SSH Inspector

- SSH Inspector Overview, on page 1
- Best Practices for Configuring the SSH Inspector, on page 2
- SSH Inspector Parameters, on page 2
- SSH Inspector Rules, on page 3
- SSH Inspector Intrusion Rule Options, on page 4

SSH Inspector Overview

Type	Inspector (service)
Usage	Inspect
Instance Type	Multiton
Other Inspectors Required	None
Enabled	`true`

Secure Shell Protocol (SSH) is network protocol that enables secure communication between a client and server over an unsecured network. SSH supports tunneling and authenticates a remote host using public-key cryptography.

You can use SSH to securely transfer files, or login into a remote host and interact with the command line. The SSH protocol uses port 22 over TCP, UDP, or SCTP.

The `ssh` inspector decodes stream packets and detects the following SSH exploits:

- Challenge-Response Buffer Overflow exploit
- CRC-32 exploit
- SecureCRT SSH Client Buffer Overflow exploit
- Incorrect SSH message direction

Challenge-Response Buffer Overflow and CRC-32 attacks occur after authentication when the network connection between hosts is encrypted. Both types of attack send a large payload of more than 20 KB to the server immediately after the authentication challenge.

The `ssh` inspector detects the Challenge-Response Buffer Overflow and CRC-32 attacks by counting the number of bytes transmitted to the server. If the bytes exceed the defined limit within a predefined number of packets, the `ssh` inspector generates an alert. CRC-32 attacks apply only to SSH Version 1 and Challenge-Response Buffer Overflow exploits apply only to SSH Version 2. The `ssh` inspector reads the SSH version string at the beginning of the session to identify the type of attack.

The SecureCRT SSH Client Buffer Overflow and protocol mismatch attacks occur before the key exchange when hosts are attempting to secure a connection. The SecureCRT SSH Client Buffer Overflow attack sends an overly long protocol identifier string to the client, causing a buffer overflow. A protocol mismatch attack occurs when either a non-SSH client application attempts to connect to a secure SSH server, or the server and client version numbers do not match.

Note The `ssh` inspector does not handle brute force attacks.

Best Practices for Configuring the SSH Inspector

We recommend that you use the default `ssh` inspector configuration settings. If you exceed the maximum number of encrypted packets for a session, defined in the `max_encrypted_packets` parameter, the `ssh` inspector stops processing traffic for that session to improve performance. The `ssh` inspector only detects SSH vulnerabilities that appear at the beginning of the SSH session.

Note If the `ssh` inspector generates a false positive on Challenge-Response Overflow or CRC 32, you can increase the number of required client bytes with the `max_client_bytes` parameter.

SSH Inspector Parameters

SSH service configuration

The `binder` inspector defines the SSH `service` configuration. For more information, see the Binder Inspector Overview.

Example:

```
[
    {
        "when": {
            "service": "ssh",
            "role": any
        },
        "use": {
            "type": "ssh"
        }
    }
]
```

max_encrypted_packets

Specifies the maximum number of encrypted packets to examine before the `ssh` inspector ignores an SSH session. If you exceed the maximum number of encrypted packets for a session, the `ssh` inspector stops processing traffic for that session to improve performance.

Type: integer

Valid range: `-1` to `65535`

Default value: `25`

max_client_bytes

Specifies the maximum number of unanswered bytes to transmit to the server before the `ssh` inspector alerts on Challenge-Response Overflow or CRC 32. If you exceed the `max_client_bytes` limit before `max_encrypted_packets` are sent, the inspector assumes an attack has occurred and ignores the traffic.

You can enable rule 128:1 to generate an alert when the inspector detects a Challenge-Response Overflow or rule 128:2 to generate an alert when the inspector detects a CRC 32 exploit.

For each valid response the client receives from the server, the `ssh` inspector resets the packet count for `max_client bytes`.

Note We do not recommend that you set `max_client_bytes` to `0` or `1`. If you set the `max_client_bytes` to `0` or `1`, the `ssh` inspector always alerts.

Type: integer

Valid range: `0` to `65535`

Default value: `19600`

max_server_version_len

Specifies the maximum length of the SSH server version string. If the length of the SSH server version string exceeds the `max_server_version_len`, the `ssh` inspector generates an alert. You can enable rule 128:3 to alert on the Secure CRT server version string overflow.

Type: integer

Valid range: `0` to `255`

Default value: `80`

Note The `ssh` inspector default configuration does not enable any alerts.

SSH Inspector Rules

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

Table 1: SSH Inspector Rules

GID:SID	Rule Message
128:1	challenge-response overflow exploit
128:2	SSH1 CRC32 exploit
128:3	server version string overflow
128:5	bad message direction
128:6	payload size incorrect for the given payload
128:7	failed to detect SSH version string

SSH Inspector Intrusion Rule Options

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