NAME

ranonymize.conf – **ranonymize(1)** configuration file.

SYNOPSIS

ranonymize.conf

DESCRIPTION

This configuration file provides the ability to specify options for argus data anoymization.

OPTIONS

The anonymization clients have a small number of options for controlling specific aspects of the anonymization function and its output.

Timestamps, Reference and Sequence Numbers

Ranonymize anonymizes various fields in Argus records, such as the network addresses, protocol specific port numbers, timestamps, transaction reference numbers, and the sequence numbers.

For some fields, specifically the timestamps, transaction reference numbers and the sequence numbers, which are generally monotonically increasing counters, a good anonymization technique is to shift the values by a constant, so that the sequential relationships between values is preserved.

The configuration provides some flexibility here, so that the user can control fixed offset shifting anonymization. The constant value can be generated by the anonymization client at "random", which is the default behavior, or the user can provide a "fixed:x", where x is the fixed offset. Of course, the keyword "none" can be used to turn off the default anonymization for these values.

RANON_TRANSREFNUM_OFFSET=random
RANON_SEQNUM_OFFSET=random
RANON_TIME_SEC_OFFSET=random
RANON_TIME_USEC_OFFSET=random

Ethernet Address Vendor Codes

When anonymizing ethernet addresses, **ranonymize** has the option to preserve the vendor portion, if desired. This allows analytical programs to differentiate anonymized data by vendor type. This feature is turned off by default.

RANON_PRESERVE_ETHERNET_VENDOR=no

Broadcast Addresses

Ranonymize has the option to preserve the semantic that an address is a broadcast address. This is very important when doing flow analysis for either operational or performance managment tasks, using anonymized data.

RANON_PRESERVE_BROADCAST_ADDRESS=yes

IPv4 Address Anonymization

IPv4 address are composed of two parts, a network part and a host part. Because the addressing strategy of a site may have integrated semantics that would want to be retained in the anonymized addresses, IPv4 address anonymization involves specifying a one-to-one translation table for both the network and host address spaces in an IPv4 address. Once a new network address has been allocated, every occurence of that network address will be substituted in the anonymizers output stream. The host address space is anonymized in an independent but similar fashion.

Ranonymize allows you to specify the type of anonymization method used in a number of categories. For network and host address conversion, ranonymize can support "sequential", "random" or "no" anonymization. Sequential anonymization involves allocating new addresses in a monotonically increasing fashion on a first come first serve basis. Random anonymization allocates random addresses from the working pool of addresses, and "no" anonymization preserves the address type, whether its network, host or both.

The default working pool of network addresses contains only non-routable addresses, and starts with 10.0.0.0. All anonymized addresses are treated as Class C network addresses, in order to conserve the anonymization allocation demands.

As an example, if the first Argus record contained the addresses 128.64.2.4 and 132.243.2.87 as the source and destination, sequential anonymization would generate the addresses 10.0.0.1 and 10.0.1.1 as the new source and destination addresses. This is because, the two addresses have differing network parts, 128.64.2 and 132.243.2, these would be allocated 10.0.0 and 10.0.1 respectively (sequential allocation). Because these are the first hosts to be allocated, the host parts are both 1.

Random anonymization could generate 10.24.31.203 and 10.1.34.18 as possible addresses, as both the Class C network address would be allocated randomly from the 10 network space, and the host address part would be allocated randomly from the possible host addresses.

Sequential randomization uses the least amount of memory and minimizes anonymization processing time, while random provides better address scrambling.

Implemenation note: currently only supporting sequential

RANON_NET_ANONYMIZATION=sequential
RANON_HOST_ANONYMIZATION=sequential

Address Hierarchy

Ranonymize has the option to preserve the network address hierarchy at various levels of granularity. This allows you to preserve the addressing relationships between addresses. The options are "cidr", "class", "subnet" and "no".

Class network adddress heirarchy preservation, causes ranonymize() to allocate new network addresses base on the address class. All CLASSA network addresses will be allocated new addresses from the Class A network pool. Network addresses will be allocated as 24 bit CIDR addresses, in that the first 24 bits will map to a unique 24 network address, and host addresses will be allocated from the 254 address pool (0 and 255 can be preserved, see below).

RANON_PRESERVE_NET_ADDRESS_HIERARCHY=cidr

Specific Network Address Aliasing

Ranonymize can be configured to perform specific network address translation. These must be specified as 24 bit CIDR addresses. RANON_PRESERVE_NET_ADDRESS_HIERARCHY must be set to "cidr", for this feature to work.

Examples would be:

RANON_SPECIFY_NET_TRANSLATION=192.168.0.0::128.2.134.0
RANON_SPECIFY_NET_TRANSLATION=64.12.0.0::134.5.0.0
RANON_SPECIFY_NET_TRANSLATION=128.2.0.0::200.200.0.0

Specific Host Address Aliasing

Ranonymize can be configured to perform specific host address translation. These addresses are allocated prior to reading any data, and are removed from the potential network address pool, regardless of the anonymization strategy. Feel free to list as many addresses that you would like.

Examples would be:

RANON_SPECIFY_HOST_TRANSLATION=192.168.0.64::128.2.34.5

Transport SAP Aliasing

Ranonymize can be configured to preserve specific ranges of port numbers. For convenience, ranonymize() can be configured to preserve the IANA well known port allocation range (0-1023), the registered ports (1024-49151) and/or the private port range (49152 - 65535). Also, ranonymize() can be configured to preserve specific port numbers. These numbers are independent of protocol type, so if port 23461 is to be preserved, it will be preserved for both tcp and udp based flows.

RANON_PRESERVE_WELLKNOWN_PORT_NUMS=yes
RANON_PRESERVE_REGISTERED_PORT_NUMS=no
RANON_PRESERVE_PRIVATE_PORT_NUMS=no

COPYRIGHT

SEE ALSO

ranonymize(1)