# Port Scan Inspector



# Port Scan Inspector Overview

| | |
|---|---|
| Type | Inspector (probe) |
| Usage | Global |
| Instance Type | Global |
| Other Inspectors Required | None |
| Enabled | false |

A port scan is a form of network reconnaissance that is often used by attackers as a prelude to an attack. In a port scan, an attacker sends packets designed to probe for network protocols and services on a targeted host. By examining the packets sent in response by a host, the attacker can determine which ports are open on the host and, either directly or by inference, which application protocols are running on these ports.

By itself, a port scan is not evidence of an attack. Legitimate users on your network may employ similar port scanning techniques used by attackers.

The `port_scan` inspector detects four types of portscan and monitors connection attempts on TCP, UDP, ICMP, and IP protocols. By detecting patterns of activity, the `port_scan` inspector helps you determine which port scans might be malicious.

***Table 1: Portscan Protocol Types***

| Protocol | Description |
|---|---|
| TCP | Detects TCP probes such as SYN scans, ACK scans, TCP connect() scans, and scans with unusual flag combinations such (Xmas tree, FIN, and NULL). |
| UDP | Detects UDP probes such as zero-byte UDP packets. |

| Protocol | Description |
|---|---|
| ICMP | Detects ICMP echo requests (pings). |
| IP | Detects IP protocol scans. Instead of looking for open ports, Snort searches for IP protocols which are supported on a target host. |

Port scans are generally divided into four types based on the number of targeted hosts, the number of scanning hosts, and the number of ports that are scanned.

***Table 2: Portscan Types***

| Type | Description |
|---|---|
| Portscan | A one-to-one port scan in which an attacker uses one or a few hosts to scan multiple ports on a single target host.<br>One-to-one port scans are characterized by:<br>• a low number of scanning hosts<br>• a single host that is scanned<br>• a high number of ports scanned<br>A portscan detects TCP, UDP, and IP port scans. |
| Portsweep | A one-to-many port sweep in which an attacker uses one or a few hosts to scan a single port on multiple target hosts.<br>Port sweeps are characterized by:<br>• a low number of scanning hosts<br>• a high number of scanned hosts<br>• a low number of unique ports scanned<br>A portsweep detects TCP, UDP, ICMP, and IP port sweeps. |
| Decoy Portscan | A one-to-one port scan in which the attacker mixes spoofed source IP addresses with the actual scanning IP address.<br>Decoy port scans are characterized by:<br>• a high number of scanning hosts<br>• a low number of ports that are scanned only once<br>• a single (or a low number of) scanned hosts<br>The decoy port scan detects TCP, UDP, and IP protocol port scans. |

| Type | Description |
|---|---|
| Distributed Portscan | A many-to-one port scan in which multiple hosts query a single host for open ports.<br>Distributed port scans are characterized by:<br>• a high number of scanning hosts<br>• a high number of ports that are scanned only once<br>• a single (or a low number of) scanned hosts<br>The distributed portscan detects TCP, UDP, and IP protocol port scans. |

### Port Scan Sensitivity Levels

The `port_scan` inspector provides three default scan sensitivity levels.

- default_low_port_scan
- default_med_port_scan
- default_high_port_scan

You can configure additional scan sensitivity levels with various filters:

- scans
- rejects
- nets
- ports

The `port_scan` inspector learns about a probe by gathering negative responses from the probed hosts. For example, when a web client uses TCP to connect to a web server, the client can assume that the web server listens on port 80. However, when an attacker probes a server, the attacker does not know in advance if the server offers web services. When the `port_scan` inspector detects a negative response (ICMP unreachable or TCP RST packet), it records the response as a potential portscan. The process is more difficult when the targeted host is on the other side of a device such as a firewall or router that filters negative responses. In this case, the `port_scan` inspector can generate filtered portscan events based on the sensitivity level that you select.

# Best Practices for Configuring the Port Scan Inspector

To optimize the detection of port scans, we recommend that you tune the `port_scan` inspector to match your networks.

- Ensure that you carefully configure the `watch_ip` parameter. The `watch_ip` parameter helps the `port_scan` inspector filter legitimate hosts that are very active on your network. Some of the most common examples are NAT IPs, DNS cache servers, syslog servers, and nfs servers.
- Most of the false positives that the `port_scan` inspector may generate are of the filtered scan `alert` type. The `alert` type may indicate that a host was overly active during a specific time period. If the host

continually generates the filtered scan `alert` type, add the host to the `ignore_scanners` list or use a lower scan sensitivity level.

- Make use of the Priority Count, Connection Count, IP Count, Port Count, IP range, and Port range to determine false positives. The easiest way to determine false positives is through simple ratio estimations. The following is a list of ratios to estimate and the associated values that indicate a legitimate scan as opposed to a false positive.
  - Connection Count / IP Count - This ratio indicates an estimated average of connections per IP. For port scans, this ratio should be high. For port sweeps, this ratio should be low.
  - Port Count / IP Count - This ratio indicates an estimated average of ports connected to per IP. For port scans, this ratio should be high and indicates that the scanned host's ports were connected to by fewer IPs. For port sweeps, this ratio should be low, indicating that the scanning host connected to few ports but on many hosts.
  - Connection Count / Port Count - This ratio indicates an estimated average of connections per port. For port scans, this ratio should be low. This indicates that each connection was to a different port. For port sweeps, this ratio should be high. This indicates that there were many connections to the same port.

  The higher the priority count, the more likely it is a real port scan or port sweep (unless the host is managed by a firewall).
- If you are unable to detect port scans, you can lower the scan sensitivity level. You get the best protection with a higher scan sensitivity level. The low scan sensitivity level only generates alerts based on error responses and does not catch filtered scans. The low scan sensitivity level error responses can indicate a port scan, and the alerts generated by the low sensitivity level are highly accurate and require the least tuning. Filtered or high sensitivity level scans are prone to false positives.

# Port Scan Inspector Parameters

### memcap

Specifies the maximum tracker memory in bytes.

**Type:** integer

**Valid range:** `1024` to `9,007,199,254,740,992 (maxSZ)`

**Default value:** `10,485,760`

### protos

Specifies the protocols to monitor. Provide a string of protocol abbreviations. To specify multiple protocols, separate each protocol abbreviation with a space.

**Type:** string

**Valid values:** `tcp`, `udp`, `icmp`, `ip`, `all`

**Default value:** `all`

#### scan_types

Specifies the types of port scan to examine. Provide a string of protocol abbreviations. To specify multiple protocols, separate each protocol string with a space.

**Type:** string

**Valid values:** `portscan`, `portsweep`, `decoy_portscan`, `distributed_portscan`, `all`

**Default value:** `all`

#### watch_ip

Specifies a list of CIDR blocks and IPs with optional ports to watch.

If `watch_ip` is not defined, the `port_scan` inspector examines all network traffic.

**Type:** string

**Valid values:** CIDR or IP address, list of CIDR or IP addresses

**Default value:** None

#### alert_all

Specifies whether to alert on all events over the threshold within the established window. If `alert_all` is set to `false`, the `port_scan` inspector only alerts on the first event over the threshold within the window.

**Type:** boolean

**Valid values:** `true`, `false`

**Default value:** `false`

#### include_midstream

Specifies whether to list CIDRs with optional ports.

**Type:** boolean

**Valid values:** `true`, `false`

**Default value:** `false`

#### tcp_decoy.rejects

Specifies the number of scan attempts with negative responses.

**Type:** integer

**Valid range:** `0` to `65535`

**Default value:** `15`

#### tcp_decoy.ports

Specifies the number of times the port (or protocol) changed from a prior attempt.

**Type:** integer

**Valid range:** `0` to `65535`

**Default value:** `25`

### tcp_decoy.scan

Specifies the number of scan attempts.

**Type:** integer

**Valid range:** `0` to `65535`

**Default value:** `100`

### tcp_decoy.nets

Specifies the number of times the address changed from prior attempts.

**Type:** integer

**Valid range:** `0` to `65535`

**Default value:** `25`

### tcp_dist.rejects

Specifies the number of scan attempts with negative responses.

**Type:** integer

**Valid range:** `0` to `65535`

**Default value:** `15`

### tcp_dist.ports

Specifies the number of times the port (or protocol) changed from a prior attempt.

**Type:** integer

**Valid range:** `0` to `65535`

**Default value:** `25`

### tcp_dist.scans

Specifies the number of scan attempts.

**Type:** integer

**Valid range:** `0` to `65535`

**Default value:** `100`

### tcp_dist.nets

Specifies the number of times the address changed from prior attempts.

**Type:** integer

**Valid range:** `0` to `65535`

**Default value:** `25`

#### tcp_ports.rejects

Specifies the number of scan attempts with negative responses.

**Type:** integer

**Valid range:** `0` to `65535`

**Default value:** `15`

#### tcp_ports.ports

Specifies the number of times the port (or protocol) changed from a prior attempt.

**Type:** integer

**Valid range:** `0` to `65535`

**Default value:** `25`

#### tcp_ports.scans

Specifies the number of scan attempts.

**Type:** integer

**Valid range:** `0` to `65535`

**Default value:** `100`

#### tcp_ports.nets

Specifies the number of times the address changed from prior attempts.

**Type:** integer

**Valid range:** `0` to `65535`

**Default value:** `25`

#### tcp_sweep.rejects

Specifies the number of scan attempts with negative responses.

**Type:** integer

**Valid range:** `0` to `65535`

**Default value:** `15`

#### tcp_sweep.ports

Specifies the number of times the port (or protocol) changed from a prior attempt.

**Type:** integer

**Valid range:** `0` to `65535`

**Default value:** `25`

#### tcp_sweep.scans

Specifies the number of scan attempts.

**Type:** integer

**Valid range:** `0` to `65535`

**Default value:** `100`

#### tcp_sweep.nets

Specifies the number of times the address changed from prior attempts.

**Type:** integer

**Valid range:** `0` to `65535`

**Default value:** `25`

#### udp_decoy.rejects

Specifies the number of scan attempts with negative responses.

**Type:** integer

**Valid range:** `0` to `65535`

**Default value:** `15`

#### udp_decoy.ports

Specifies the number of times the port (or protocol) changed from a prior attempt.

**Type:** integer

**Valid range:** `0` to `65535`

**Default value:** `25`

#### udp_decoy.scans

Specifies the of number scan attempts.

**Type:** integer

**Valid range:** `0` to `65535`

**Default value:** `100`

#### udp_decoy.nets

Specifies the number of times the address changed from prior attempts.

**Type:** integer

**Valid range:** `0` to `65535`

**Default value:** `25`

#### udp_dist.rejects

Specifies the number of scan attempts with negative responses.

**Type:** integer

**Valid range:** `0` to `65535`

**Default value:** `15`

#### udp_dist.ports

Specifies the number of times the port (or protocol) changed from a prior attempt.

**Type:** integer

**Valid range:** `0` to `65535`

**Default value:** `25`

#### udp_dist.scans

Specifies the number of scan attempts.

**Type:** integer

**Valid range:** `0` to `65535`

**Default value:** `100`

#### udp_dist.nets

Specifies the number of times the address changed from prior attempts.

**Type:** integer

**Valid range:** `0` to `65535`

**Default value:** `25`

#### udp_ports.rejects

Specifies the number of scan attempts with negative responses.

**Type:** integer

**Valid range:** `0` to `65535`

**Default value:** `15`

#### udp_ports.ports

Specifies the number of times the port (or protocol) changed from a prior attempt.

**Type:** integer

**Valid range:** `0` to `65535`

**Default value:** `25`

#### udp_ports.scans

Specifies the number of scan attempts.

**Type:** integer

**Valid range:** `0` to `65535`

**Default value:** `100`

#### udp_ports.nets

Specifies the number of times the address changed from prior attempts.

**Type:** integer

**Valid range:** `0` to `65535`

**Default value:** `25`

#### udp_sweep.rejects

Specifies the number of scan attempts with negative responses.

**Type:** integer

**Valid range:** `0` to `65535`

**Default value:** `15`

#### udp_sweep.ports

Specifies the number of times the port (or protocol) changed from a prior attempt.

**Type:** integer

**Valid range:** `0` to `65535`

**Default value:** `25`

#### udp_sweep.scans

Specifies the number of scan attempts.

**Type:** integer

**Valid range:** `0` to `65535`

**Default value:** `100`

#### udp_sweep.nets

Specifies the number of times the address changed from prior attempts.

**Type:** integer

**Valid range:** `0` to `65535`

**Default value:** `25`

### ip_decoy.rejects

Specifies the number of scan attempts with negative responses.

**Type:** integer

**Valid range:** `0` to `65535`

**Default value:** `15`

### ip_decoy.ports

Specifies the number of times the port (or protocol) changed from a prior attempt.

**Type:** integer

**Valid range:** `0` to `65535`

**Default value:** `25`

### ip_decoy.scans

Specifies the number of scan attempts.

**Type:** integer

**Valid range:** `0` to `65535`

**Default value:** `100`

### ip_decoy.nets

Specifies the number of times the address changed from prior attempts.

**Type:** integer

**Valid range:** `0` to `65535`

**Default value:** `25`

### ip_dist.rejects

Specifies the number of scan attempts with negative responses.

**Type:** integer

**Valid range:** `0` to `65535`

**Default value:** `15`

### ip_dist.ports

Specifies the number of times the port (or protocol) changed from a prior attempt.

**Type:** integer

**Valid range:** `0` to `65535`

**Default value:** `25`

### ip_dist.scans

Specifies the number of scan attempts.

**Type:** integer

**Valid range:** `0` to `65535`

**Default value:** `100`

### ip_dist.nets

Specifies the number of times the address changed from prior attempts.

**Type:** integer

**Valid range:** `0` to `65535`

**Default value:** `25`

### ip_sweep.rejects

Specifies the number of scan attempts with negative responses.

**Type:** integer

**Valid range:** `0` to `65535`

**Default value:** `15`

### ip_sweep.ports

Specifies the number of times the port (or protocol) changed from a prior attempt.

**Type:** integer

**Valid range:** `0` to `65535`

**Default value:** `25`

### ip_sweep.scans

Specifies the of number scan attempts.

**Type:** integer

**Valid range:** `0` to `65535`

**Default value:** `100`

### ip_sweep.nets

Specifies the number of times the address changed from prior attempts.

**Type:** integer

**Valid range:** `0` to `65535`

**Default value:** `25`

#### ip_proto.rejects

Specifies the number of scan attempts with negative responses.

**Type:** integer

**Valid range:** `0` to `65535`

**Default value:** `15`

#### ip_proto.ports

Specifies the number of times the port (or protocol) changed from a prior attempt.

**Type:** integer

**Valid range:** `0` to `65535`

**Default value:** `25`

#### ip_proto.scans

Specifies the number of scan attempts.

**Type:** integer

**Valid range:** `0` to `65535`

**Default value:** `100`

#### ip_proto.nets

Specifies the number of times the address changed from prior attempts.

**Type:** integer

**Valid range:** `0` to `65535`

**Default value:** `25`

#### icmp_sweep.rejects

Specifies the number of scan attempts with negative responses.

**Type:** integer

**Valid range:** `0` to `65535`

**Default value:** `15`

#### icmp_sweep.ports

Specifies the number of times the port (or protocol) changed from a prior attempt.

**Type:** integer

**Valid range:** `0` to `65535`

**Default value:** `25`

### icmp_sweep.scans

Specifies the number of scan attempts.

**Type:** integer

**Valid range:** `0` to `65535`

**Default value:** `100`

### icmp_sweep.nets

Specifies the number of times the address changed from prior attempts.

**Type:** integer

**Valid range:** `0` to `65535`

**Default value:** `25`

### tcp_window

Specifies the detection interval for transmission control protocol (TCP) scans.

**Type:** integer

**Valid range:** `0` to `4,294,967,295 (max32)`

**Default value:** `0`

### udp_window

Specifies the detection interval for user datagram protocol (UDP) scans.

**Type:** integer

**Valid range:** `0` to `4,294,967,295 (max32)`

**Default value:** `0`

### ip_window

Specifies the detection interval for internet protocol (IP) scans.

**Type:** integer

**Valid range:** `0` to `4,294,967,295 (max32)`

**Default value:** `0`

### icmp_window

Specifies the detection interval for internet control message protocol (ICMP) scans.

**Type:** integer

**Valid range:** `0` to `4,294,967,295 (max32)`

**Default value:** `0`

# Port Scan Inspector Rules

Enable the `port_scan` inspector rules to generate events and, in an inline deployment, drop offending packets.

***Table 3: Port Scan Inspector Rules***

| GID:SID | Rule Message |
|---|---|
| 122:1 | TCP portscan |
| 122:2 | TCP decoy portscan |
| 122:3 | TCP portsweep |
| 122:4 | TCP distributed portscan |
| 122:5 | TCP filtered portscan |
| 122:6 | TCP filtered decoy portscan |
| 122:7 | TCP filtered portsweep |
| 122:8 | TCP filtered distributed portscan |
| 122:9 | IP protocol scan |
| 122:10 | IP decoy protocol scan |
| 122:11 | IP protocol sweep |
| 122:12 | IP distributed protocol scan |
| 122:13 | IP filtered protocol scan |
| 122:14 | IP filtered decoy protocol scan |
| 122:15 | IP filtered protocol sweep |
| 122:16 | IP filtered distributed protocol scan |
| 122:17 | UDP portscan |
| 122:18 | UDP decoy portscan |
| 122:19 | UDP portsweep |
| 122:20 | UDP distributed portscan |
| 122:21 | UDP filtered portscan |
| 122:22 | UDP filtered decoy portscan |
| 122:23 | UDP filtered portsweep |
| 122:24 | UDP filtered distributed portscan |

| GID:SID | Rule Message |
|---|---|
| 122:25 | ICMP sweep |
| 122:26 | ICMP filtered sweep |
| 122:27 | open port |

# Port Scan Inspector Intrusion Rule Options

The `port_scan` inspector does not have any intrusion rule options.