



READY-MADE EVASION TEST LAB

EVADER

A STONESOFT INNOVATION

User's Guide

STONESOFT

USING STONESOFT EVADER

Stonesoft Evader allows you to test the effectiveness of security devices in your network environment against advanced evasion techniques.

The following sections are included:

- ▶ [Getting Started With Stonesoft Evader](#) (page 2)
- ▶ [Installing the Evader Test Environment](#) (page 3)
- ▶ [Evader Command Syntax](#) (page 5)
- ▶ [Mongbat Command Syntax](#) (page 10)
- ▶ [Supported Evasions](#) (page 13)

Getting Started With Stonesoft Evader

An evasion is an attempt to disguise attacks in order to avoid detection and blocking by network security systems. Evasions can be used on normal requests as well as on attacks. An attack consists of a delivery mechanism (for example, a buffer overflow) and a malicious payload (for example, code that is executed by the victim computer). The attack is considered successful if the delivery mechanism succeeds in gaining access to the victim computer, regardless of whether the security device detects or responds to the attack.

Evasion techniques can be divided into the following categories:

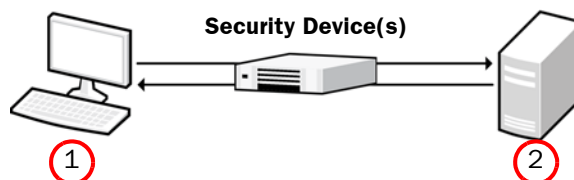
- Defined in a specification and used according to the specification (example: IPFragmentation).
- Defined in a specification but never actually used according to the specification (example: MSRPC BigEndian).
- Defined in a specification for some other component, but not defined for the way it is used in the evasion technique (example: MSRPC NDR Flags).
- Forbidden by a specification, but accepted by the victim system (example: TCP overlap).

The well-known exploits available in Stonesoft Evader are used as a carrier for evasion testing only.

Installation Environment

The installation environment consists of an attacker computer that runs the Stonesoft Evader software and one or more victim computers. The examples in this document are based on the following example installation. Your installation may vary according to your environment.

Illustration 1.1 Example Installation



1. Attacker computer
 - IPv4 Address: 172.16.120.1
 - Netmask: 255.255.255.128 (25)
 - Attacker source IPv4 address range: 172.16.120.30 - 172.16.120.126
2. Linux victim computer
 - IPv4 Address: 172.16.120.21
 - Netmask: 255.255.255.128 (25)

The following credentials are used to log in to the Linux operating system on both computers:

- **User:** root
- **Password:** password

Installing the Evader Test Environment

Installation Files

The attacker computer and the Linux victim computer are distributed as one OVF Template. The attacker computer includes the Evader software. The Linux victim computer includes the following vulnerable software:

- Apache HTTP Server version 2.0.64
- MySQL 4.1.22
- PHP 4.2.2
- phpBB 2.0.10 (CVE-2004-1315)

System Requirements

The OVF Template can be deployed on the virtualization platform of your choice. You may need to enable Promiscuous mode on the virtual switches to get your security device(s) to work correctly.

If you want use a Windows XP victim computer as the target for testing evasions with the conficker attack, you must install the operating system and configure the vulnerable services according to the requirements below:

- Windows XP (en-US) SP2 without patches
- MSRPC Server Service configured to allow unauthenticated MSRPC binds (CVE-2008-4250)

Example IP Addresses for Windows Victim Computer:

- IPv4 Address: 172.16.120.20
- Netmask: 255.255.255.128 (25)

Deploying the Attacker and Victim Virtual Machines in VMware

These instructions describe how to deploy OVF Template for the attacker and victim virtual machines in VMware. For other virtualization platforms, consult your platform-specific documentation.

▼ To deploy the attacker and victim virtual machines in VMware

1. Start the vSphere Client and select the Resource Pool where you want to deploy the virtual machine.
2. Select **File**→**Deploy OVF Template**. The Deploy OVF Template Wizard opens.
3. Select **Import from file** and **Browse** to the OVF File.
4. Click **Next**. The OVF Template Details open, showing the detailed information of the virtual machine to be deployed.
5. Click **Next**.
6. Enter a **Name** for the virtual machine and click **Next**.
7. Map the networks defined in the OVF template to the networks in your inventory.
8. Click **Next** and check that the deployment settings are correct.

9. Click **Finish** to deploy the virtual machine. When the deployment is finished, click **Close** to close the status dialog.

What's Next?

- ▶ Repeat these steps to deploy the other virtual machine, then continue by [Configuring the Attacker Computer](#) (page 4).

Configuring the Attacker Computer

▼ To configure the attacker computer

1. Log in to the attacker virtual machine with the following credentials:
 - **User:** root
 - **Password:** password
2. Edit the following settings in the `/etc/network/interfaces` file according to your network environment:

```
auto eth0

iface eth0 inet static
    address 172.16.120.1
    netmask 255.255.255.128
```
3. Enter the following command to restart networking and apply your network settings:

```
/etc/init.d/networking restart
```

What's Next?

- ▶ Continue by [Configuring the Linux Victim Computer](#) (page 4).

Configuring the Linux Victim Computer

▼ To configure the Linux victim computer

1. Log in to the victim virtual machine with the following credentials:
 - **User:** root
 - **Password:** password
2. Edit the following settings in the `/etc/network/interfaces` file according to your network environment:

```
auto eth0

iface eth0 inet static
    address 172.16.120.21
    netmask 255.255.255.128
```
3. Enter the following command to restart networking and apply your network settings:

```
/etc/init.d/networking restart
```
4. Enter the following command to start the vulnerable services:

```
./start_victim_services.sh
```

Evader Command Syntax

Stonesoft Evader can be used directly to test single evasions. Testing single evasions is recommended before beginning automated tests of combined evasions to identify any single evasions that are not detected by the security device. Any successful single evasions can be excluded from automatic testing. To use automatic testing, see [Mongbat Command Syntax](#) (page 10).

▼ To give Evader commands

1. (Optional) If you want to use a different keyboard layout than the default (UK), enter the following command to change the keyboard layout: `loadkeys <key map>`

Example `loadkeys fi`

2. Change to the `/home/<your account>/evader` directory.
3. Enter the following command to verify connectivity to the victim computer before testing evasions:
`./evader --attack=http_phpbb_highlight --if=eth0
--src_ip=172.16.120.100 --dst_ip=172.16.120.21
--verifydelay=1000 --clean`
4. Enter commands in the following format: `./evader [options]`.

Example `./evader --attack=conficker --evasion=tcp_overlap,345,old,random`

The options for the `evader` command are explained in the tables below.

Table 1.1 Options for Network Configuration

Option	Description
<code>--if=<name></code>	Interface from which the attacks originate.
<code>--src_ip=<IP address></code>	The source IP address for the attacks. This address must be a unique IP address from the same subnet as the IP address of the attacker computer. Note! The Evader tool implements its own TCP/IP stack. Do not use the attacker computer's IP address as the source IP address.
<code>--src_port=<port number></code>	The source port number for the attacks. By default, a random port number is used.
<code>--src_mask=<netmask></code>	The source netmask in CIDR notation.
<code>--gw=<IP address></code>	The IP address of the default gateway if needed.
<code>--dst_ip=<IP address></code>	The IP address of the victim computer.
<code>--dst_port=<port number></code>	The port number of a vulnerable service on the victim computer.

Table 1.2 Options for Attack Configuration

Option	Description
<code>--attacks -a</code>	Lists the supported attacks.
<code>--info=<attack name></code>	Shows detailed information about the specified attack and the options that can be used with the attack.
<code>--attack=<attack name></code>	Specifies the attack to use.
<code>--clean</code>	Sends only a non-malicious payload to check victim availability.
<code>--obfuscate</code>	Set all obfuscation flags in the attack.
<code>--extra=<option=value> (,<option=value>)*</code>	<p>Specifies additional options that configure the behavior of the attack used with the evasion. Separate multiple options with a comma. The extra exploit options for each attack are listed when you use <code>--info=<attack name></code> to display detailed information about the attack.</p> <p>Example: <code>--extra=bindport=4567,obfuscate_enc=true,no_banner=true</code></p>

Table 1.3 Options for Evasion Configuration

Option	Description
<code>--evasions -e</code>	<p>Lists the supported evasion techniques for a specific attack and the evasion-specific options. Not all evasions can be used with all attacks. An attack must be selected using the <code>--attack=<attack name></code> option.</p> <p>Example: <code>--attack=conficker --evasions.</code></p>
<code>--evasion= [<start stage>,<end stage>] <evasion_name>, <evasion_option=value> (,<evasion_option=value>)*</code>	<p>Specifies the evasion to use and the evasion-specific options. Separate multiple options with a comma. An attack must be selected using the <code>--attack=<attack name></code> option.</p> <p>Stages are different points in the progression of an attack. You can optionally specify that the evasion is only applied during a particular time period in the attack. The supported stages for an attack are listed in the attack description. The stages at which a particular evasion can be applied are listed in the evasion description.</p>

Each evasion has its own set of options that are specific to that evasion. Use the following options to show detailed information about the options for a specific evasion:

```
--info=<evasion_name>
```

Each evasion-specific option accepts one of the following three types of values (the evasion description indicates what type of value each option accepts):

- **Integer:** An integer value. The evasion description indicates the range of valid integers and how to use them with the evasion.
- **Probability:** Indicates how often the evasion is applied to the traffic. Probability values can be entered in three ways:
 - As a percentage. For example, 75%. The evasion is applied with a probability of 75% to each packet, so on average at the end of the attack, 75% of packets will have received the evasion.
 - As a number. For example, 3. The evasion is applied to exactly every third packet.
 - As a list of iterations each preceded by '#'. For example, #3#6#13. The evasion is applied exactly at the third, sixth and thirteenth iterations.
- **Multiple Choice:** A specific named option. The evasion description specifies whether single or multiple options can be used with the evasion. The syntax for the option depends on whether single options or multiple options can be used:
 - **Single Valid:** A single named option. Quotes are optional.
 - **Multiple Valid:** A single named option or a list of named options separated with a pipe (|). The entire list of options must be enclosed in quotes to prevent the pipe from being interpreted as part of the Linux shell command.

Table 1.4 Other Options

Option	Description
--version -v	Shows the Evader software version.
--cfg_file=<file name>	Reads the configuration from a configuration file.
--autoclose	Automatically close shells without interaction after a successful attack.
--shell_tcp	Opens the shell control channel to a TCP socket instead of standard IO.
--summary -s	Prints a summary before exiting.
--iterations=<number>	Number of consecutive iterations.
--verifydelay=<length of delay>	Length of time (in milliseconds) to wait before verifying the attack result. The default is 100.
--randseed=<string>	Sets the seed to use for random number generation. The random number generator generates random data for the payload of certain attacks.
--record=<file name>	Records all generated traffic in PCAP format to the specified file.

Example Evader Output

The following examples show the output when using Evader with the conficker attack.

Table 1.5 Example Evader Command

```
./evader --if=eth2 --src_ip=10.102.2.71 --dst_ip=10.102.0.4
--src_mask=16 --autoclose --attack=conficker --verifydelay=200 --obfuscate
--randseed=wIBtcAUUS4 --evasion=msrpc_bigendian
Info: Using random seed +wIBtcAUUSI
The following evasions are applied from stage msrcp_bind to end:
  - MSRPC messages are sent in the big endian byte order
```

Table 1.6 Successful Attack

```
Info: NetBIOS connection 10.102.2.71:54263 -> 10.102.0.4:445
Info: SMB Native OS is "Windows 5.1", targeting Windows XP SP2
Info: Sending MSRPC request with exploit
Info: Shell found, attack succeeded
Info: Shell closed
0: Success.
```

Table 1.7 Successful Attack With Late IPS Termination

```
Info: NetBIOS connection 10.102.2.71:59220 -> 10.102.0.4:445
Info: SMB Native OS is "Windows 5.1", targeting Windows XP SP2
Info: Sending MSRPC request with exploit
Info: Failed to send MSRPC request containing the exploit.
Info: TCP socket closed due to the maximum number of retransmits sent - probable IPS
termination.
Info: Shell found, attack succeeded
Info: Shell closed
0: Success.
```

Table 1.8 Failed Attack With IPS Termination

```
Info: NetBIOS connection 10.102.2.71:59220 -> 10.102.0.4:445
Info: SMB Native OS is "Windows 5.1", targeting Windows XP SP2
Info: Sending MSRPC request with exploit
Info: Failed to send MSRPC request containing the exploit.
Info: TCP socket closed due to the maximum number of retransmits sent - probable IPS
termination.
Info: No shell, attack failed
200: Connection terminated.
```

Table 1.9 Connection Failed

```
Info: NetBIOS connection 10.104.4.71:64609 -> 10.104.0.10:445
Info: MSRPCServerExploit::MSRPCBind() - Failed to connect to 10.104.0.10:445
Error: Exploit running failed
300: TCP connection failed.
```

Using the Evader Web Interface

The Evader web interface requires a graphical user interface with a web browser, which are not included in the attacker virtual machine.

▼ To use the Evader web interface

1. Change to the `/home/<your account>/evader` directory.
2. Enter the following command: `ruby webgui.rb`.
3. Open a web browser and browse to `http://localhost:8000`. The Evader web interface opens.

The screenshot shows the Evader web interface. Callout 3 points to the 'Module' dropdown menu where 'HTTP phpBB highlight' is selected. Callout 4 points to the 'Description' field for the selected module. Callout 5 points to the 'Device Under Testing (DUT)' section, specifically the 'Interface' field set to 'eth0'. Callout 6 points to the 'Evasions' section on the right, where various evasion options are listed with checkboxes. Callout 7 points to the 'Execute' button in the 'Execution Configuration' section.

4. Select the attack **Module** for testing the evasions.



Note – If you want to test evasions against a Windows XP victim computer with the Conficker module, you must install the operating system and configure the vulnerable services. See the [System Requirements](#) (page 3) for more information.

5. Define the module-specific options according to your environment.
6. Define the network options for the Evader command as instructed below:

Setting	Description
Interface	Enter the name of the interface on the attacker computer from which the attacks originate.
Source IP	Enter the source IP address for the attacks. This address must be a unique IP address from the same subnet as the IP address of the attacker computer. Note! The Evader tool implements its own TCP/IP stack. Do not use the attacker computer's IP address as the source IP address.
Destination IP	Enter the IP address of the victim computer.
Gateway IP	Enter IP address of the default gateway if needed.

7. Select the evasions to use and configure evasion-specific options.
8. Click **Execute**. The Evader command is executed and the output is shown in the text box at the bottom of the page.

Mongbat Command Syntax

Mongbat is a testing automation tool that combines evasions to attack the target host. Mongbat can run multiple instances of the Evader tool with the specified parameters.

▼ To give Mongbat commands

1. Change to the `/home/<your account>/evader` directory.
2. Enter commands in the following format: `ruby mongbat.rb [options]`.

Example `ruby mongbat.rb --iface=eth1 --attack=http_phpbb_highlight
--attacker=172.16.120.30 --victim=172.16.120.21 --workers=16 --time=3600`

The options for the mongbat command are explained in the table below.

Table 1.10 Mongbat Options

Option	Description
<code>--mode=(solo dual random)</code>	The mode of attack: Solo mode uses individual evasions with some options. Dual mode combines two evasions from the list of enabled evasions. Random mode uses random evasions from the list of enabled evasions with random options. The default is random.
<code>--attack=<attack name></code>	The attack to use. Default: conficker.
<code>--iface=<interface></code>	Interface from which the attacks originate.
<code>--attacker=<src ip></code>	The starting source IP address for the attacks. This address must be a unique IP address from the same subnet as the IP address of the attacker computer. The first worker uses this address, and any additional workers use the next sequential IP addresses. There must be a sufficient range of free IP addresses to provide a unique IP address for each worker. Note! The Evader tool implements its own TCP/IP stack. Do not use the attacker computer's IP address as the source IP address.
<code>--victim=<dst ip></code>	The IP address of the victim computer.
<code>--mask=<netmask></code>	The netmask in CIDR notation.
<code>--gw=<IP address></code>	The gateway address if the victim is not in the local network. Default: empty.
<code>--time=<time in seconds></code>	The duration of the attack in seconds when random mode is used. The default is 60 seconds.
<code>--workers=<worker count></code>	The number of workers to use for the attacks. Each worker is a separate instance of the Evader program. There must be a sufficient range of free IP addresses to provide a unique IP address for each worker. The default is 1 worker.
<code>--use_evasions=<evasion> (,evasion)*</code>	Use only the specified evasion(s).

Table 1.10 Mongbat Options (Continued)

Option	Description
<code>--disable_evasions=<evasion> (,evansion)*</code>	Exclude the specified evasions from testing. Disabling evasions is recommended when dual mode or random mode is used for attacks. Disabling single evasions that are successful against the security device is also recommended.
<code>--check_victim=(true false)</code>	Check that the victim allows legal traffic without evasions before attacking. Default: true.
<code>--record=<directory name></code>	Records the attacks to the specified directory in pcap format.
<code>--min_evasions=<number></code>	The minimum number of evasions to try in random mode. Default: 1.
<code>--max_evasions=<number></code>	The maximum number of evasions to try in random mode. A value of 0 means an unlimited number of evasions will be tried. Default: 0.
<code>--index=<begin(-end)?></code>	The start index and optional stop index for solo and dual mode.
<code>--stop_on_success</code>	Stop if an attack is successful.
<code>--passthrough</code>	Pass remaining unknown arguments directly to Evader.

Example Mongbat Output

The following example shows the Mongbat output for a 60 second run with one successful attack, 21 clean check failures (indicated by 'C') and 135 failed attack attempts (indicated by '.'). By default, Mongbat first checks that a non-malicious connection to the victim computer works, and then attacks with evasions. A log file is written to `mongbat.rb.log` in the directory where the Mongbat command is run.

Table 1.11 Example Mongbat Output

```
ruby mongbat.rb --mode=random --attack=conficker --victim=10.102.0.4
--iface=eth2 --attacker=10.102.100.100 --mask=16 --validator=externals/
conficker_validator.rb --workers=4 --time=60
2012-07-09 11:21:57 INFO Using binary /root/evader_0_9_8_556/evader version Evader
2012-07-09 11:21:57 INFO Using rand seed IyQlyTgzma4=
2012-07-09 11:21:57 INFO loading externals/conficker_validator.rb
2012-07-09 11:21:57 INFO External Validator:
/root/evader_0_9_8_556/externals/conficker_validator.rb: Validate Conficker against
Windows XP SP2 Starting evasions generator: Random evasions generator (Evasion adding
percentage is 0.003278688524590164)
```

Table 1.12 Example Mongbat Output (Continued)

```

0 runs averaging 0.00 runs / second ; progress: 1/60....C.....
10 runs averaging 1.26 runs / second ; progress: 8/60...C.....C...
24 runs averaging 1.85 runs / second ; progress: 13/60...C.....C...
38 runs averaging 2.12 runs / second ; progress: 18/60...C.....C.
50 runs averaging 2.18 runs / second ; progress: 23/60.....C.....
63 runs averaging 2.25 runs / second ; progress: 28/60C.....C.....
77 runs averaging 2.34 runs / second ; progress: 33/60..C.....C....
evader --if=eth2 --src_ip=10.102.4.33 --dst_ip=10.102.0.4 --src_mask=16 --autoclose
--attack=conficker --verifydelay=200 --obfuscate --randseed=u5MBrg3jsSU=
--evasion=msrpc_bigendian
Info: Using random seed +wIBtcAUUSI
The following evasions are applied from stage msrpc_bind to end:
    - MSRPC messages are sent in the big endian byte order

Info: NetBIOS connection 10.102.4.71:54263 -> 10.102.0.4:445
Info: SMB Native OS is "Windows 5.1", targeting Windows XP SP2
Info: Sending MSRPC request with exploit
Info: Shell found, attack succeeded
Info: Shell closed
0: Success.
91 runs averaging 2.40 runs / second ; progress: 38/60...C.....C..
105 runs averaging 2.44 runs / second ; progress: 43/60.....C.....C
119 runs averaging 2.48 runs / second ; progress: 48/60.....C.....
132 runs averaging 2.49 runs / second ; progress: 53/60..C.....C.....
147 runs averaging 2.53 runs / second ; progress: 58/60..C.....C
2012-07-09 11:22:59 INFO Done.
Printing test result
2012-07-09 11:22:59 INFO Mongbat test report

Using /root/evader_0_9_8_556/evader version Evader licensed to

Started : 2012-07-09 11:21:57 +0300
Finished: 2012-07-09 11:22:59 +0300
Attack : conficker

Network setup:
Attackers: 10.102.100.100-10.102.100.104
Victim : 10.102.0.4

Number summary

```

	Total	Clean	Exploit
Attempts	156	156	134
Failures	156	21	134
Success	1	135	1

Supported Evasions

The evasions available depend on the selected attack. For example, HTTP evasions can only be used with HTTP attacks. Use the following command to list the supported evasion techniques for a specific attack:

```
./evader --attack=<attack name> --evasions
```

Table 1.13 Supported Evasions

Evasion	Description	Attack(s)
ipv4_frag	IPv4 fragmentation	conficker, http_phpbb_highlight
ipv4_opt	IPv4 options	conficker, http_phpbb_highlight
msrpc_bigendian	MSRPC big endian	conficker
msrpc_groupends	Group MSRPC fragments into a single send	conficker
msrpc_ndrflag	MSRPC NDR modifications	conficker
msrpc_seg	MSRPC request segmentation	conficker
netbios_chaff	NetBIOS chaff	conficker
netbios_init_chaff	NetBIOS initial chaff	conficker
smb_chaff	SMB chaff	conficker
smb_decoytrees	SMB decoy trees	conficker
smb_fnameobf	SMB filename obfuscation	conficker
smb_seg	SMB write segmentation	conficker
smb_writeandxpad	SMB WriteAndX padding	conficker
tcp_chaff	TCP Chaff	conficker, http_phpbb_highlight
tcp_initialseq	TCP initial sequence number	conficker, http_phpbb_highlight
tcp_inittsopt	TCP timestamp option settings	conficker, http_phpbb_highlight
tcp_nocwnd	Disable TCP congestion avoidance	conficker, http_phpbb_highlight
tcp_nofastretrans	Disable TCP fast retransmit	conficker, http_phpbb_highlight
tcp_order	TCP segment order	conficker, http_phpbb_highlight
tcp_overlap	TCP segment overlap	conficker, http_phpbb_highlight
tcp_paws	TCP PAWS elimination	conficker, http_phpbb_highlight
tcp_rcv_window	TCP receive window	conficker, http_phpbb_highlight
tcp_seg	TCP segmentation	conficker, http_phpbb_highlight
tcp_timewait	TCP TIME-WAIT decoys	conficker, http_phpbb_highlight

Table 1.13 Supported Evasions

Evasion	Description	Attack(s)
tcp_tsopreply	TCP timestamp echo reply modifications	conficker, http_phpbb_highlight
tcp_urgent	TCP urgent data	conficker, http_phpbb_highlight
http_header_lws	HTTP header linear whitespace	http_phpbb_highlight
http_known_user_agent	HTTP known user agent	http_phpbb_highlight
http_request_line_separator	HTTP request line separator	http_phpbb_highlight
http_request_method	HTTP request method	http_phpbb_highlight
http_request_pipelined	HTTP request pipelined	http_phpbb_highlight
http_url_absolute	HTTP URL absolute	http_phpbb_highlight
http_url_dummpath	HTTP dummy paths	http_phpbb_highlight
http_url_encoding	HTTP URL encoding	http_phpbb_highlight
http_version	HTTP request version	http_phpbb_highlight

Stonesoft Guides

Administrator's Guides - step-by-step instructions for configuring and managing the system.

Installation Guides - step-by-step instructions for installing and upgrading the system.

Reference Guides - system and feature descriptions with overviews to configuration tasks.

User's Guides - step-by-step instructions for end-users.

For more documentation, visit
www.stonesoft.com/support/

Stonesoft Corporation

Itälahdenkatu 22 A
FI-00210 Helsinki
Finland

Tel. +358 9 476 711
Fax +358 9 4767 1349

Stonesoft Inc.

1050 Crown Pointe Parkway
Suite 900
Atlanta, GA 30338
USA

Tel. +1 770 668 1125
Fax +1 770 668 1131