Chapter 1. Components of the Katzenpost mixnet

Table of Contents

Directory authorities 1
    Configuring directory authorities 1
Mix, gateway, and service nodes 10
    Configuring mix nodes 10
    Configuring gateway nodes 17
    Configuring service nodes 25

To do: Introduction

Directory authorities

To do: Introduction

Configuring directory authorities[1]

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

Note

Katzenpost configuration files are written in TOML [https://toml.io/en/v1.0.0]. A block within single square brackets describes a *table*, which is a list of key/value pairs. A block within double square brackets describes an array of tables, where the declaration is also the first element of the array.

Server section

To do: Introduction

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

- **Identifier**

  A human-readable identifier for the peer, for example, an FQDN.

  Type: string

- **WireKEMScheme**

[1] dwrob: After first use, should we refer to directory authorites as authorities, nodes, or peers?

Specifies the wire protocol KEM scheme to use.

Type: string

- **PKISignatureScheme**

  Specifies the cryptographic signature scheme.

  Type: string

- **Addresses**

  A list of IP address/port combinations that the peer will bind to for incoming connections.

  Type: []string

- **DataDir**

  The absolute path to the peer's state files.

  Type: string

Authorities section

An Authorities section is configured for each peer directory authority.

```
[[Authorities]]
    Identifier = "auth1"
    IdentityPublicKey = "-----BEGIN ED25519 PUBLIC KEY-----\n/v3qYgh2TvV5ZqEVgw
    PKISignatureScheme = "Ed25519"
    LinkPublicKey = "-----BEGIN XWING PUBLIC KEY-----\nJeFaZoYQEOO71zPFFWjL7DyD
    WireKEMScheme = "xwing"
    Addresses = ["127.0.0.1:30001"]

[[Authorities]]
    Identifier = "auth2"
    IdentityPublicKey = "-----BEGIN ED25519 PUBLIC KEY-----\n60KQRhG7njt+kLQuwW
    PKISignatureScheme = "Ed25519"
    LinkPublicKey = "-----BEGIN XWING PUBLIC KEY-----\nHVR2m7i6G6cf1qxUvyEr3KC7
    WireKEMScheme = "xwing"
    Addresses = ["127.0.0.1:30002"]

[[Authorities]]
    Identifier = "auth3"
    IdentityPublicKey = "-----BEGIN ED25519 PUBLIC KEY-----\naZUXqznyLO2mKDceID
    PKISignatureScheme = "Ed25519"
    LinkPublicKey = "-----BEGIN XWING PUBLIC KEY-----\nEZukXtZwHTjGj7tCI0kmUcq0
    WireKEMScheme = "xwing"
    Addresses = ["127.0.0.1:30003"]
```

- **Identifier**

  A human-readable identifier for the peer, for example, an FQDN.

  Type: string

- **IdentityPublicKey**

  The peer's public identity key in PEM format.

Type: sign.PublicKey

- **PKISignatureScheme**

  Specifies the peer's cryptographic signature scheme.

  Type: string

- **LinkPublicKey**

  The peer's public link layer key in PEM format.

  Type: kem.PublicKey

- **WireKEMScheme**

  Specifies the wire protocol KEM scheme to use.

  Type: string

- **Addresses**

  A list of local IP address/port combinations that the peer will bind to for incoming connections. These can be specified as either IPv4 or IPv6 addresses.

  Type: []string

Logging section

The logging configuration section controls log storage and logging level.

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

- **Disable**

  If **true**, logging is disabled.

  Type: bool

- **File**

  Specifies the log file. If omitted, logging is written to stdout.

  Type: string

- **Level**

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

  Type: string

Warning

The DEBUG log level is unsafe for production use because it discloses sensitive information.

Parameters section

The Parameters section defines the values of network parameters.

[2]

```
[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**

  Maximum rate of packets per client per minute.[3]

  Type: uint64

- **Mu**

  The inverse of the mean of the exponential distribution[4] used to determine the Sphinx packet per-hop mixing delay.

  Type: float64

- **MuMaxDelay**

  Sets the maximum delay for Mu, in millisecods.

  Type: uint64

- **LambdaP**

  Specifies the inverse of the mean of the exponential distribution that a client uses to determine the delay interval between packets leaving its FIFO egress queue or, if the queue is empty, before dropping decoy packets.

  Type: float64

- **LambdaPMaxDelay**

  Sets the maximum delay for LambdaP, in milliseconds.

  Type: uint64

- **LambdaL**

  Specifies the inverse of the mean of the exponential distribution that clients use to select the delay interval between loop decoy packets.

  Type: float64

[2] dwrob: I am only pretending to understand the math involved here, so please read my wording critically.

[3] dwrob: Why is this set to zero?

[4] dwrob: Could we just substitute "rate parameter" for each use of this phrase?

- **LambdaLMaxDelay**

  Sets the maximum delay for LambdaL, in milliseconds.

  Type: uint64

- **LambdaD**

  Specifies the inverse of the mean of the exponential distribution that clients use to determine the delay interval before sending decoy drop messages.

  Type: float64

- **LambdaDMaxDelay**

  Sets the maximum delay for LambdaD, in milliseconds.

  Type: uint64

- **LambdaM**

  Specifies the inverse of the mean of the exponential distribution that mixes use to determine the send timing of mix loop decoy traffic.

  Type: float64

- **LambdaG**

  Specifies 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.[5]

  Type: float64

- **LambdaMMaxDelay**

  Sets the maximum delay for LambdaM, in milliseconds.

  Type: uint64

- **LambdaGMaxDelay**

  Sets the maximum delay for LambdaG, in milliseconds.

  Type: uint64

Debug section

To do: Introduction

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

- **Layers**

[5] dwrob: What does this mean and why is it a warning?

Number of non-provider layers[6] in the network topology.

Type: int

- **MinNodesrPerLayer**

  Minimum number of nodes[7] per layer required to form a valid consensus 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.[8]

```
[[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.[9]

```
[[GatewayNodes]]
```

[6] dwrob: What are these, is is "provider" the desired term here?
[7] dwrob: What kind of nodes are these?
[8] dwrob: These definitions differ significantly from the code comments.
[9] dwrob: These definitions differ significantly from the code comments.

```
    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. [10]

```
[[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"
```

[10] dwrob: These definitions differ significantly from the code comments.

```
    [[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

To do: Introduction

```
[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 total length of a Sphinx packet.

  Type: int

- **NrHops**

  NrHops is the number of permitted hops for a packet. This setting influences 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 Sphinx packet header.

Type: int

- **PerHopRoutingInfoLength**

  PerHopRoutingInfoLength is the length of the per-hop routing info in the Sphinx packet header.

  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 Sphinx packet's usable payload.

  Type: int

- **NextNodeHopLength**

  NextNodeHopLength is derived from 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 packets. 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.

Note

Katzenpost configuration files are written in TOML [https://toml.io/en/v1.0.0]. A block within single square brackets describes a *table*, which is a list of key/value pairs. A block within double square brackets describes an array of tables, where the declaration is also the first element of the array.

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**

  A human-readable identifier for the node, for example, an 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**

  A list of IP address/port combinations that the server will bind to for incoming connections to the mixnet.

  Type: []string

- **OnlyAdvertiseAltAddresses**

  If **true**, **true**, only advertise AltAddresses to the PKI, not Addresses.

  Type: bool

- **MetricsAddress**

  MetricsAddress is the IP 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**

  If **true**, specifies that the server is a gateway node.

  Type: bool

- **IsServiceNode**

  If **true**, specifies that the server is a service node.

  Type: bool

- **[Server.AltAddresses]**

  A 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 log storage and logging level.

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

- **Disable**

  If **true**, logging is disabled.

  Type: bool

- **File**

  Specifies the log file. If omitted, logging is written to stdout.

  Type: string

- **Level**

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

Type: string

Warning

The DEBUG log level is unsafe for production use because it discloses sensitive information.

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/v3qYgh2Tv
            PKISignatureScheme = "Ed25519"
            LinkPublicKey = "-----BEGIN XWING PUBLIC KEY-----\nJeFaZoYQEOO71zPF
            WireKEMScheme = "xwing"
            Addresses = ["127.0.0.1:30001"]

        [[PKI.Voting.Authorities]]
            Identifier = "auth2"
            IdentityPublicKey = "-----BEGIN ED25519 PUBLIC KEY-----\n60KQRhG7nj
            PKISignatureScheme = "Ed25519"
            LinkPublicKey = "-----BEGIN XWING PUBLIC KEY-----\nHVR2m7i6G6cf1qxU
            WireKEMScheme = "xwing"
            Addresses = ["127.0.0.1:30002"]

        [[PKI.Voting.Authorities]]
            Identifier = "auth3"
            IdentityPublicKey = "-----BEGIN ED25519 PUBLIC KEY-----\naZUXqznyLO
            PKISignatureScheme = "Ed25519"
            LinkPublicKey = "-----BEGIN XWING PUBLIC KEY-----\nEZukXtZwHTjGj7tC
            WireKEMScheme = "xwing"
            Addresses = ["127.0.0.1:30003"]
```

- **Identifier**

  A human-readable identifier for the node, for example, an FQDN.

  Type: string

- **IdentityPublicKey**

  The public identity key in PEM format.

  Type: string

- **PKISignatureScheme**

  Specifies the cryptographic signature scheme

  Type: string

- **LinkPublicKey**

  The peer's public link-layer key in PEM format.

Type: string

- **WireKEMScheme**

  Specifies the wire protocol KEM scheme.

  Type: string

- **Addresses**

  A list of IP address/port combinations that peer authority[11] 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.
  [12]

  Type: string

SphinxGeometry section

To do: Introduction

```
[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
```

[11] dwrob: Should be "the service node"?

[12] dwrob: Confusing wording.

```
NIKEName = "x25519"
KEMName = ""
```

- **PacketLength**

  PacketLength is the total length of a Sphinx packet.

  Type: int

- **NrHops**

  NrHops is the number of permitted hops for a packet. This setting influences 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 Sphinx packet header.

  Type: int

- **PerHopRoutingInfoLength**

  PerHopRoutingInfoLength is the length of the per-hop routing info in the Sphinx packet header.

  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 Sphinx packet's usable payload.

  Type: int

- **NextNodeHopLength**

  NextNodeHopLength is derived from 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 packets. NIKEName and KEMName are mutually exclusive.

  Type: string

Debug section

The Katzenpost server debug configuration is used 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 for processing inbound Sphinx packets.

  Type: int

- **NumProviderWorkers**

Specifies the number of worker instances for processing provider-specific packets.

Type: int

- **NumKaetzchenWorkers**

  Specifies the number of worker instances for processing Kaetzchen-specific packets.

  Type: int

- **SchedulerExternalMemoryQueue**

  If **true**, enables the experimental external memory queue that is backed backed up to disk.

  Type: bool

- **SchedulerQueueSize**

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

  Type: int

- **SchedulerMaxBurst**

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

  Type:

- **UnwrapDelay**

  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**

  The maximum allowed provider delay due to queueing, in milliseconds.

  Type: int

- **KaetzchenDelay**

  The maximum allowed kaetzchen delay due to queueing, in milliseconds.

  Type: int

- **SchedulerSlack**

  The maximum allowed scheduler slack due to queueing and/or processing, in milliseconds.

  Type: int

- **SendSlack**

  The maximum allowed send queue slack due to queueing and/or congestion, in milliseconds.

Type: int

- **DecoySlack**

  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 after which a connection will be reauthenticated, in milliseconds.

  Type: int

- **SendDecoyTraffic**

  If **true**, enables sending decoy traffic. Disabled by default.

  Type: bool

- **DisableRateLimit**

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

  Type: bool

- **GenerateOnly**

  If **true**, halts and cleans up the server 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**

  A human-readable identifier for the node, for example, an 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**

  A list of IP address/port combinations that the server will bind to for incoming connections to the mixnet.

  Type: []string

- **OnlyAdvertiseAltAddresses**

  If **true**, **true**, only advertise AltAddresses to the PKI, not Addresses.

  Type: bool

- **MetricsAddress**

  MetricsAddress is the IP 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**

  If **true**, specifies that the server is a gateway node.

  Type: bool

- **IsServiceNode**

  If **true**, specifies that the server is a service node.

Type: bool

- **[Server.AltAddresses]**

  A 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 log storage and logging level.

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

- **Disable**

  If **true**, logging is disabled.

  Type: bool

- **File**

  Specifies the log file. If omitted, logging is written to stdout.

  Type: string

- **Level**

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

  Type: string

Warning

The DEBUG log level is unsafe for production use because it discloses sensitive information.

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"
```

-
-
-

[14]

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/v3qYgh2Tv
            PKISignatureScheme = "Ed25519"
            LinkPublicKey = "-----BEGIN XWING PUBLIC KEY-----\nJeFaZoYQEOO71zPF
            WireKEMScheme = "xwing"
            Addresses = ["127.0.0.1:30001"]

        [[PKI.Voting.Authorities]]
            Identifier = "auth2"
            IdentityPublicKey = "-----BEGIN ED25519 PUBLIC KEY-----\n60KQRhG7nj
            PKISignatureScheme = "Ed25519"
            LinkPublicKey = "-----BEGIN XWING PUBLIC KEY-----\nHVR2m7i6G6cf1qxU
            WireKEMScheme = "xwing"
            Addresses = ["127.0.0.1:30002"]

        [[PKI.Voting.Authorities]]
            Identifier = "auth3"
            IdentityPublicKey = "-----BEGIN ED25519 PUBLIC KEY-----\naZUXqznyLO
            PKISignatureScheme = "Ed25519"
            LinkPublicKey = "-----BEGIN XWING PUBLIC KEY-----\nEZukXtZwHTjGj7tC
            WireKEMScheme = "xwing"
            Addresses = ["127.0.0.1:30003"]
```

- **Identifier**

  A human-readable identifier for the node, for example, an FQDN.

  Type: string

- **IdentityPublicKey**

  The public identity key in PEM format.

  Type: string

- **PKISignatureScheme**

  Specifies the cryptographic signature scheme

  Type: string

- **LinkPublicKey**

  The peer's public link-layer key in PEM format.

  Type: string

- **WireKEMScheme**

[14] dwrob: To do

Specifies the wire protocol KEM scheme.

Type: string

- **Addresses**

  A list of IP address/port combinations that peer authority[15] 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.
  [16]

  Type: string

SphinxGeometry section

To do: Introduction

```
[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 total length of a Sphinx packet.

[15] dwrob: Should be "the service node"?
[16] dwrob: Confusing wording.

Type: int

- **NrHops**

  NrHops is the number of permitted hops for a packet. This setting influences 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 Sphinx packet header.

  Type: int

- **PerHopRoutingInfoLength**

  PerHopRoutingInfoLength is the length of the per-hop routing info in the Sphinx packet header.

  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 Sphinx packet's usable payload.

  Type: int

- **NextNodeHopLength**

  NextNodeHopLength is derived from 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 packets. NIKEName and KEMName are mutually exclusive.

  Type: string

Debug section

The Katzenpost server debug configuration is used 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 for processing inbound Sphinx packets.

  Type: int

- **NumProviderWorkers**

  Specifies the number of worker instances for processing provider-specific packets.

  Type: int

- **NumKaetzchenWorkers**

  Specifies the number of worker instances for processing Kaetzchen-specific packets.

Type: int

- **SchedulerExternalMemoryQueue**

  If **true**, enables the experimental external memory queue that is backed backed up to disk.

  Type: bool

- **SchedulerQueueSize**

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

  Type: int

- **SchedulerMaxBurst**

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

  Type:

- **UnwrapDelay**

  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**

  The maximum allowed provider delay due to queueing, in milliseconds.

  Type: int

- **KaetzchenDelay**

  The maximum allowed kaetzchen delay due to queueing, in milliseconds.

  Type: int

- **SchedulerSlack**

  The maximum allowed scheduler slack due to queueing and/or processing, in milliseconds.

  Type: int

- **SendSlack**

  The maximum allowed send queue slack due to queueing and/or congestion, in milliseconds.

  Type: int

- **DecoySlack**

  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 after which a connection will be reauthenticated, in milliseconds.

  Type: int

- **SendDecoyTraffic**

  If **true**, enables sending decoy traffic. Disabled by default.

  Type: bool

- **DisableRateLimit**

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

  Type: bool

- **GenerateOnly**

  If **true**, halts and cleans up the server 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**

A human-readable identifier for the node, for example, an 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**

  A list of IP address/port combinations that the server will bind to for incoming connections to the mixnet.

  Type: []string

- **OnlyAdvertiseAltAddresses**

  If **true**, **true**, only advertise AltAddresses to the PKI, not Addresses.

  Type: bool

- **MetricsAddress**

  MetricsAddress is the IP 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**

  If **true**, specifies that the server is a gateway node.

  Type: bool

- **IsServiceNode**

  If **true**, specifies that the server is a service node.

  Type: bool

- **[Server.AltAddresses]**

  A 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 log storage and logging level.

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

- **Disable**

  If **true**, logging is disabled.

  Type: bool

- **File**

  Specifies the log file. If omitted, logging is written to stdout.

  Type: string

- **Level**

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

  Type: string

Warning

The DEBUG log level is unsafe for production use because it discloses sensitive information.

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**

  The provider-side endpoint for the agent accepts requests. While not required by the spec, this server only

  supports Endpoints that are lower-case local-parts of an e-mail address.[17]

  Type: string

- **Command**

  The path and filename of the external plugin program that implements this Kaetzchen service.

  Type: string

- **MaxConcurrency**

  The number of worker goroutines to start for this service.

  Type: int

- **Config**

[17] dwrob: What does this mean? Does it need to be here?

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

Type: map[string]interface{}

- **Disable**

  If true, disables a configured agent.

  Type: bool

Per-service parameters:[18]

- **Kaetzchen**[20]
- **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.

[18] dwrob: About CBOR: https://pkg.go.dev/github.com/katzenpost/katzenpost@v0.0.35/server/cborplugin#ResponseFactory Package cborplugin is a plugin system allowing mix network services to be added in any language. It communicates queries and responses to and from the mix server using CBOR over UNIX domain socket. Beyond that, a client supplied SURB is used to route the response back to the client as described in our Kaetzchen specification document:

[20] dwrob: Needs explanation

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 [19]

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/v3qYgh2Tv
            PKISignatureScheme = "Ed25519"
            LinkPublicKey = "-----BEGIN XWING PUBLIC KEY-----\nJeFaZoYQEOO71zPF
            WireKEMScheme = "xwing"
            Addresses = ["127.0.0.1:30001"]

        [[PKI.Voting.Authorities]]
            Identifier = "auth2"
            IdentityPublicKey = "-----BEGIN ED25519 PUBLIC KEY-----\n60KQRhG7nj
            PKISignatureScheme = "Ed25519"
            LinkPublicKey = "-----BEGIN XWING PUBLIC KEY-----\nHVR2m7i6G6cf1qxU
            WireKEMScheme = "xwing"
            Addresses = ["127.0.0.1:30002"]

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

[19] dwrob: To oo

```
Identifier = "auth3"
IdentityPublicKey = "-----BEGIN ED25519 PUBLIC KEY-----\naZUXqznyLO
PKISignatureScheme = "Ed25519"
LinkPublicKey = "-----BEGIN XWING PUBLIC KEY-----\nEZukXtZwHTjGj7tC
WireKEMScheme = "xwing"
Addresses = ["127.0.0.1:30003"]
```

- **Identifier**

  A human-readable identifier for the node, for example, an FQDN.

  Type: string

- **IdentityPublicKey**

  The public identity key in PEM format.

  Type: string

- **PKISignatureScheme**

  Specifies the cryptographic signature scheme

  Type: string

- **LinkPublicKey**

  The peer's public link-layer key in PEM format.

  Type: string

- **WireKEMScheme**

  Specifies the wire protocol KEM scheme.

  Type: string

- **Addresses**

  A list of IP address/port combinations that peer authority[21] 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**

[21] dwrob: Should be "the service node"?

Specifies the path to the management interface socket. If left empty, then management_sock will be used under the DataDir.
[22]

Type: string

SphinxGeometry section

To do: Introduction

```
[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 total length of a Sphinx packet.

  Type: int

- **NrHops**

  NrHops is the number of permitted hops for a packet. This setting influences 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 Sphinx packet header.

  Type: int

- **PerHopRoutingInfoLength**

  PerHopRoutingInfoLength is the length of the per-hop routing info in the Sphinx packet header.

  Type: int

- **SURBLength**

[22] dwrob: Confusing wording.

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 Sphinx packet's usable payload.

  Type: int

- **NextNodeHopLength**

  NextNodeHopLength is derived from 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 packets. NIKEName and KEMName are mutually exclusive.

  Type: string

Debug section

The Katzenpost server debug configuration is used 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 for processing inbound Sphinx packets.

  Type: int

- **NumProviderWorkers**

  Specifies the number of worker instances for processing provider-specific packets.

  Type: int

- **NumKaetzchenWorkers**

  Specifies the number of worker instances for processing Kaetzchen-specific packets.

  Type: int

- **SchedulerExternalMemoryQueue**

  If **true**, enables the experimental external memory queue that is backed backed up to disk.

  Type: bool

- **SchedulerQueueSize**

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

  Type: int

- **SchedulerMaxBurst**

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

  Type:

- **UnwrapDelay**

  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**

  The maximum allowed provider delay due to queueing, in milliseconds.

  Type: int

- **KaetzchenDelay**

  The maximum allowed kaetzchen delay due to queueing, in milliseconds.

  Type: int

- **SchedulerSlack**

  The maximum allowed scheduler slack due to queueing and/or processing, in milliseconds.

  Type: int

- **SendSlack**

  The maximum allowed send queue slack due to queueing and/or congestion, in milliseconds.

  Type: int

- **DecoySlack**

  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 after which a connection will be reauthenticated, in milliseconds.

  Type: int

- **SendDecoyTraffic**

  If **true**, enables sending decoy traffic. Disabled by default.

  Type: bool

- **DisableRateLimit**

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

Type: bool

- **GenerateOnly**

  If **true**, halts and cleans up the server after long term key generation.

  Type: bool