CybOX 3.0 Specification - Pre-Draft

CybOX Network Objects – Version 0.1

Specifications Cover Page

1. IPv4 Address Object

Type Name: `ipv4-address-object`

The IPv4 Address Object represents one or more IPv4 addresses expressed using CIDR notation.

1.1. Properties

Property Name	Type	Description
`type` (required)	`string`	The value of this field **MUST** be `ipv4-address-object`.
`value` (required)	`string`	Specifies one or more IPv4 addresses expressed using CIDR notation. If a given IPv4 Address Object represents a single IPv4 address the CIDR /32 suffix **MAY** be omitted.
`resolves_to_refs` (optional)	`list` of type `object-ref`	Specifies a list of references to one or more Media Access Control (MAC) addresses that the IPv4 address resolves to. The objects referenced in this list **MUST** be of type `mac-address-object`.
`belongs_to_refs` (optional)	`list` of type `object-ref`	Specifies a reference to one or more autonomous systems (AS) that the IPv4 address belongs to. The objects referenced in this list **MUST** be of type `as-object`.

1.2. Examples

IPv4 single address

```
{
  "type": "cybox-container",
  "spec_version": "3.0",
  "objects": {
    "0": {
      "type": "ipv4-address-object",
      "value": "1.2.3.4"
    }
  }
}
```

IPv4 CIDR block

```
{
  "type": "cybox-container",
  "spec_version": "3.0",
  "objects": {
    "0": {
      "type": "ipv4-address-object",
      "value": "192.168.0.0/16"
    }
  }
}
```

2. IPv6 Address Object

Type Name: `ipv6-address-object`

The IPv6 Address Object represents one or more IPv6 addresses expressed using CIDR notation.

2.1. Properties

Property Name	Type	Description
`type` (required)	`string`	The value of this field **MUST** be `ipv6-address-object`.
`value` (required)	`string`	Specifies one or more IPv6 addresses expressed using CIDR notation. If a given IPv6 Address Object represents a single IPv6 address the CIDR /128 suffix **MAY** be omitted.

resolves_to_refs (optional)	list of type object-ref	Specifies a list of references to one or more Media Access Control (MAC) addresses that the IPv6 address resolves to. The objects referenced in this list **MUST** be of type mac-address-object.
belongs_to_refs (optional)	list of type object-ref	Specifies a reference to one or more autonomous systems (AS) that the IPv6 address belongs to. The objects referenced in this list **MUST** be of type as-object.

2.2. Examples

IPv6 single address

```
{
  "type": "cybox-container",
  "spec_version": "3.0",
  "objects": {
    "0": {
      "type": "ipv6-address-object",
      "value": "2001:0db8:85a3:0000:0000:8a2e:0370:7334"
    }
  }
}
```

IPv6 CIDR block

```
{
  "type": "cybox-container",
  "spec_version": "3.0",
  "objects": {
    "0": {
      "type": "ipv6-address-object",
      "value": "2001:db8::/96"
    }
  }
}
```

3. MAC Address Object

Type Name: mac-address-object

The MAC Address Object represents a single Media Access Control (MAC) address.

3.1. Properties

Property Name	Type	Description
type (required)	`string`	The value of this field **MUST** be `mac-address-object`.
value (required)	`string`	Specifies a single MAC address. The MAC address value **MUST** be represented as a single colon-delimited, lowercase MAC-48 address, which **MUST** include leading zeros for each octet.

3.2. Examples

Typical MAC address

```
{
  "type": "cybox-container",
  "spec_version": "3.0",
  "objects": {
    "0": {
      "type": “mac-address-object”,
      "value": "d2:fb:49:24:37:18"
    }
  }
}
```

4. Email Address Object

Type Name: `email-address-object`

The Email Address Object represents a single email address.

4.1. Properties

Property Name	Type	Description
type (required)	`string`	The value of this field **MUST** be `email-address-object`.

value (required)	`string`	Specifies a single email address. This **MUST** not include the display name. This property corresponds to the *addr-spec* construction in RFC 5322 Section 3.4.
display_name (optional)	`string`	Specifies a single email display name, i.e., the name that is displayed to the human user of a mail application. This property corresponds to the *display-name* construction in RFC 5322 Section 3.4.

4.2. Example

```
{
  "type": "cybox-container",
  "spec_version": "3.0",
  "objects": {
    "0": {
      "type": "email-address-object",
      "value": "bruce@wayneindustries.com",
      "display_name": "Bruce Wayne"
    }
  }
}
```

5. URL Object

Type Name: `url-object`

The URL Object represents the properties of a uniform resource locator (URL).

5.1. Properties

Property Name	Type	Description
type (required)	`string`	The value of this field **MUST** be `url-object`.
value (required)	`string`	Specifies the value of the URL.

5.2. Example

Basic URL

```
{
  "type": "cybox-container",
  "spec_version": "3.0",
  "objects": {
    "0": {
      "type": "url-object",
      "value": "https://wayneindustries.com"
    }
  }
}
```

6. Domain Name Object

Type Name: `domain-name-object`

The Domain Name represents the properties of a network domain name.

6.1. Properties

Property Name	Type	Description
`type` (required)	`string`	The value of this field **MUST** be `domain-name-object`.
`value` (required)	`string`	Specifies the value of the domain name.
`resolves_to`	`list` of type `object-ref`	Specifies a list of references to one or more IP addresses or domain names that the domain name resolves to. The objects referenced in this list **SHOULD** be of type `ipv4-address-object` or `ipv6-address-object` or **MAY** be of type `domain-name-object` for cases such as CNAME records.

6.2. Example

Basic FQDN

```
{
  "type": "cybox-container",
  "spec_version": "3.0",
  "objects": {
    "0": {
      "type": "domain-name-object",
      "value":"www.example.com"
    }
  }
}
```

7. AS Object

Type Name: `as-object`

The AS object represents the properties of an Autonomous System (AS).

7.1. Properties

Property Name	Type	Description
`type` (required)	`string`	The value of this field **MUST** be `as-object`.
`number` (required)	`integer`	Specifies the number assigned to the Autonomous System (AS). Such assignments are typically performed by a Regional Internet Registries (RIR).
`name` (optional)	`string`	Specifies the name of the Autonomous System (AS).
`regional_internet_registry` (optional)	`string`	Specifies the name of the Regional Internet registry (RIR) that assigned the number to the Autonomous System (AS).
`extended_properties` (optional)	`dictionary`	Specifies any extended properties of the object, as a dictionary.

		Dictionary keys **MUST** identify the extension type by name. The corresponding dictionary values **MUST** contain the contents of the extension instance.

7.2. Example

Basic AS Object

```
{
  "type": "cybox-container",
  "spec_version": "3.0",
  "objects": {
    "0": {
      "type": "as-object",
      "number":"15139",
      "name":"Wayne Industries",
      "handle":"AS15139",
      "regional_internet_registry":"ARIN"
    }
  }
}
```

8. Network Connection

Type Name: `network-connection-object`

The Network Connection Object represents an instance of a unicast, multicast, or broadcast network connection.

A Network Connection Object **MUST** contain at least one of **src_ref** OR **dst_ref** and **SHOULD** contain **protocols**, **src_port,** and **dst_port**.

8.1. Properties

Property Name	Type	Description
type (required)	`string`	The value of this field **MUST** be

		`network-connection-object`.
`start` (optional)	`timestamp`	Specifies the date/time the network connection was initiated, if known.
`end` (optional)	`timestamp`	Specifies the date/time the network connection was closed, if known. If the **is_active** property is true, then the **end** property **MUST NOT** be included.
`is_active` (optional)	`boolean`	Indicates whether the network connection is still active.
`src_ref` (optional)	`object-ref`	Specifies the source of the network connection, as a reference to one or more CybOX Objects. The objects referenced in this list **SHOULD** be of type `ipv4-address-object`, `ipv6-address-object`, `mac-address-object` or **MAY** be of type `domain-name-object` for cases where the IP address for a domain name is unknown.
`dst_ref` (optional)	`object-ref`	Specifies the destination of the network connection, as a reference to one or more CybOX Objects. The objects referenced in this list **SHOULD** be of type `ipv4-address-object`, `ipv6-address-object`, `mac-address-object` or **MAY** be of type `domain-name-object` for cases where the IP address for a domain name is unknown.
`src_port` (optional)	`integer`	Specifies the source port used in the connection, as an integer in the range of 0 - 65535.
`dst_port` (optional)	`integer`	Specifies the destination port used in the connection, as an integer in the range of 0 - 65535.

protocols (optional)	`list` of type `string`	Specifies the protocols used in the network connection, along with their corresponding state. Protocols **MUST** be listed in low to high order, from outer to inner in terms of packet encapsulation. That is, the protocols in the outer level of the packet, such as IP, **MUST** be listed first. The protocol names **SHOULD** come from the service names defined in the Service Name column of the IANA Service Name and Port Number Registry [reference] and **MUST** be represented as a lowercase string with spaces and underscores replaced with dashes. If a protocol name is not defined in this registry, then it **MUST** be represented as a lower-cased string version of the common name of the protocol and **MUST** have any spaces and underscores replaced with dashes.
src_byte_count (optional)	`integer`	The number of bytes sent from the source to the destination.
dst_byte_count (optional)	`integer`	The number of bytes sent from the destination to the source.
src_packets (optional)	`integer`	The number of packets sent from the source to the destination.
dst_packets (optional)	`integer`	The number of packets sent destination to the source.
ipfix_data (optional)	`dictionary`	Specifies any IP Flow Information Export (IPFIX) data for the flow, as a dictionary. Each key/value pair in the dictionary represents the name/value of a single IPFIX element. Accordingly, each dictionary key **MUST** be a lowercase string version of the IPFIX element name, e.g., “octetdeltacount”. Each dictionary value **MUST** be either

		an integer or a string.
src_payload_ref (optional)	`object-ref`	Specifies the bytes sent from the source to the destination. The object referenced in this field **MUST** be of type `artifact-object`.
dst_payload_ref (optional)	`object-ref`	Specifies the bytes sent from the destination to the source. The object referenced in this field **MUST** be of type `artifact-object`.
extended_properties (optional)	`dictionary`	Specifies any extended properties of the object, as a dictionary. Dictionary keys **MUST** identify the extension type by name. The corresponding dictionary values **MUST** contain the contents of the extension instance.

8.2. Example

Basic Network Connection

```
{
  "type":"cybox-container",
  "spec_version":"3.0",
  "objects":{
    "0":{
      "type":"ipv4-address-object",
      "value":"1.2.3.4"
    },
    "1":{
      "type":"ipv4-address-object",
      "value":"2.3.4.5"
    },
    "2":{
      "type":"network-connection-object",
      "src_refs":[
        "0"
      ],
      "dst_refs":[
```

```
        "1"
      ],
      "protocols":["tcp"]
    }
  }
}
```

Network Connection with Netflow Data

```
{
  "type":"cybox-container",
  "spec_version":"3.0",
  "objects":{
    "0":{
      "type":"ipv4-address-object",
      "value":"192.168.43.9"
    },
    "1":{
      "type":"ipv4-address-object",
      "value":"192.168.22.101"
    },
    "2":{
      "type":"network-connection-object",
      "src_refs":[
        "0"
      ],
      "dst_refs":[
        "1"
      ],
      "protocols":["tcp"],
      "src_bytes":147600,
      "src_packets":100
      }
    }
  }
}
```

8.3.1. HTTP Extension

Type Name: `http-extension`

The HTTP extension specifies a default extension for capturing network connection properties specific to HTTP. The key for this extension when used in the **`extended_properties`** dictionary MUST be *http*.

8.3.1.1. Properties

Property Name	Type	Description
`request_method` (optional)	`string`	Specifies the HTTP method portion of the HTTP request line, as a lowercase string.
`request_value` (optional)	`string`	Specifies the value (typically a resource path) portion of the HTTP request line.
`request_version` (optional)	`string`	Specifies the HTTP version portion of the HTTP request line, as a lowercase string.
`request_header` (optional)	`dictionary`	Specifies all of the HTTP header fields that may be found in the HTTP client request, as a dictionary. Each key in the dictionary **MUST** be the name of the header field as a lowercase string, e.g., “user-agent”. The value for each key **MUST** be the header field value for the corresponding header field.
`message_body_length` (optional)	`integer`	Specifies the length of the HTTP message body, if included, in bytes.
`message_body_data_ref` (optional)	`object-ref`	Specifies the data contained in the HTTP message body, if included. The object referenced in this field **MUST** be of type `artifact-object`.

8.3.1.2.2. Example

```
{
  "type":"cybox-container",
  "spec_version":"3.0",
  "objects":{
    "0":{
      "type":"ipv4-address-object",
      "value":"65.208.228.223"
    },
    "1":{
      "type":"network-connection-object",
      "dst_refs":[
        "1"
      ],
      "protocols":["tcp", "http"],
      "extended_properties":{
        "http":{
          "http_request_line":{
            "http_method":"get",
            "value":"/download.html",
            "version":"http/1.1"
          },
          "http_request_header":{
            "accept-encoding":"gzip,deflate",
            "user-agent":"Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.6)
Gecko/20040113",
            "host":"www.ethereal.com"
          }
        }
      }
    }
  }
}
```

8.3.2. TCP Extension

Type Name: `tcp-extension`

The TCP extension specifies a default extension for capturing network connection properties specific to TCP. The key for this extension when used in the **`extended_properties`** dictionary MUST be *tcp*.

8.3.2.1. Properties

Property Name	Type	Description

src_flags (optional)	`hex`	Specifies the source TCP flags.
dst_flags (optional)	`hex`	Specifies the destination TCP flags.

8.3.2.2. Example

```
{
  "type":"cybox-container",
  "spec_version":"3.0",
  "objects":{
    "0":{
      "type":"ipv4-address-object",
      "value":"1.2.3.4"
    },
    "1":{
      "type":"ipv4-address-object",
      "value":"2.3.4.5"
    },
    "2":{
      "type":"network-connection-object",
      "src_refs":[
        "0"
      ],
      "dst_refs":[
        "1"
      ],
      "protocols":["tcp"],
      "extended_properties":{
        "tcp":{
          "src_port":3372,
          "dst_port":80,
          "src_flags":"\\x00\\x00\\x00\\x02"
        }
      }
    }
  }
}
```

8.3.3. ICMP Extension

Type Name: `icmp-extension`	**Status:** Review **MVP**: Yes

The ICMP extension specifies a default extension for capturing network connection properties specific to ICMP. The key for this extension when used in the **`extended_properties`** dictionary MUST be *icmp*.

8.3.3.1. Properties

Property Name	Type	Description
icmp_type (required)	`hex`	Specifies the ICMP type byte.
icmp_code (required)	`hex`	Specifies the ICMP code byte.

8.3.3.2. Example

```
{
  "type":"cybox-container",
  "spec_version":"3.0",
  "objects":{
    "0":{
      "type":"ipv4-address-object",
      "value":"192.168.43.9"
    },
    "1":{
      "type":"ipv4-address-object",
      "value":"8.8.8.8"
    },
    "2":{
      "type":"network-connection-object",
      "src_refs":[
        "0"
      ],
      "dst_refs":[
        "1"
      ],
      "protocols":["icmp"],
      "extended_properties":{
        "icmp":{
          "icmp_type":"\\x08",
          "icmp_code":"\\x00"
        }
      }
    }
  }
}
```

8.3.4. Network Socket Extension

Type Name: `network-socket-extension`	**Status:** Review **MVP**: Yes

The Network Socket extension specifies a default extension for capturing network connection properties associated with network sockets. The key for this extension when used in the **`extended_properties`** dictionary **MUST** be *socket*.

Note that port numbers for bound socket addresses may be specified through either the TCP (for sockets of socket_type `sock_stream`) or UDP (for sockets of socket_type `sock_dgram`) Network Connection Object extensions.

8.3.4.1. Properties

Property Name	Type	Description
address_family (required)	`controlled-vocab`	Specifies the address family (AF_*) that the socket is configured for. This is a controlled vocabulary and values **MUST** come from the `socket-address-family-cv` vocabulary.
is_blocking (optional)	`boolean`	Specifies whether the socket is in blocking mode.
is_listening (optional)	`boolean`	Specifies whether the socket is in listening mode.
protocol_family (optional)	`controlled-vocab`	Specifies the protocol family (PF_*) that the socket is configured for. This is a controlled vocabulary and values **MUST** come from the `socket-protocol-family-cv` vocabulary.
options (optional)	`dictionary`	Specifies any options (SO_*) that may be used by the socket, as a dictionary. Each key in the dictionary **MUST** be a lowercase version of the option name, e.g., “ip_tos”. Each key value in the dictionary **MUST** be the value set for the corresponding options key.
socket_type (optional)	`controlled-vocab`	Specifies the type of the socket. This is a controlled

		vocabulary and values **MUST** come from the `network-socket-type-cv` vocabulary.
socket_descriptor (optional)	`integer`	Specifies the socket file descriptor value associated with the socket, as a non-negative integer.
socket_handle (optional)	`integer`	Specifies the handle or inode value associated with the socket.

8.3.4.2. Vocabularies

8.3.4.2.1. Network Socket Address Family Vocabulary

Type Name: `network-socket-address-family-cv`

Status: Review

MVP: Yes

A controlled vocabulary of network socket address family types.

Vocabulary Value	Description
`af_unspec`	Specifies an unspecified address family.
`af_inet`	Specifies the IPv4 address family.
`af_ipx`	Specifies the IPX (Novell Internet Protocol) address family.
`af_appletalk`	Specifies the APPLETALK DDP address family.
`af_netbios`	Specifies the NETBIOS address family.
`af_inet6`	Specifies the IPv6 address family.
`af_irda`	Specifies IRDA sockets.
`af_bth`	Specifies BTH sockets.

8.3.4.2.2. Network Socket Protocol Family Vocabulary

Type Name:	Status: Review
`network-socket-protocol-family-cv`	MVP: Yes

A controlled vocabulary of network socket protocol family types.

Vocabulary Value	Description
`pf_inet`	Specifies the IP protocol family.
`pf_ax25`	Specifies the amateur radio AX.25 family.
`pf_ipx`	Specifies the Novell Internet Protocol family.
`pf_inet6`	Specifies the IP version 6 family.
`pf_appletalk`	Specifies the Appletalk DDP protocol family.
`pf_netrom`	Specifies the Amateur radio NetROM protocol family.
`pf_bridge`	Specifies the Multiprotocol bridge protocol family.
`pf_atmpvc`	Specifies the ATM PVCs protocol family.
`pf_x25`	Specifies the protocol family reserved for the X.25 project.
`pf_rose`	Specifies the PF_KEY key management API family.
`pf_decnet`	Specifies the protocol family reserved for the DECnet project.
`pf_netbeui`	Specifies the protocol family reserved for the 802.2LLC project.
`pf_security`	Specifies the Security callback pseudo AF protocol family.
`pf_key`	Specifies the PF_KEY key management API protocol family.
`pf_netlink`	Specifies the netlink routing API family.
`pf_route`	Specifies the PF_ROUTE routing API family.
`pf_packet`	Specifies the packet family.
`pf_ash`	Specifies the Ash family.
`pf_econet`	Specifies the Acorn Econet family.
`pf_atmsvc`	Specifies the ATM SVCs protocol family.

`pf_sna`	Specifies the Linux SNA Project protocol family.
`pf_irda`	Specifies IRDA sockets.
`pf_pppox`	Specifies PPPoX sockets.
`pf_wanpipe`	Specifies Wanpipe API sockets.
`pf_bluetooth`	Specifies Bluetooth sockets.

8.3.4.2.3. Network Socket Type Vocabulary

Type Name: `network-socket-type-cv`	Status: Review MVP: Yes

A controlled vocabulary of network socket types.

Vocabulary Value	Description
`sock_stream`	Specifies a pipe-like socket which operates over a connection with a particular remote socket, and transmits data reliably as a stream of bytes.
`sock_dgram`	Specifies a socket in which individually-addressed packets are sent (datagram).
`sock_raw`	Specifies raw sockets which allow new IP protocols to be implemented in user space. A raw socket receives or sends the raw datagram not including link level headers.
`sock_rdm`	Specifies a socket indicating a reliably-delivered message.
`sock_seqpacket`	Specifies a datagram congestion control Protocol socket.

8.3.4.3. Example

```
{
  "type":"cybox-container",
  "spec_version":"3.0",
  "objects":{
    "0":{
      "type":"ipv4-address-object",
      "value":"192.168.1.2"
    },
    "1":{
```

```
      "type":"network-connection-object",
      "src_refs":[
        "0"
      ],
      "protocols":{
        "layer4":{
          "value":"tcp"
        }
      },
      "extended_properties":{
        "tcp":{
          "src_port":223
        },
        "socket":{
          "is_listening":"true",
          "address_family":"af_inet",
          "protocol_family":"pf_inet",
          "socket_type":"sock_stream"
        }
      }
    }
  }
}
```

9. Email Message Object

Type Name: `email-message-object`	Status: Review MVP: Yes

The Email Message Object represents an instance of an email message, corresponding to the internet message format described in RFC 5322 and related RFCs.

9.1. Properties

Property Name	Type	Description
`type` (required)	`string`	The value of this field **MUST** be `email-message-object`.

`is_multipart` (required)	`boolean`	Indicates whether the email body contains multiple MIME parts.
received_lines (optional)	`list` of type `string`	Specifies one or more *Received* header fields that may be included in the email headers. List values **MUST** appear in the same order as present in the email message.
date (optional)	`timestamp`	Specifies the date/time that the email message was sent.
content_type (optional)	`string`	Specifies the value of the “Content-Type” header of the email message.
from_ref (optional)	`object-ref`	Specifies the value of the “From:” header of the email message. The "From:" field specifies the author(s) of the message, that is, the mailbox(es) of the person(s) or system(s) responsible for the writing of the message. The object referenced in this field **MUST** be of type `email-address-object`.
sender_ref (optional)	`object-ref`	Specifies the value of the “From” field of the email message. The "Sender:" field specifies the mailbox of the agent responsible for the actual transmission of the message. The object referenced in this field **MUST** be of type `email-address-object`.
to_refs (optional)	`list` of type `object-ref`	Specifies the mailboxes that are “To:” recipients of the email message. The objects referenced in this list **MUST** be of type `email-address-object`.
cc_refs (optional)	`list` of type `object-ref`	Specifies the mailboxes that are “CC:” recipients of the email message. The objects referenced in this list **MUST** be of type `email-address-object`.

bcc_refs (optional)	`list` of type `object-ref`	Specifies the mailboxes that are "BCC:" recipients of the email message. As per RFC 5322, this list may be empty, which should not be treated the same as the key being absent. The objects referenced in this list **MUST** be of type `email-address-object`.
subject (optional)	`string`	Specifies the subject of the email message.
`other_header_fields` (optional)	`dictionary`	Specifies any other header fields (except for **date**, **received_lines**, **content_type**, **from_ref**, **sender_ref**, **to_refs**, **cc_refs**, **bcc_refs**, and **subject**) found in the email message, as a dictionary. Each key/value pair in the dictionary represents the name/value of a single header field. Accordingly, each dictionary key **MUST** be a lowercase string version of the header field name, with dashes ("-") replaced with underscores ("_"), e.g., "x_mailer". The corresponding value for each dictionary key **MUST** be a `string`.

body (optional)	`string`
	or
	`list` of type `mime-part-type`
		If the **`is_multipart`** boolean is true, specifies a list of the MIME parts. Otherwise, this is a `string` containing the email body

raw_email_ref (optional)	`object-ref`	Specifies the raw binary contents of the email message, including both the headers and body, as a reference to an Artifact Object. The object referenced in this field **MUST** be of type `artifact-object`.
extended_properties (optional)	`dictionary`	Specifies any extended properties of the object, as a dictionary.

		Dictionary keys **MUST** identify the extension type by name. The corresponding dictionary values **MUST** contain the contents of the extension instance.

9.2. Types

9.2.2. Email MIME Component Type (mime-part-type)

Specifies a component of a multi-part email body.

9.2.2.1. Properties

Property Name	Type	Description

body (required)	`string`
	or
	`object-ref`
		Specifies the contents of the MIME part. If the **content_type** is not provided **OR** starts with `text/` (e.g., in the case of plain text or HTML email) then a `string` **MUST** be used. Otherwise, it **MUST** be an `object-ref` that references a CybOX Artifact object representing the data contained within the MIME part.

content_type (optional)	`string`	Specifies the value of the “Content-Type” header of the MIME part.
content_disposition (optional)	`string`	Specifies the value of the “Content-Disposition” header of the MIME part.

9.3. Examples

9.3.0.1. Simple message

```
{
  "type":"cybox-container",
```

```
  "spec_version":"3.0",
  "objects":{
    "0":{
      "type":"email-address-object",
      "value":"jdoe@machine.example",
      "display_name":"John Doe"
    },
    "1":{
      "type":"email-address-object",
      "value":"mary@example.net",
      "display_name":"Mary Smith"
    },
    "2":{
      "type":"email-message-object",
      "from_ref":"0",
      "to_refs":["1"],
      "date":"1997-11-21T15:55:06Z",
      "subject":"Saying Hello"
    }
  }
}
```

9.3.0.2. Complex MIME Message

```
{
  "type": "cybox-container",
  "spec_version": "3.0",
  "objects": {
    "0": {
      "type": "email-message-object",
      "is_multipart": true,
      "received_lines": [
          "from mail.wayneindustries.com ([1.2.3.4])
            by smtp.gmail.com with ESMTPSA id q23sm23309939wme.17.2016.07.19.07.20.32
            (version=TLS1_2 cipher=ECDHE-RSA-AES128-GCM-SHA256 bits=128/128);
            Tue, 19 Jul 2016 07:20:40 -0700 (PDT)"],
      "content_type": "multipart/mixed",
      "date": "2016-06-19T14:20:40Z",
      "from_ref": "2",
      "to_refs": ["3"],
      "cc_refs": ["4"],
      "subject": "Check out this picture of the Riddler!"
      "other_header_fields":{
                            "content_disposition": "inline",
                            "x_mailer": "Mutt/1.5.23",
                            "x_originating_ip": "1"
       }
      "body": {
        [
```

```
          {"content_type": "text/plain; charset=utf-8",
            "content_disposition": "inline",
            “body”: "The Riddler is such a funny guy!"},
          {"content_type": "image/png",
            "content_disposition": "attachment; filename=\"riddler.png\"",
            "body": "5"
            }
          ]
    }
    "1": {
      "type": "ipv4-address-object",
      "value": "1.2.3.4"},
    "2": {
      "type": "email-address-object",
      "value": "bwayne@wayneindustries.com",
      "display_name": "Bruce Wayne"},
    "3": {
      "type": "email-address-object",
      "value": "robin@batcave.com",
      "display_name": "Robin"},
    "4": {
      "type": "email-address-object",
      "value": "apennyworth@wayneindustries.com",
      "display_name": "Alfred Pennyworth"},
    "5": {
      "type": "artifact-object",
      "mime_type": "image/jpeg",
      "payload": "VBORw0KGgoAAAANSUhEUgAAADI== ..."
      }
    }
  }
}
```