

Otoro

fat tuna in the OpenBSD ports fish-tank

30 June 2016

Adam Wolk

OpenBSD Developer

Otoro

(oh-toh-roh) The fattiest portion of the tuna, found on the underside of the fish.

<http://www.sushifaq.com/sushi-sashimi-info/sushi-terminology/>

Obrazy dla otoro

Zgłoś grafiki



Więcej zdjęć dla zapytania otoro

Otoro | Otoro Sushi

otoro.com/ ▼ Tłumaczenie strony

Otoro is the most desired part of the tuna belly that usually doesn't get a price on any sushi house menus. Toro or o-toro is the best part of the tuna.

About me



- OpenBSD User since April 2014;
- OpenBSD Developer since February 2016;
- Co-Founder of koparo.com;
- Linux user since mid 90's;
- Interested in infosec, programming & gaming;
- 7 years in the banking sector.

About OpenBSD

Started as a fork of NetBSD in 1995

Mostly known for:

- OpenSSH
- LibreSSL (no, not OpenSSL)
- pf
- carp
- clockwork releases every 6 months
- with release songs & artwork
- more...

[innovations](https://www.openbsd.org/innovations.html) (https://www.openbsd.org/innovations.html)

Being MAINTAINER

Being a maintainer is more than just getting a piece of software to compile.

- test the software, fully
- patch for platform best practices (strcpy, arc4random, pledge)
- upstream patches
- proxy bug reports from OpenBSD users to upstream
- become upstream
- make choices, cut dependencies, write rc scripts

Don't be afraid to disable features, changing a default wildcard network listen to a localhost one etc.

OpenBSD expects the packagers to care about people using the software.

Binary packages

We expect people to use binary packages. The actual ports system is used by people preparing the ports & building binary packages.

Binary packages are provided for:

- the stable release
- snapshots of -CURRENT

Ongoing updates for stable releases without going -CURRENT are possible by using

[m:tier](https://stable.mtier.org) (<https://stable.mtier.org>)

Binary package update frequencies depend on the platform, amd64 is out almost daily. You might need to wait longer for others.

DPB & Proot - distributed ports builder & ports chroot builder

dpb is used to build ports on a cluster of machines, or on a single machine with several cores. dpb walks the ports tree to figure out dependencies, and starts building ports as soon as it can.

Distributed Ports Builds in OpenBSD (Marc Espie EuroBSDcon 2010)

(https://www.openbsd.org/papers/eurobsdcon_2010_dpb/index.html)

we also have the recent addition of proot which is a nice solution combining dpb & chroot.

[proot: dpb meets chroot](http://www.undeadly.org/cgi?action=article&sid=20160430163301) (<http://www.undeadly.org/cgi?action=article&sid=20160430163301>)

Reviewing port updates

Two developers need to look at a port at minimum to get it included.

What is often found in a typical port review?

- non-portable constructs (hello bash users)
- security issues (horrible defaults)
- tree pollution
- software crashing
- regular bugs
- missing dependencies
- wrong build options (debug/release)

Worth the time to double check.

Porting tools

Important resources

[Porting Guide](https://www.openbsd.org/faq/ports/guide.html) (https://www.openbsd.org/faq/ports/guide.html)

[portroach](http://portroach.openbsd.org/) (http://portroach.openbsd.org/)

[man\(1\)](http://man.openbsd.org/) (http://man.openbsd.org/)

Important tools

- portcheck
- make port-lib-depends-check | update-plist
- ktrace
- gdb (egdb) - for C/C++ based ports
- portimport (for developers)
- man pages: port-modules / bsd.port.mk / ports

SQL ports

What if you could have the whole ports tree as a single sqlite3 database?

Need a list of all ports depending on lang/go?

```
$ sqlite3 /usr/local/share/sqlports "select * from modules where value='lang/go';"  
devel/go-check-v1|lang/go  
devel/go-tools|lang/go  
devel/hub|lang/go  
net/go-net|lang/go  
...
```

Just `pkg_add sqlports` and query all the info you need.

Meaning of ports for OpenBSD?

We often turn on intrusive exploit mitigations by default. W^X, ASLR and pledge as quick examples.

The ports tree is a great way to have a large portion of existing software exposed to those mitigations at large.

We might not be as big as your typical Linux distribution, but we are also large enough to get noticed.

We do upstream patches and we do annoy upstream with stuff like arc4random.

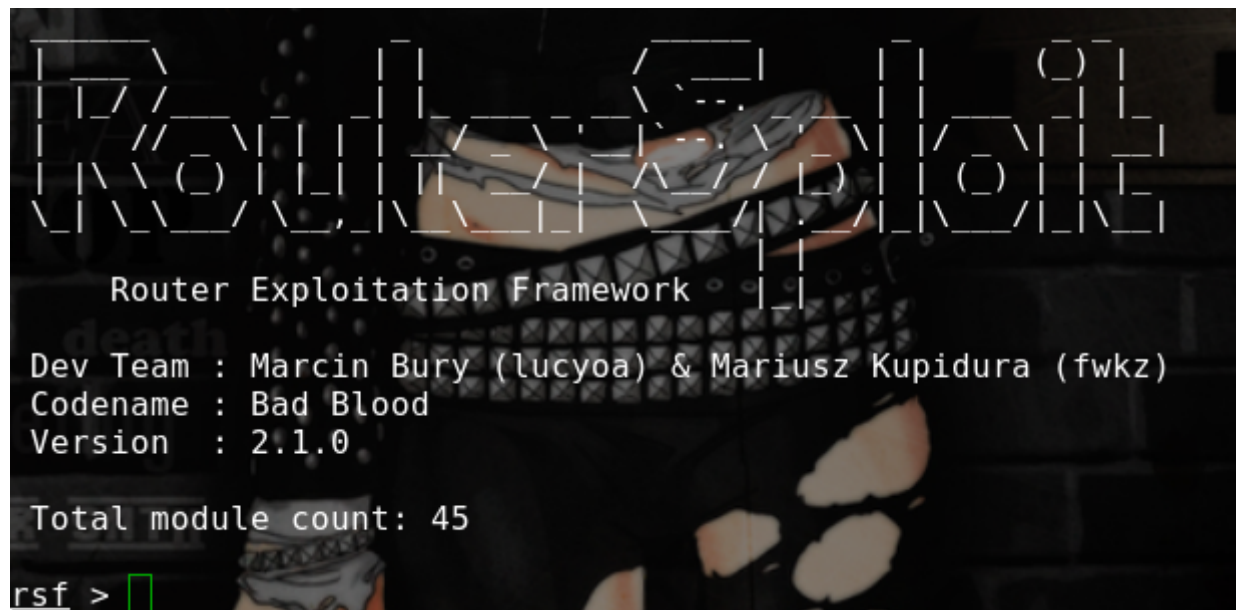
M. W. Lucas nicely noted on several occasions that OpenBSD is the pressure cooker of the open source ecosystem.

We break our ports tree often, so other *nix systems can have the nice security features enabled by default.

We are not afraid to remove ports.

Is this network secure?

Hands on - porting routersploit

A terminal window showing the RouterSploit framework. The background is a dark image of a person's hands holding a laptop. The text in the terminal is as follows:

```
RouterSploit
Router Exploitation Framework
Dev Team : Marcin Bury (lucyoo) & Mariusz Kupidura (fwkz)
Codename : Bad Blood
Version  : 2.1.0
Total module count: 45
rsf > 
```

Supporting the project

The OpenBSD Foundation (<http://www.openbsdoundation.org/>)

OpenBSD Donations (<http://www.openbsd.org/donations.html>)

CD's & other swag (<http://openbsdstore.com/>)



Any questions?

Thank you

Adam Wolk

OpenBSD Developer

awolk@openbsd.org (mailto:awolk@openbsd.org)

<https://www.openbsd.org> (https://www.openbsd.org)

<https://blog.tintagel.pl> (https://blog.tintagel.pl)

[@mulander](http://twitter.com/mulander) (http://twitter.com/mulander)

<https://koparo.com> (https://koparo.com)

[@koparo_golf](http://twitter.com/koparo_golf) (http://twitter.com/koparo_golf)