Tranalyzer2

sshDecode

Secure Shell (SSH)

Tranalyzer Development Team

Contents

1 sshDecode

1.1 Description

This plugin analyzes SSH traffic.

1.2 Dependencies

This plugin requires the **libssl**.

Ubuntu:	`sudo apt-get install`	`libssl-dev`
Arch:	`sudo pacman -S`	`openssl`
openSUSE:	`sudo zypper install`	`libopenssl-devel`
Red Hat/Fedora[1]**:**	`sudo dnf install`	`openssl-devel`
macOS[2]**:**	`brew install`	`openssl@1.1`

1.3 Configuration Flags

The following flags can be used to control the output of the plugin:

Name	Default	Description	Flags
`SSH_USE_PORT`	0	1: Count all packets to/from `SSH_PORT` as SSH (useful if version exchange was not captured)	
`SSH_DECODE`	2	0: Do not decode SSH handshake messages 1: Only decode SSH Key Exchange Init messages 2: Decode all SSH Exchange messages	
`SSH_FINGERPRINT`	1	Algorithm to use for the fingerprint: 0: No fingerprint, 1: MD5, 1: SHA256	`SSH_DECODE=2`
`SSH_ALGO`	1	Output chosen algorithms	`SSH_DECODE>0`
`SSH_LISTS`	0	Output lists of supported algorithms	`SSH_DECODE>0`
`SSH_HASSH`	1	Output HASSH fingerprint (hash and description)	
`SSH_HASSH_STR`	0	Also output HASSH fingerprint before hashing	`SSH_HASSH=1`
`SSH_HASSH_DLEN`	512	Max length for HASSH descriptions	`SSH_HASSH=1`
`SSH_HASSH_STR_LEN`	1024	Max length for uncompressed HASSH signatures	`SSH_HASSH=1`
`SSH_BUF_SIZE`	512	Max length for strings	
`SSH_HKT_SIZE`	48	Max length for host key type and chosen algorithms	
`SSH_DEBUG`	0	Activate debug output	

In addition, the name of the HASSH database is controlled by the `SSH_HASSH_NAME` flag and defaults to `"hassh_fingerprints.tsv"`.

[1] If the `dnf` command could not be found, try with `yum` instead

[2] Brew is a packet manager for macOS that can be found here: `https://brew.sh`

1.3.1 Environment Variable Configuration Flags

The following configuration flags can also be configured with environment variables (`ENVCNTRL>0`):

- `SSH_HASSH_NAME`

1.4 Flow File Output

The sshDecode plugin outputs the following columns:

Column	Type	Description	Flags
`sshStat`	H16	Status	
`sshVersion`	R(S)	SSH version and software	
`sshHostKeyType`	R(SC)	SSH host key type	`SSH_DECODE=2`

`sshFingerprint`	R(SC)	SSH public key fingerprint	`SSH_DECODE=2&&`
			`SSH_FINGERPRINT>0`

`sshCookie`	R(SC)	SSH cookie	`SSH_DECODE>0`
If `SSH_DECODE>0&&SSH_ALGO=1`, the following columns are displayed:			
`sshKEX`	R(S)	SSH chosen KEX algorithm	
`sshSrvHKeyAlgo`	R(S)	SSH chosen server host key algorithm	
`sshEncCS`	R(S)	SSH chosen encryption algorithm client to server	
`sshEncSC`	R(S)	SSH chosen encryption algorithm server to client	
`sshMacCS`	R(S)	SSH chosen MAC algorithm client to server	
`sshMacSC`	R(S)	SSH chosen MAC algorithm server to client	
`sshCompCS`	R(S)	SSH chosen compression algorithm client to server	
`sshCompSC`	R(S)	SSH chosen compression algorithm server to client	
`sshLangCS`	R(S)	SSH chosen language client to server	
`sshLangSC`	R(S)	SSH chosen language server to client	
If `SSH_DECODE>0&&SSH_LISTS=1`, the following columns are displayed:			
`sshKEXList`	R(S)	SSH KEX algorithms	
`sshSrvHKeyAlgoList`	R(S)	SSH server host key algorithms	
`sshEncCSList`	R(S)	SSH encryption algorithms client to server	
`sshEncSCList`	R(S)	SSH encryption algorithms server to client	
`sshMacCSList`	R(S)	SSH MAC algorithms client to server	
`sshMacSCList`	R(S)	SSH MAC algorithms server to client	
`sshCompCSList`	R(S)	SSH compression algorithms client to server	
`sshCompSCList`	R(S)	SSH compression algorithms server to client	
`sshLangCSList`	R(S)	SSH languages client to server	
`sshLangSCList`	R(S)	SSH languages server to client	
If `SSH_HASSH=1`, the following columns are displayed:			
`sshHassh`	R(SC)	SSH HASSH fingerprint	
`sshHasshDesc`	R(S)	SSH HASSH description	

Column	Type	Description	Flags
sshHasshStr	R(S)	SSH HASSH string	

1.4.1 sshStat

The sshStat column is to be interpreted as follows:

sshStat	Description
2^{0} (=0x0001)	Flow contains SSH protocol
2^{1} (=0x0002)	Keeps track of who sent the SSH banner first
2^{2} (=0x0004)	Banner does not end with CRLF or contains NULL byte
2^{3} (=0x0008)	Key Exchange Init message seen
2^{4} (=0x0010)	Diffie-Hellman Key Exchange Init message seen
2^{5} (=0x0020)	Diffie-Hellman Key Exchange Reply message seen
2^{6} (=0x0040)	Elliptic Curve Diffie-Hellman Key Exchange Init message seen
2^{7} (=0x0080)	Elliptic Curve Diffie-Hellman Key Exchange Reply message seen
2^{8} (=0x0100)	Diffie-Hellman Group Exchange Group message seen
2^{9} (=0x0200)	Diffie-Hellman Group Exchange Init message seen
2^{10} (=0x0400)	Diffie-Hellman Group Exchange Request message seen
2^{11} (=0x0800)	Diffie-Hellman Group Exchange Reply message seen
2^{12} (=0x1000)	New Keys message seen
2^{13} (=0x2000)	String truncated... increase SSH_BUF_SIZE
2^{14} (=0x4000)	Host key type or chosen algorithm truncated... increase SSH_HKT_SIZE
2^{15} (=0x8000)	Malformed (decoding error, encrypted, ...)

1.4.2 sshFingerprint

The fingerprint of a public key can be computed as follows:

```
ssh-keygen -lf id_rsa.pub
```

To compute the fingerprint of each host listed in *~/.ssh/known_hosts*, use the following command:

```
ssh-keygen -lf ~/.ssh/known_hosts
```

Note that the default SHA256 algorithm can be changed with the -E md5 option.

1.5 Packet File Output

In packet mode (-s option), the sshDecode plugin outputs the following columns:

Column	Type	Description	Flags
sshStat	H16	Status	

1.6 Monitoring Output

In monitoring mode, the sshDecode plugin outputs the following columns:

Column	Type	Description	Flags
sshNFlows	U64	Number of SSH flows	
sshStat	H16	Status	

1.7 Plugin Report Output

The following information is reported:

- Aggregated sshStat
- Number of SSH flows
- Number of HASSH signatures matched