

# [NEWS] UNIX Entropy Source Can Be Used For Keystroke Timing Attacks

*Source:* <http://www.derkeiler.com/Mailing-Lists/Securiteam/2003-08/0051.html>

---

*From:* SecuriTeam ([support\\_at\\_securiteam.com](mailto:support_at_securiteam.com))

*Date:* 08/18/03

To: [list@securiteam.com](mailto:list@securiteam.com)

Date: 18 Aug 2003 14:01:37 +0200

The following security advisory is sent to the securiteam mailing list, and can be found at the SecuriTeam web site: <http://www.securiteam.com>

-- promotion

The SecuriTeam alerts list – Free, Accurate, Independent.

Get your security news from a reliable source.

<http://www.securiteam.com/maillinglist.html>

-----

UNIX Entropy Source Can Be Used For Keystroke Timing Attacks

---

## SUMMARY

Several UNIX systems provide a secure entropy source maintained by collecting certain information that is supposed to be practically unpredictable (such as interrupt timings, keyboard scancodes or device request times), then running it thru one-way hashing function (SHA1 or MD5).

A device called `/dev/random` estimates the amount of entropy available in the entropy pool, and blocks on read when the pool gets empty. This continues until the buffer is seeded with some new data due to one of the events mentioned.

It is possible to monitor `/dev/random` and learn some interesting information about the events.

## DETAILS

By emptying the `/dev/random`, and then timing the moments when data becomes available, we can precisely determine at what intervals those events occur. While we cannot determine what data is being added to the entropy

## Securiteam: [NEWS] UNIX Entropy Source Can Be Used For Keystroke Timing Attacks

pool, we can rather easily tell a situation when a keystroke data is added due to a specific pattern triggered by it. For example, on Linux the sequence is:

Keypress scancode in: one to two bytes available  
50–150 ms delay (keypress duration)  
Key release scancode in: one to two bytes available  
50 ms or more delay

Other types of events, such as disk activity, usually generate a burst of events, usually under 1 ms away from each other, or have other distinct patterns (besides, those events happen only sporadically).

Because of this, it is possible to measure keypress AND key release timings very precisely, for any console user of a machine we have an unprivileged account on. Timings between keystrokes depend on the distance between subsequently pressed keys for each hand, and the placement of the hand, keystroke durations usually depend on the finger used – all this making it quite easy to come up with a nice subset of possible passwords, and not impossible to determine some of the commands typed.

There is some mature research in the field of recovering typed information or its certain properties from the timing information (both for the purpose of biometrics and surveillance), so I don't think to get too far into this. Michal will just provide an example of the difference in average keystroke timings for two related words typed by a keyboard-proficient user:

1) "evil"

```
e press |
e release | =====
v press | =====
i press | =====
v release | ==
i release | =====
l press | =====
l release | =====
```

Note out of sync 'v' release after hand switch.

2) "good"

```
g press |
g release | =====
o press | =====
o release | =====
o press | =====
d press | =====
o release | =====
d release | =====
```

Securiteam: [NEWS] UNIX Entropy Source Can Be Used For Keystroke Timing Attacks

Same for second 'o'.

A workaround would be to add some latency before unblocking pending read(s) when a new information becomes available. /dev/random is a very high latency device, and no program depends on how fast the data is available for its normal operation.

ADDITIONAL INFORMATION

The information has been provided by <mailto:lcamtuf@ghettot.org> Michal Zalewski

=====

This bulletin is sent to members of the SecuriTeam mailing list.  
To unsubscribe from the list, send mail with an empty subject line and body to:  
list-unsubscribe@securiteam.com  
In order to subscribe to the mailing list, simply forward this email to: list-subscribe@securiteam.com

=====  
=====

DISCLAIMER:

The information in this bulletin is provided "AS IS" without warranty of any kind.  
In no event shall we be liable for any damages whatsoever including direct, indirect, incidental, consequential, loss of business profits or special damages.