is the inverse of the mean of the exponential distribution that is used to select the delay between clients sending loop decoys.

  Type: float64

- **LambdaLMaxDelay**

  sets the maximum send interval for LambdaL in milliseconds.

  LambdaLMaxDelay sets the maximum delay for LambdaP.

  Type: uint64

- **LambdaD**

  is the inverse of the mean of the exponential distribution that clients will sample to determine the time interval between sending decoy drop messages.

  // LambdaD is the inverse of the mean of the exponential distribution

  // that is used to select the delay between clients sending deop decoys.

  Type: float64

- **LambdaDMaxDelay**

  is the maximum send interval in milliseconds.

  LambdaDMaxDelay sets the maximum delay for LambdaP.

  Type: uint64

- **LambdaM**

  is the inverse of the mean of the exponential distribution that mixes will sample to determine send timing of mix loop decoy traffic.

  // LambdaM is the inverse of the mean of the exponential distribution

  // that is used to select the delay between sending mix node decoys.

  Type: float64

- **LambdaG**

  // LambdaG is the inverse of the mean of the exponential distribution

// that is used to select the delay between sending gateway node decoys.

//

// WARNING: This is not used via the TOML config file; this field is only

// used internally by the dirauth server state machine.

Type: float64

- **LambdaMMaxDelay**

  sets the maximum delay for LambdaM

  LambdaMMaxDelay sets the maximum delay for LambdaP.

  Type: uint64

- **LambdaGMaxDelay**

  LambdaGMaxDelay sets the maximum delay for LambdaG.

  Type: uint64

Debug section

```
[Debug]
    Layers = 3
    MinNodesPerLayer = 1
    GenerateOnly = false
```

- **Layers**

  Number of non-provider layers in the network topology.

  Type: int

- **MinNodesrPerLayer**

  Minimum number of nodes per layer required to form a valid document.

  Type: int

- **GenerateOnly**

  If set to true, the server halts and cleans up the data directory immediately after long-term key generation.

  Type: bool

Mixes sections

The Mixes configuration section lists mix nodes that are known to the authority.

```
[[Mixes]]
    Identifier = "mix1"
    IdentityPublicKeyPem = "../mix1/identity.public.pem"

[[Mixes]]
    Identifier = "mix2"
    IdentityPublicKeyPem = "../mix2/identity.public.pem"

[[Mixes]]
    Identifier = "mix3"
    IdentityPublicKeyPem = "../mix3/identity.public.pem"
```

- **Identifier**

  A human readable mix node identifier.

  Type: string

- **IdentityPublicKeyPem**

  Path and file name of a mix node's public EdDSA signing key, also known as the identity key, in Base16 or Base64 format.

  Type: string

GatewayNodes sections

The GatewayNodes configuration section lists gateway nodes that are known to the authority.

```
[[GatewayNodes]]
    Identifier = "gateway1"
    IdentityPublicKeyPem = "../gateway1/identity.public.pem"
```

- **Identifier**

  A human readable gateway node identifier.

  Type: string

- **IdentityPublicKeyPem**

  Path and file name of a gateway node's public EdDSA signing key, also known as the identity key, in Base16 or Base64 format.

  Type: string

ServiceNodes sections

The ServiceNodes configuration section lists service nodes that are known to the authority.

```
[[ServiceNodes]]
    Identifier = "servicenode1"
    IdentityPublicKeyPem = "../servicenode1/identity.public.pem"
```

- **Identifier**

  A human readable service node identifier.

  Type: string

- **IdentityPublicKeyPem**

  Path and file name of a service node's public EdDSA signing key, also known as the identity key, in Base16 or Base64 format.

  Type: string

Topology section

The Topology configuration section defines the layers of the mix network and the mix nodes in each layer.

```
[Topology]

    [[Topology.Layers]]

        [[Topology.Layers.Nodes]]
            Identifier = "mix1"
            IdentityPublicKeyPem = "../mix1/identity.public.pem"

    [[Topology.Layers]]

        [[Topology.Layers.Nodes]]
            Identifier = "mix2"
            IdentityPublicKeyPem = "../mix2/identity.public.pem"

    [[Topology.Layers]]

        [[Topology.Layers.Nodes]]
            Identifier = "mix3"
            IdentityPublicKeyPem = "../mix3/identity.public.pem"
```

- **Identifier**

  A human readable mix node identifier.

  Type: string

- **IdentityPublicKeyPem**

  Path and file name of a mix node's public EdDSA signing key, also known as the identity key, in Base16 or Base64 format.

  Type: string

SphinxGeometry section

```
[SphinxGeometry]
    PacketLength = 3082
    NrHops = 5
    HeaderLength = 476
    RoutingInfoLength = 410
    PerHopRoutingInfoLength = 82
    SURBLength = 572
    SphinxPlaintextHeaderLength = 2
    PayloadTagLength = 32
```

```
ForwardPayloadLength = 2574
UserForwardPayloadLength = 2000
NextNodeHopLength = 65
SPRPKeyMaterialLength = 64
NIKEName = "x25519"
KEMName = ""
```

- **PacketLength**

  PacketLength is the length of a packet.

  Type: int

- **NrHops**

  // NrHops is the number of hops, this indicates the size

  // of the Sphinx packet header.

  Type: int

- **HeaderLength**

  HeaderLength is the length of the Sphinx packet header in bytes.

  Type: int

- **RoutingInfoLength**

  RoutingInfoLength is the length of the routing info portion of the header.

  Type: int

- **PerHopRoutingInfoLength**

  PerHopRoutingInfoLength is the length of the per hop routing info.

  Type: int

- **SURBLength**

  SURBLength is the length of SURB.

  Type: int

- **SphinxPlaintextHeaderLength**

  SphinxPlaintextHeaderLength is the length of the plaintext header.

  Type: int

- **PayloadTagLength**

  PayloadTagLength is the length of the payload tag.

  Type: int

- **ForwardPayloadLength**

ForwardPayloadLength is the size of the payload.

Type: int

- **UserForwardPayloadLength**

  the size of the usable payload.

  Type: int

- **NextNodeHopLength**

  // NextNodeHopLength is derived off the largest routing info

  // block that we expect to encounter. Everything else just has a

  // NextNodeHop + NodeDelay, or a Recipient, both cases which are

  // shorter.

  Type: int

- **SPRPKeyMaterialLength**

  SPRPKeyMaterialLength is the length of the SPRP key.

  Type: int

- **NIKEName**

  // NIKEName is the name of the NIKE scheme used by the mixnet's Sphinx packet.

  // NIKEName and KEMName are mutually exclusive.

  Type: string

- **KEMName**

  KEMName is the name of the KEM scheme used by the mixnet's Sphinx packet. NIKEName and KEMName are mutually exclusive.

  Type: string

Mix, gateway, and service nodes

Configuring mix nodes

The following configuration is drawn from the reference implementation in `katzenpost/docker/voting_mixnet/mix1/katzenpost.toml`. In a real-world mixnet, the component hosts would not be sharing a single IP address. For more information about the test mixnet, see Using the Katzenpost test network.

Server section

```
[Server]
  Identifier = "mix1"
  WireKEM = "xwing"
  PKISignatureScheme = "Ed25519"
  Addresses = ["127.0.0.1:30008"]
  OnlyAdvertiseAltAddresses = false
  MetricsAddress = "127.0.0.1:30009"
  DataDir = "/voting_mixnet/mix1"
  IsGatewayNode = false
  IsServiceNode = false
  [Server.AltAddresses]
```

- **Identifier**

  Identifier is the human readable identifier for the node (eg: FQDN).

  Type: string

- **WireKEM**

  // WireKEM is the KEM string representing the chosen KEM scheme with which to communicate

  // with the mixnet and dirauth nodes.

  Type: string

- **PKISignatureScheme**

  PKISignatureScheme specifies the cryptographic signature scheme

  Type: string

- **Addresses**

  // Addresses are the IP address/port combinations that the server will bind

  // to for incoming connections.

  Type: []string

- **OnlyAdvertiseAltAddresses**

  // If set to true then only advertise to the PKI the AltAddresses

  // and do NOT send any of the Addresses.

  Type: bool

- **MetricsAddress**

  MetricsAddress is the address/port to bind the prometheus metrics endpoint to.

  Type: string

- **DataDir**

  DataDir is the absolute path to the server's state files.

Type: string

- **IsGatewayNode**

  IsGatewayNode specifies if the server is a gateway or not.

  Type: bool

- **IsServiceNode**

  IsServiceNode specifies if the server is a service node or not.

  Type: bool

- **[Server.AltAddresses]**

  Map of additional transport protocols and addresses at which the node is reachable by clients, in the form

  ```
  [Server.AltAddresses]
      TCP = ["localhost:30004"]
  ```

  Type: []string

Logging section

The logging configuration section controls logging.

```
[Logging]
    Disable = false
    File = "katzenpost.log"
    Level = "INFO"
```

- **Disable**

  Disables logging if set to **true**.

  Type: bool

- **File**

  Specifies the log file. If omitted, stdout is used.

  Type: string

- **Level**

  Supported values are ERROR | WARNING | NOTICE |INFO | DEBUG.

  Type: string

Warning

The DEBUG log level is unsafe for production use.

PKI section

The PKI section contains the directory authority configuration for a mix, gateway, or service node.

```
[PKI]
    [PKI.Voting]

        [[PKI.Voting.Authorities]]
            Identifier = "auth1"
            IdentityPublicKey = "-----BEGIN ED25519 PUBLIC KEY-----\n/v3qYgh2TvV5Z
            PKISignatureScheme = "Ed25519"
            LinkPublicKey = "-----BEGIN XWING PUBLIC KEY-----\nJeFaZoYQEOO71zPFFWj
            WireKEMScheme = "xwing"
            Addresses = ["127.0.0.1:30001"]

        [[PKI.Voting.Authorities]]
            Identifier = "auth2"
            IdentityPublicKey = "-----BEGIN ED25519 PUBLIC KEY-----\n60KQRhG7njt+k
            PKISignatureScheme = "Ed25519"
            LinkPublicKey = "-----BEGIN XWING PUBLIC KEY-----\nHVR2m7i6G6cf1qxUvyE
            WireKEMScheme = "xwing"
            Addresses = ["127.0.0.1:30002"]

        [[PKI.Voting.Authorities]]
            Identifier = "auth3"
            IdentityPublicKey = "-----BEGIN ED25519 PUBLIC KEY-----\naZUXqznyLO2mK
            PKISignatureScheme = "Ed25519"
            LinkPublicKey = "-----BEGIN XWING PUBLIC KEY-----\nEZukXtZwHTjGj7tCI0k
            WireKEMScheme = "xwing"
            Addresses = ["127.0.0.1:30003"]
```

- **Identifier**

  Identifier is the human readable identifier for the node (eg: FQDN).

  Type: string

- **IdentityPublicKey**

  // IdentityPublicKeyPem is a string in PEM format containing

  // the public identity key key.

  Type: string

- **PKISignatureScheme**

  PKISignatureScheme specifies the cryptographic signature scheme

  Type: string

- **LinkPublicKey**

  LinkPublicKeyPem is string containing the PEM format of the peer's public link layer key.

  Type: string

- **WireKEMScheme**

  WireKEMScheme is the wire protocol KEM scheme to use.

Type: string

- **Addresses**

  // Addresses are the IP address/port combinations that the peer authority

  // uses for the Directory Authority service.

  Type: []string

Management section

Management is the Katzenpost management interface configuration. The management section specifies connectivity information for the Katzenpost control protocol which can be used to make configuration changes during run-time. An example configuration looks like this:

```
[Management]
    Enable = false
    Path = "/voting_mixnet/mix1/management_sock"
```

- **Enable**

  Enables the management interface if set to true.

  Type: bool

- **Path**

  Specifies the path to the management interface socket. If left empty, then management_sock will be used under the DataDir.

  Type: string

SphinxGeometry section

```
[SphinxGeometry]
    PacketLength = 3082
    NrHops = 5
    HeaderLength = 476
    RoutingInfoLength = 410
    PerHopRoutingInfoLength = 82
    SURBLength = 572
    SphinxPlaintextHeaderLength = 2
    PayloadTagLength = 32
    ForwardPayloadLength = 2574
    UserForwardPayloadLength = 2000
    NextNodeHopLength = 65
    SPRPKeyMaterialLength = 64
    NIKEName = "x25519"
    KEMName = ""
```

- **PacketLength**

  PacketLength is the length of a packet.

Type: int

- **NrHops**

  // NrHops is the number of hops, this indicates the size

  // of the Sphinx packet header.

  Type: int

- **HeaderLength**

  HeaderLength is the length of the Sphinx packet header in bytes.

  Type: int

- **RoutingInfoLength**

  RoutingInfoLength is the length of the routing info portion of the header.

  Type: int

- **PerHopRoutingInfoLength**

  PerHopRoutingInfoLength is the length of the per hop routing info.

  Type: int

- **SURBLength**

  SURBLength is the length of SURB.

  Type: int

- **SphinxPlaintextHeaderLength**

  SphinxPlaintextHeaderLength is the length of the plaintext header.

  Type: int

- **PayloadTagLength**

  PayloadTagLength is the length of the payload tag.

  Type: int

- **ForwardPayloadLength**

  ForwardPayloadLength is the size of the payload.

  Type: int

- **UserForwardPayloadLength**

  the size of the usable payload.

  Type: int

- **NextNodeHopLength**

  // NextNodeHopLength is derived off the largest routing info

  // block that we expect to encounter. Everything else just has a

  // NextNodeHop + NodeDelay, or a Recipient, both cases which are

  // shorter.

  Type: int

- **SPRPKeyMaterialLength**

  SPRPKeyMaterialLength is the length of the SPRP key.

  Type: int

- **NIKEName**

  // NIKEName is the name of the NIKE scheme used by the mixnet's Sphinx packet.

  // NIKEName and KEMName are mutually exclusive.

  Type: string

- **KEMName**

  KEMName is the name of the KEM scheme used by the mixnet's Sphinx packet. NIKEName and KEMName are mutually exclusive.

  Type: string

Debug section

Debug is the Katzenpost server debug configuration for advanced tuning.

```
[Debug]
                    NumSphinxWorkers = 16
                    NumServiceWorkers = 3
                    NumGatewayWorkers = 3
                    NumKaetzchenWorkers = 3
                    SchedulerExternalMemoryQueue = false
                    SchedulerQueueSize = 0
                    SchedulerMaxBurst = 16
                    UnwrapDelay = 250
                    GatewayDelay = 500
                    ServiceDelay = 500
                    KaetzchenDelay = 750
                    SchedulerSlack = 150
                    SendSlack = 50
                    DecoySlack = 15000
                    ConnectTimeout = 60000
                    HandshakeTimeout = 30000
                    ReauthInterval = 30000
                    SendDecoyTraffic = false
                    DisableRateLimit = false
```

```
GenerateOnly = false
```

- **NumSphinxWorkers**

  specifies the number of worker instances to use for inbound Sphinx packet processing.

  Type: int

- **NumProviderWorkers**

  specifies the number of worker instances to use for provider specific packet processing.

  Type: int

- **NumKaetzchenWorkers**

  specifies the number of worker instances to use for Kaetzchen specific packet processing.

  Type: int

- **SchedulerExternalMemoryQueue**

  will enable the experimental external memory queue that is backed by disk.

  Type: bool

- **SchedulerQueueSize**

  is the maximum allowed scheduler queue size before random entries will start getting dropped. A value <= 0 is treated as unlimited.

  Type: int

- **SchedulerMaxBurst**

  is the maximum number of packets that will be dispatched per scheduler wakeup event.

  Type:

- **UnwrapDelay**

  is the maximum allowed unwrap delay due to queueing in milliseconds.

  Type: int

- **GatewayDelay**

  the maximum allowed gateway node worker delay due to queueing

  in milliseconds.

  Type: int

- **ServiceDelay**

  is the maximum allowed provider delay due to queueing in milliseconds.

  Type: int

- **KaetzchenDelay**

  is the maximum allowed kaetzchen delay due to queueing in milliseconds.

  Type: int

- **SchedulerSlack**

  is the maximum allowed scheduler slack due to queueing and or processing in milliseconds.

  Type: int

- **SendSlack**

  is the maximum allowed send queue slack due to queueing and or congestion in milliseconds.

  Type: int

- **DecoySlack**

  is the maximum allowed decoy sweep slack due to various external delays such as latency before a loop decoy packet will be considered lost.

  Type: int

- **ConnectTimeout**

  specifies the maximum time a connection can take to establish a TCP/IP connection in milliseconds.

  Type: int

- **HandshakeTimeout**

  specifies the maximum time a connection can take for a link protocol handshake in milliseconds.

  Type: int

- **ReauthInterval**

  specifies the interval at which a connection will be reauthenticated in milliseconds.

  Type: int

- **SendDecoyTraffic**

  enables sending decoy traffic. This is still experimental and untuned and thus is disabled by default. WARNING: This option will go away once decoy traffic is more concrete.

  Type: bool

- **DisableRateLimit**

  disables the per-client rate limiter. This option should only be used for testing.

  Type: bool

- **GenerateOnly**

  halts and cleans up the server right after long term key generation.

Type: bool

Configuring gateway nodes

The following configuration is drawn from the reference implementation in `katzenpost/docker/voting_mixnet/gateway1/katzenpost.toml`. In a real-world mixnet, the component hosts would not be sharing a single IP address. For more information about the test mixnet, see Using the Katzenpost test network.

Server section

```
[Server]
    Identifier = "gateway1"
    WireKEM = "xwing"
    PKISignatureScheme = "Ed25519"
    Addresses = ["127.0.0.1:30004"]
    OnlyAdvertiseAltAddresses = false
    MetricsAddress = "127.0.0.1:30005"
    DataDir = "/voting_mixnet/gateway1"
    IsGatewayNode = true
    IsServiceNode = false
    [Server.AltAddresses]
        TCP = ["localhost:30004"]
```

- **Identifier**

  Identifier is the human readable identifier for the node (eg: FQDN).

  Type: string

- **WireKEM**

  // WireKEM is the KEM string representing the chosen KEM scheme with which to communicate

  // with the mixnet and dirauth nodes.

  Type: string

- **PKISignatureScheme**

  PKISignatureScheme specifies the cryptographic signature scheme

  Type: string

- **Addresses**

  // Addresses are the IP address/port combinations that the server will bind

  // to for incoming connections.

  Type: []string

- **OnlyAdvertiseAltAddresses**

  // If set to true then only advertise to the PKI the AltAddresses

// and do NOT send any of the Addresses.

Type: bool

- **MetricsAddress**

  MetricsAddress is the address/port to bind the prometheus metrics endpoint to.

  Type: string

- **DataDir**

  DataDir is the absolute path to the server's state files.

  Type: string

- **IsGatewayNode**

  IsGatewayNode specifies if the server is a gateway or not.

  Type: bool

- **IsServiceNode**

  IsServiceNode specifies if the server is a service node or not.

  Type: bool

- **[Server.AltAddresses]**

  Map of additional transport protocols and addresses at which the node is reachable by clients, in the form

  ```
  [Server.AltAddresses]
      TCP = ["localhost:30004"]
  ```

  Type: []string

Logging section

The logging configuration section controls logging.

```
[Logging]
    Disable = false
    File = "katzenpost.log"
    Level = "INFO"
```

- **Disable**

  Disables logging if set to **true**.

  Type: bool

- **File**

  Specifies the log file. If omitted, stdout is used.

  Type: string

- **Level**

  Supported values are ERROR | WARNING | NOTICE |INFO | DEBUG.

  Type: string

Warning

The DEBUG log level is unsafe for production use.

Gateway section

```
[Gateway]
    [Gateway.UserDB]
        Backend = "bolt"
            [Gateway.UserDB.Bolt]
                UserDB = "/voting_mixnet/gateway1/users.db"
    [Gateway.SpoolDB]
        Backend = "bolt"
            [Gateway.SpoolDB.Bolt]
                SpoolDB = "/voting_mixnet/gateway1/spool.db"
```

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

PKI section

The PKI section contains the directory authority configuration for a mix, gateway, or service node.

```
[PKI]
    [PKI.Voting]
```

```
        [[PKI.Voting.Authorities]]
            Identifier = "auth1"
            IdentityPublicKey = "-----BEGIN ED25519 PUBLIC KEY-----\n/v3qYgh2TvV5Z
            PKISignatureScheme = "Ed25519"
            LinkPublicKey = "-----BEGIN XWING PUBLIC KEY-----\nJeFaZoYQEOO71zPFFWj
            WireKEMScheme = "xwing"
            Addresses = ["127.0.0.1:30001"]

        [[PKI.Voting.Authorities]]
            Identifier = "auth2"
            IdentityPublicKey = "-----BEGIN ED25519 PUBLIC KEY-----\n60KQRhG7njt+k
            PKISignatureScheme = "Ed25519"
            LinkPublicKey = "-----BEGIN XWING PUBLIC KEY-----\nHVR2m7i6G6cf1qxUvyE
            WireKEMScheme = "xwing"
            Addresses = ["127.0.0.1:30002"]

        [[PKI.Voting.Authorities]]
            Identifier = "auth3"
            IdentityPublicKey = "-----BEGIN ED25519 PUBLIC KEY-----\naZUXqznyLO2mK
            PKISignatureScheme = "Ed25519"
            LinkPublicKey = "-----BEGIN XWING PUBLIC KEY-----\nEZukXtZwHTjGj7tCI0k
            WireKEMScheme = "xwing"
            Addresses = ["127.0.0.1:30003"]
```

- **Identifier**

  Identifier is the human readable identifier for the node (eg: FQDN).

  Type: string

- **IdentityPublicKey**

  // IdentityPublicKeyPem is a string in PEM format containing

  // the public identity key key.

  Type: string

- **PKISignatureScheme**

  PKISignatureScheme specifies the cryptographic signature scheme

  Type: string

- **LinkPublicKey**

  LinkPublicKeyPem is string containing the PEM format of the peer's public link layer key.

  Type: string

- **WireKEMScheme**

  WireKEMScheme is the wire protocol KEM scheme to use.

  Type: string

- **Addresses**

// Addresses are the IP address/port combinations that the peer authority

// uses for the Directory Authority service.

Type: []string

Management section

Management is the Katzenpost management interface configuration. The management section specifies connectivity information for the Katzenpost control protocol which can be used to make configuration changes during run-time. An example configuration looks like this:

```
[Management]
    Enable = false
    Path = "/voting_mixnet/mix1/management_sock"
```

- **Enable**

  Enables the management interface if set to true.

  Type: bool

- **Path**

  Specifies the path to the management interface socket. If left empty, then management_sock will be used under the DataDir.

  Type: string

SphinxGeometry section

```
[SphinxGeometry]
    PacketLength = 3082
    NrHops = 5
    HeaderLength = 476
    RoutingInfoLength = 410
    PerHopRoutingInfoLength = 82
    SURBLength = 572
    SphinxPlaintextHeaderLength = 2
    PayloadTagLength = 32
    ForwardPayloadLength = 2574
    UserForwardPayloadLength = 2000
    NextNodeHopLength = 65
    SPRPKeyMaterialLength = 64
    NIKEName = "x25519"
    KEMName = ""
```

- **PacketLength**

  PacketLength is the length of a packet.

  Type: int

- **NrHops**

  // NrHops is the number of hops, this indicates the size

  // of the Sphinx packet header.

  Type: int

- **HeaderLength**

  HeaderLength is the length of the Sphinx packet header in bytes.

  Type: int

- **RoutingInfoLength**

  RoutingInfoLength is the length of the routing info portion of the header.

  Type: int

- **PerHopRoutingInfoLength**

  PerHopRoutingInfoLength is the length of the per hop routing info.

  Type: int

- **SURBLength**

  SURBLength is the length of SURB.

  Type: int

- **SphinxPlaintextHeaderLength**

  SphinxPlaintextHeaderLength is the length of the plaintext header.

  Type: int

- **PayloadTagLength**

  PayloadTagLength is the length of the payload tag.

  Type: int

- **ForwardPayloadLength**

  ForwardPayloadLength is the size of the payload.

  Type: int

- **UserForwardPayloadLength**

  the size of the usable payload.

  Type: int

- **NextNodeHopLength**

  // NextNodeHopLength is derived off the largest routing info

// block that we expect to encounter. Everything else just has a

// NextNodeHop + NodeDelay, or a Recipient, both cases which are

// shorter.

Type: int

- **SPRPKeyMaterialLength**

  SPRPKeyMaterialLength is the length of the SPRP key.

  Type: int

- **NIKEName**

  // NIKEName is the name of the NIKE scheme used by the mixnet's Sphinx packet.

  // NIKEName and KEMName are mutually exclusive.

  Type: string

- **KEMName**

  KEMName is the name of the KEM scheme used by the mixnet's Sphinx packet. NIKEName and KEMName are mutually exclusive.

  Type: string

Debug section

Debug is the Katzenpost server debug configuration for advanced tuning.

```
[Debug]
                    NumSphinxWorkers = 16
                    NumServiceWorkers = 3
                    NumGatewayWorkers = 3
                    NumKaetzchenWorkers = 3
                    SchedulerExternalMemoryQueue = false
                    SchedulerQueueSize = 0
                    SchedulerMaxBurst = 16
                    UnwrapDelay = 250
                    GatewayDelay = 500
                    ServiceDelay = 500
                    KaetzchenDelay = 750
                    SchedulerSlack = 150
                    SendSlack = 50
                    DecoySlack = 15000
                    ConnectTimeout = 60000
                    HandshakeTimeout = 30000
                    ReauthInterval = 30000
                    SendDecoyTraffic = false
                    DisableRateLimit = false
                    GenerateOnly = false
```

- **NumSphinxWorkers**

specifies the number of worker instances to use for inbound Sphinx packet processing.

Type: int

- **NumProviderWorkers**

  specifies the number of worker instances to use for provider specific packet processing.

  Type: int

- **NumKaetzchenWorkers**

  specifies the number of worker instances to use for Kaetzchen specific packet processing.

  Type: int

- **SchedulerExternalMemoryQueue**

  will enable the experimental external memory queue that is backed by disk.

  Type: bool

- **SchedulerQueueSize**

  is the maximum allowed scheduler queue size before random entries will start getting dropped. A value <= 0 is treated as unlimited.

  Type: int

- **SchedulerMaxBurst**

  is the maximum number of packets that will be dispatched per scheduler wakeup event.

  Type:

- **UnwrapDelay**

  is the maximum allowed unwrap delay due to queueing in milliseconds.

  Type: int

- **GatewayDelay**

  the maximum allowed gateway node worker delay due to queueing

  in milliseconds.

  Type: int

- **ServiceDelay**

  is the maximum allowed provider delay due to queueing in milliseconds.

  Type: int

- **KaetzchenDelay**

  is the maximum allowed kaetzchen delay due to queueing in milliseconds.

Type: int

- **SchedulerSlack**

  is the maximum allowed scheduler slack due to queueing and or processing in milliseconds.

  Type: int

- **SendSlack**

  is the maximum allowed send queue slack due to queueing and or congestion in milliseconds.

  Type: int

- **DecoySlack**

  is the maximum allowed decoy sweep slack due to various external delays such as latency before a loop decoy packet will be considered lost.

  Type: int

- **ConnectTimeout**

  specifies the maximum time a connection can take to establish a TCP/IP connection in milliseconds.

  Type: int

- **HandshakeTimeout**

  specifies the maximum time a connection can take for a link protocol handshake in milliseconds.

  Type: int

- **ReauthInterval**

  specifies the interval at which a connection will be reauthenticated in milliseconds.

  Type: int

- **SendDecoyTraffic**

  enables sending decoy traffic. This is still experimental and untuned and thus is disabled by default. WARNING: This option will go away once decoy traffic is more concrete.

  Type: bool

- **DisableRateLimit**

  disables the per-client rate limiter. This option should only be used for testing.

  Type: bool

- **GenerateOnly**

  halts and cleans up the server right after long term key generation.

  Type: bool

Configuring service nodes

The following configuration is drawn from the reference implementation in `katzenpost/docker/voting_mixnet/servicenode1/authority.toml`. In a real-world mixnet, the component hosts would not be sharing a single IP address. For more information about the test mixnet, see Using the Katzenpost test network.

Server section

The Server section contains mandatory information common to all nodes, for example:

```
[Server]
    Identifier = "servicenode1"
    WireKEM = "xwing"
    PKISignatureScheme = "Ed25519"
    Addresses = ["127.0.0.1:30006"]
    OnlyAdvertiseAltAddresses = false
    MetricsAddress = "127.0.0.1:30007"
    DataDir = "/voting_mixnet/servicenode1"
    IsGatewayNode = false
    IsServiceNode = true
    [Server.AltAddresses]
```

- **Identifier**

  Identifier is the human readable identifier for the node (eg: FQDN).

  Type: string

- **WireKEM**

  // WireKEM is the KEM string representing the chosen KEM scheme with which to communicate

  // with the mixnet and dirauth nodes.

  Type: string

- **PKISignatureScheme**

  PKISignatureScheme specifies the cryptographic signature scheme

  Type: string

- **Addresses**

  // Addresses are the IP address/port combinations that the server will bind

  // to for incoming connections.

  Type: []string

- **OnlyAdvertiseAltAddresses**

  // If set to true then only advertise to the PKI the AltAddresses

  // and do NOT send any of the Addresses.

Type: bool

- **MetricsAddress**

  MetricsAddress is the address/port to bind the prometheus metrics endpoint to.

  Type: string

- **DataDir**

  DataDir is the absolute path to the server's state files.

  Type: string

- **IsGatewayNode**

  IsGatewayNode specifies if the server is a gateway or not.

  Type: bool

- **IsServiceNode**

  IsServiceNode specifies if the server is a service node or not.

  Type: bool

- **[Server.AltAddresses]**

  Map of additional transport protocols and addresses at which the node is reachable by clients, in the form

  ```
  [Server.AltAddresses]
      TCP = ["localhost:30004"]
  ```

  Type: []string

Logging section

The logging configuration section controls logging.

```
[Logging]
    Disable = false
    File = "katzenpost.log"
    Level = "INFO"
```

- **Disable**

  Disables logging if set to **true**.

  Type: bool

- **File**

  Specifies the log file. If omitted, stdout is used.

  Type: string

- **Level**

Supported values are ERROR | WARNING | NOTICE |INFO | DEBUG.

Type: string

Warning

The DEBUG log level is unsafe for production use.

ServiceNode section

The service node configuration section contains subsections with settings for each service that Katzenpost supports. In a production network, the various services would be hosted on dedicated systems.

```
[ServiceNode]

    [[ServiceNode.Kaetzchen]]
        Capability = "echo"
        Endpoint = "+echo"
        Disable = false

    [[ServiceNode.CBORPluginKaetzchen]]
        Capability = "spool"
        Endpoint = "+spool"
        Command = "/voting_mixnet/memspool.alpine"
        MaxConcurrency = 1
        Disable = false
        [ServiceNode.CBORPluginKaetzchen.Config]
            data_store = "/voting_mixnet/servicenode1/memspool.storage"
            log_dir = "/voting_mixnet/servicenode1"

    [[ServiceNode.CBORPluginKaetzchen]]
        Capability = "pigeonhole"
        Endpoint = "+pigeonhole"
        Command = "/voting_mixnet/pigeonhole.alpine"
        MaxConcurrency = 1
        Disable = false
        [ServiceNode.CBORPluginKaetzchen.Config]
            db = "/voting_mixnet/servicenode1/map.storage"
            log_dir = "/voting_mixnet/servicenode1"

    [[ServiceNode.CBORPluginKaetzchen]]
        Capability = "panda"
        Endpoint = "+panda"
        Command = "/voting_mixnet/panda_server.alpine"
        MaxConcurrency = 1
        Disable = false
        [ServiceNode.CBORPluginKaetzchen.Config]
            fileStore = "/voting_mixnet/servicenode1/panda.storage"
            log_dir = "/voting_mixnet/servicenode1"
            log_level = "INFO"

    [[ServiceNode.CBORPluginKaetzchen]]
        Capability = "http"
        Endpoint = "+http"
        Command = "/voting_mixnet/proxy_server.alpine"
        MaxConcurrency = 1
        Disable = false
        [ServiceNode.CBORPluginKaetzchen.Config]
            host = "localhost:4242"
            log_dir = "/voting_mixnet/servicenode1"
            log_level = "DEBUG"
```

Common parameters:

- **Capability**

  The capability exposed by the agent.

  Type: string

- **Endpoint**

  // Endpoint is the provider side endpoint that the agent will accept

  // requests at. While not required by the spec, this server only

  // supports Endpoints that are lower-case local-parts of an e-mail

  // address.

  Type: string

- **Command**

  // Command is the full file path to the external plugin program

  // that implements this Kaetzchen service.

  Type: string

- **MaxConcurrency**

  // MaxConcurrency is the number of worker goroutines to start

  // for this service.

  Type: int

- **Config**

  The extra per agent arguments to be passed to the agent's initialization routine.

  Type: map[string]interface{}

- **Disable**

  disabled a configured agent.

  Type: bool

Per-service parameters:

- **Kaetzchen**
- **spool**
  - **data_store**

    Type:
  - **log_dir**

    Type:
- **pigeonhole**
  - **db**

    Type:
  - **log_dir**

    Type:
- **panda**
  - **fileStore**

    Type:
  - **log_dir**

    Type:
  - **log_level**

    Supported values are ERROR | WARNING | NOTICE |INFO | DEBUG.

    Type: string

    Warning

    The DEBUG log level is unsafe for production use.

    Type: string
- **http**
  - **host**

    Type:
  - **log_dir**

Type:

- **log_level**

  Supported values are ERROR | WARNING | NOTICE |INFO | DEBUG.

  Type: string

  Warning

  The DEBUG log level is unsafe for production use.

  Type: string

PKI section

The PKI section contains the directory authority configuration for a mix, gateway, or service node.

```
[PKI]
    [PKI.Voting]

        [[PKI.Voting.Authorities]]
            Identifier = "auth1"
            IdentityPublicKey = "-----BEGIN ED25519 PUBLIC KEY-----\n/v3qYgh2TvV5Z
            PKISignatureScheme = "Ed25519"
            LinkPublicKey = "-----BEGIN XWING PUBLIC KEY-----\nJeFaZoYQEOO71zPFFWj
            WireKEMScheme = "xwing"
            Addresses = ["127.0.0.1:30001"]

        [[PKI.Voting.Authorities]]
            Identifier = "auth2"
            IdentityPublicKey = "-----BEGIN ED25519 PUBLIC KEY-----\n60KQRhG7njt+k
            PKISignatureScheme = "Ed25519"
            LinkPublicKey = "-----BEGIN XWING PUBLIC KEY-----\nHVR2m7i6G6cf1qxUvyE
            WireKEMScheme = "xwing"
            Addresses = ["127.0.0.1:30002"]

        [[PKI.Voting.Authorities]]
            Identifier = "auth3"
            IdentityPublicKey = "-----BEGIN ED25519 PUBLIC KEY-----\naZUXqznyLO2mK
            PKISignatureScheme = "Ed25519"
            LinkPublicKey = "-----BEGIN XWING PUBLIC KEY-----\nEZukXtZwHTjGj7tCI0k
            WireKEMScheme = "xwing"
            Addresses = ["127.0.0.1:30003"]
```

- **Identifier**

  Identifier is the human readable identifier for the node (eg: FQDN).

  Type: string

- **IdentityPublicKey**

  // IdentityPublicKeyPem is a string in PEM format containing

// the public identity key key.

Type: string

- **PKISignatureScheme**

  PKISignatureScheme specifies the cryptographic signature scheme

  Type: string

- **LinkPublicKey**

  LinkPublicKeyPem is string containing the PEM format of the peer's public link layer key.

  Type: string

- **WireKEMScheme**

  WireKEMScheme is the wire protocol KEM scheme to use.

  Type: string

- **Addresses**

  // Addresses are the IP address/port combinations that the peer authority

  // uses for the Directory Authority service.

  Type: []string

Management section

Management is the Katzenpost management interface configuration. The management section specifies connectivity information for the Katzenpost control protocol which can be used to make configuration changes during run-time. An example configuration looks like this:

```
[Management]
    Enable = false
    Path = "/voting_mixnet/mix1/management_sock"
```

- **Enable**

  Enables the management interface if set to true.

  Type: bool

- **Path**

  Specifies the path to the management interface socket. If left empty, then management_sock will be used under the DataDir.

  Type: string

SphinxGeometry section

```
[SphinxGeometry]
```

```
PacketLength = 3082
NrHops = 5
HeaderLength = 476
RoutingInfoLength = 410
PerHopRoutingInfoLength = 82
SURBLength = 572
SphinxPlaintextHeaderLength = 2
PayloadTagLength = 32
ForwardPayloadLength = 2574
UserForwardPayloadLength = 2000
NextNodeHopLength = 65
SPRPKeyMaterialLength = 64
NIKEName = "x25519"
KEMName = ""
```

- **PacketLength**

  PacketLength is the length of a packet.

  Type: int

- **NrHops**

  // NrHops is the number of hops, this indicates the size

  // of the Sphinx packet header.

  Type: int

- **HeaderLength**

  HeaderLength is the length of the Sphinx packet header in bytes.

  Type: int

- **RoutingInfoLength**

  RoutingInfoLength is the length of the routing info portion of the header.

  Type: int

- **PerHopRoutingInfoLength**

  PerHopRoutingInfoLength is the length of the per hop routing info.

  Type: int

- **SURBLength**

  SURBLength is the length of SURB.

  Type: int

- **SphinxPlaintextHeaderLength**

  SphinxPlaintextHeaderLength is the length of the plaintext header.

  Type: int

- **PayloadTagLength**

  PayloadTagLength is the length of the payload tag.

  Type: int

- **ForwardPayloadLength**

  ForwardPayloadLength is the size of the payload.

  Type: int

- **UserForwardPayloadLength**

  the size of the usable payload.

  Type: int

- **NextNodeHopLength**

  // NextNodeHopLength is derived off the largest routing info

  // block that we expect to encounter. Everything else just has a

  // NextNodeHop + NodeDelay, or a Recipient, both cases which are

  // shorter.

  Type: int

- **SPRPKeyMaterialLength**

  SPRPKeyMaterialLength is the length of the SPRP key.

  Type: int

- **NIKEName**

  // NIKEName is the name of the NIKE scheme used by the mixnet's Sphinx packet.

  // NIKEName and KEMName are mutually exclusive.

  Type: string

- **KEMName**

  KEMName is the name of the KEM scheme used by the mixnet's Sphinx packet. NIKEName and KEMName are mutually exclusive.

  Type: string

Debug section

Debug is the Katzenpost server debug configuration for advanced tuning.

```
[Debug]
                    NumSphinxWorkers = 16
                    NumServiceWorkers = 3
```

```
NumGatewayWorkers = 3
NumKaetzchenWorkers = 3
SchedulerExternalMemoryQueue = false
SchedulerQueueSize = 0
SchedulerMaxBurst = 16
UnwrapDelay = 250
GatewayDelay = 500
ServiceDelay = 500
KaetzchenDelay = 750
SchedulerSlack = 150
SendSlack = 50
DecoySlack = 15000
ConnectTimeout = 60000
HandshakeTimeout = 30000
ReauthInterval = 30000
SendDecoyTraffic = false
DisableRateLimit = false
GenerateOnly = false
```

- **NumSphinxWorkers**

  specifies the number of worker instances to use for inbound Sphinx packet processing.

  Type: int

- **NumProviderWorkers**

  specifies the number of worker instances to use for provider specific packet processing.

  Type: int

- **NumKaetzchenWorkers**

  specifies the number of worker instances to use for Kaetzchen specific packet processing.

  Type: int

- **SchedulerExternalMemoryQueue**

  will enable the experimental external memory queue that is backed by disk.

  Type: bool

- **SchedulerQueueSize**

  is the maximum allowed scheduler queue size before random entries will start getting dropped. A value <= 0 is treated as unlimited.

  Type: int

- **SchedulerMaxBurst**

  is the maximum number of packets that will be dispatched per scheduler wakeup event.

  Type:

- **UnwrapDelay**

is the maximum allowed unwrap delay due to queueing in milliseconds.

Type: int

- **GatewayDelay**

  the maximum allowed gateway node worker delay due to queueing

  in milliseconds.

  Type: int

- **ServiceDelay**

  is the maximum allowed provider delay due to queueing in milliseconds.

  Type: int

- **KaetzchenDelay**

  is the maximum allowed kaetzchen delay due to queueing in milliseconds.

  Type: int

- **SchedulerSlack**

  is the maximum allowed scheduler slack due to queueing and or processing in milliseconds.

  Type: int

- **SendSlack**

  is the maximum allowed send queue slack due to queueing and or congestion in milliseconds.

  Type: int

- **DecoySlack**

  is the maximum allowed decoy sweep slack due to various external delays such as latency before a loop decoy packet will be considered lost.

  Type: int

- **ConnectTimeout**

  specifies the maximum time a connection can take to establish a TCP/IP connection in milliseconds.

  Type: int

- **HandshakeTimeout**

  specifies the maximum time a connection can take for a link protocol handshake in milliseconds.

  Type: int

- **ReauthInterval**

  specifies the interval at which a connection will be reauthenticated in milliseconds.

Type: int

- **SendDecoyTraffic**

  enables sending decoy traffic. This is still experimental and untuned and thus is disabled by default. WARNING: This option will go away once decoy traffic is more concrete.

  Type: bool

- **DisableRateLimit**

  disables the per-client rate limiter. This option should only be used for testing.

  Type: bool

- **GenerateOnly**

  halts and cleans up the server right after long term key generation.

  Type: bool