MAGAZINE

BSD

FOR NOVICE AND ADVANCED USERS

SSH Hardening with Google Authenticator and OpenPAM

ZFS Feature Flags

Can DevOps Really Be Defined?

Debugging and Troubleshooting

WannaCry / Ransomware

Static Sites Alongside Dokku on Digital Ocean

Sapere Aude & Vitaly Repin

Interview with Daniel Cialdella Converti

VOL 11 NO 05
ISSUE 05/2017 (93)
ISSN 1898-9144

FREENAS MINI
STORAGE APPLIANCE

IT *SAVES* YOUR LIFE.

HOW IMPORTANT IS YOUR DATA?

Years of family photos. Your entire music and movie collection. Office documents you've put hours of work into. Backups for every computer you own. We ask again, *how important is your data?*

NOW IMAGINE LOSING IT ALL

Losing one bit - that's all it takes. One single bit, and your file is gone.

The worst part? **You won't know until you absolutely need that file again.**

Example of one-bit corruption

THE SOLUTION

The FreeNAS Mini has emerged as the clear choice to save your digital life. **No other NAS in its class offers ECC (error correcting code) memory and ZFS bitrot protection to ensure data always reaches disk without corruption and *never degrades over time*.**

No other NAS combines the inherent data integrity and security of the ZFS filesystem with fast on-disk encryption. No other NAS provides comparable power and flexibility. The FreeNAS Mini is, hands-down, the best home and small office storage appliance you can buy on the market. **When it comes to saving your important data, there simply is no other solution.**

The Mini boasts these state-of-the-art features:

- 8-core 2.4GHz Intel® Atom™ processor
- Up to 16TB of storage capacity
- 16GB of ECC memory (with the option to upgrade to 32GB)
- 2 x 1 Gigabit network controllers
- Remote management port (IPMI)
- Tool-less design; hot swappable drive trays
- FreeNAS installed and configured

iXsystems™

http://www.iXsystems.com/mini

intel® inside™ ATOM™

FREENAS CERTIFIED STORAGE

With over six million downloads, FreeNAS is undisputedly *the* most popular storage operating system in the world.

Sure, you could build your own FreeNAS system: research every hardware option, order all the parts, wait for everything to ship and arrive, vent at customer service because it *hasn't*, and finally build it yourself while hoping everything fits - only to install the software and discover that the system you spent *days* agonizing over **isn't even compatible**. Or...

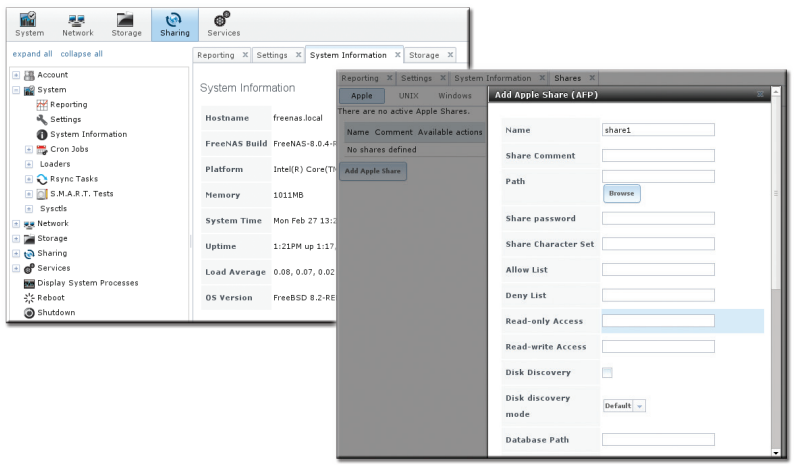
MAKE IT EASY ON YOURSELF

As the sponsors and lead developers of the FreeNAS project, iXsystems has combined over 20 years of hardware experience with our FreeNAS expertise to bring you FreeNAS Certified Storage. **We make it easy to enjoy all the benefits of FreeNAS without the headache of building, setting up, configuring, and supporting it yourself.** As one of the leaders in the storage industry, you know that you're getting the best combination of hardware designed for optimal performance with FreeNAS.

Every FreeNAS server we ship is...

» Custom built and optimized for your use case
» Installed, configured, tested, and guaranteed to work out of the box
» Supported by the Silicon Valley team that designed and built it
» Backed by a 3 years parts and labor limited warranty

As one of the leaders in the storage industry, you know that you're getting the best combination of hardware designed for optimal performance with FreeNAS. **Contact us today for a FREE Risk Elimination Consultation with one of our FreeNAS experts.** Remember, every purchase directly supports the FreeNAS project so we can continue adding features and improvements to the software for years to come. **And really - why would you buy a FreeNAS server from *anyone* else?**



FreeNAS 1U

- Intel® Xeon® Processor E3-1200v2 Family
- Up to 16TB of storage capacity
- 16GB ECC memory (upgradable to 32GB)
- 2 x 10/100/1000 Gigabit Ethernet controllers
- Redundant power supply

FreeNAS 2U

- 2x Intel® Xeon® Processors E5-2600v2 Family
- Up to 48TB of storage capacity
- 32GB ECC memory (upgradable to 128GB)
- 4 x 1GbE Network interface (Onboard) - (Upgradable to 2 x 10 Gigabit Interface)
- Redundant Power Supply

http://www.iXsystems.com/storage/freenas-certified-storage/

EDITOR'S WORD

MAGAZINE BSD

Dear Readers,

It is nice to meet you again and present to you with the next BSD issue. I hope you will find the articles interesting and eventually learn many useful tips and tricks. This time, I want to focus on short but useful hints that will help you in your daily tasks. Enjoy reading and share your thoughts with me.

For those of you who prefer to visit our website, I want to start publishing some of the articles from the BSD magazine online on our blog. I am doing this for convenience purposes so that those of you who do not have time to read the issue all at once but prefer to visit our website more often to search easily for the most-wanted information can do so with ease. If you have your favourite articles that you think should be published, please feel free to contact me.

Also, I plan to organise a contest to choose the best articles that were written in the first half of 2017. I hope you like the idea and you will take part in it. I will publish more details on the BSD magazine website and in our newsletter. That's all for my announcements, but I would like to add that I'm still looking for new ideas for topics on articles. I look forward to your emails and your suggestions on what is new and trending. You can always reach me at ewa@bsdmag.org.

Having said that, let us now see what we have inside this BSD magazine issue. Carlos will refresh his own article for you to remind you about debugging skills. In his tutorial, he used a real life situation where debugging skills will save us time, headaches and possibly find a solution with a minimal amount of effort. For those of you who want to know more about DevOps, please go to E.G Nadhan. He will explain what he has in mind. The definition that he compiled is based upon myriad interpretations that he has come across in his interactions with his peers, as well as customers, partners, service providers and colleagues t colleagues over social media.

I would like to highlight two articles for this month. The first one is ZFS Feature Flags written by Natalia Portillo. It is a great article which offers an easy explanation for those who want to know more about ZFS feature flags. The second one was written by Abdorrahman Homaei entitled SSH Hardening with Google Authenticator and OpenPAM. From this article, you will learn about techniques one of which is called 2-Step Verification. It is a very good and simple how-to article.

Then, I am encouraging you to read our new column with short but very practical articles which contain knowledge on how to solve your problems and show simple solutions or advice on where, how and what to do. This time you can read about WannaCry / Ransomware and Static Sites Alongside Dokku on Digital Ocean. Please feel free to read the table of contents on the next page for more information.

As always, I would like to thank you all for submitting great articles and willingness to help me create this issue of BSD magazine.

Enjoy reading!
Ewa and the BSD Team

Table of Contents

iXsystems' TrueNAS Receives Veeam Backup Certification

Certification testing showed that TrueNAS is two to four times faster than Veeam requirements.

iXsystems, the industry leader in storage and servers driven by Open Source, today announced that it has passed the Veeam Backup and Replication v9.5 for VMware certification tests for the TrueNAS Z20, TrueNAS Z30, TrueNAS Z35, and TrueNAS Z50 TrueFlash line of unified enterprise storage systems that are running TrueNAS v9.10 or later. This certification tests the speed and power of the data storage repository using a testing methodology defined by Veeam for Full Backups, Full Restores, Synthetic Full Backups, and Instant VM Recovery from within the Veeam Backup and Recovery environment.

iXsystems created a blog describing the tests and the test results. Our test results exceeded Veeam's minimums, with full backups taking half as long as Veeam requires. The certification shows that TrueNAS is ideal for supporting Veeam backups.

Backup Image Repository

A TrueNAS Z20 Hybrid Storage array with 8.4 TB of data was used as the repository of the Veeam backup images generated by the certification testing. Two of the four certification tests required that these images be used during restoration, and the TrueNAS Z20 was used for that as well.

VMware Servers

Two servers, each with dual Xeon v4 CPUs, were used to run the certification tests. These servers ran all the virtual machines, including the Windows 2012 R2 server that ran the Veeam Backup and Replication 9.5 servers. In addition, these servers locally stored the test Windows server VMs and ran the vCenter server VM.

VM Repository

iXsystems' FreeNAS All-Flash Array with 2.88 TB was used to store all the Linux test VMs that were backed up and recovered for Veeam's certification tests.

Following the requirements of the "Performance Testing Guide for Backup Storage" provided by Veeam, the TrueNAS Z20, TrueNAS Z30,TrueNAS Z35, and TrueNAS Z50 TrueFlash satisfy all the requirements for the Veeam storage repository, and achieved certification status as a result of the tests. Customers can feel confident implementing the TrueNAS storage platform as the backup and recovery storage repository within their Veeam infrastructure.

Ideal for backup, TrueNAS configurations include flash-assisted or all-flash storage solutions. Both have blazing performance, share the same work-flow, and act as a SAN and a NAS, meeting the needs of any enterprise data storage and backup workload. TrueNAS offers High Availability to continue providing backup services in the unlikely event of a failure. Every TrueNAS model supports storage controller redundancy, hot spares, and redundant power. This enables TrueNAS to provide non-disruptive firmware updates and around-the-clock backup services with zero-downtime. It also uses OpenZFS, which ensures that backup images stay pristine. To learn more about how to use Veeam with TrueNAS or

to obtain a no-risk quote on a TrueNAS configuration, visit www.iXsystems.com/TrueNAS, email sales@iXsystems.com, or call us at 1-855-GREP-4-IX.

iXsystems Executive's Quote:

Gary Archer, Director Storage Marketing

"Backups are like insurance — needed when something goes wrong. The value of backups is to shorten the time to get a business back to work. Surveys show that more than half of our customers use TrueNAS to store their backup images, and Veeam is the #1 backup product they utilize."

Customer's Quote:

Todd Lamonia, President & CEO, IT Worldwide Services.

"The Veeam backups are working great. I am using Veeam's Direct SAN Access transport mode which connects directly to the TrueNAS using iSCSI which improves the data transfer throughput and reduces the amount of time it takes to backup each VM significantly."

Veeam Quote:

Andy Vandeveld, VP of Global Strategic Alliances, Veeam

"Veeam is strategically focused on the virtualized server market. With iXsystems TrueNAS, our combined strengths give iXsystems clients a VM-aware backup and restore solution for their TrueNAS unified storage solution. The certification of Veeam Backup & Replication software with the TrueNAS Z product line enables iXsystems to offer a joint solution to a broader range of customers. The joint solution provides them with a high-performance, high-value, and self-healing storage for virtual server data protection and replication, that keeps backup images safe."

Source: https://www.ixsystems.com/blog/ixsystems-truenas-receives-veeam-backup-certification/

June 2017 FreeBSD Developers Summit, June 7-9, 2017, Ottawa, Ontario, Canada

The June 2017 FreeBSD Developers Summit will be held in conjunction with BSDCan 2017. The FreeBSD Developer Summit will take place on the 7th and 8th of June on the same days as the BSDCan tutorials. Many developers will arrive on the night of the 6th and meet for dinner and drinks before things kick off. Most developers will stay on through BSDCan to give and attend talks. It is an excellent conference--a good FreeBSD developer showing has countless benefits, not least the opportunity to tell the world what we're doing. Not only is the event a great opportunity to meet with other developers in person, it is also a very valuable venue for companies that use FreeBSD in products to interact with the developer community and coordinate projects.

Source: https://wiki.freebsd.org/DevSummit/201706

BSDCan 2017, June 7-10, 2017, Ottawa, Ontario, Canada

BSDCan, a BSD conference held in Ottawa, Canada, quickly established itself as the technical conference for people working on and with 4.4BSD based operating systems and related projects. The organizers have found a fantastic formula that appeals to a wide range of people from extreme novices to advanced developers.

BSDCan is a technical conference for people working on and with BSD operating systems and BSD-related projects. It is the conference for developers who are focused on emerging technologies, research projects, and works in progress. However, if you want to know more about Userland infrastructure projects, you should be there.

Source: http://www.bsdcan.org/2017/

pfSense 2.3.4 Released

pfSense® software version 2.3.4 was released. This is a maintenance release in the 2.3.x series, bringing stability and bug fixes, fixes for a few security issues, and a handful of new features. The full list of changes is on the 2.3.4 New Features and Changes page, including a list of FreeBSD and internal security advisories addressed by this release. This release includes fixes for 24 bugs and 11 features.

Source: https://www.netgate.com/blog/pfsense-2-3-4-release-now-available.html

FreeNAS 11.0-RC is Available

I'm pleased to provide a quick update on the status of FreeNAS 11.0. The RC1 update was released this morning, and can be installed via ISO or updated by switching to the FreeNAS-11-STABLE train in the System -> Update tab. We decided to start this series off with a Release Candidate (RC) version, because it is rebased on a newer version of FreeBSD (11-STABLE). This version has been tested in the nightlies for several months now, but just to play it safe, we are asking for users to test out this release and let us know immediately if anything regresses. Or, if you want to let us know that it improves specific things, that's cool as well.
Now, for the stats. Your loyal, dedicated, and attractive FreeNAS development team has been working hard on this release. As of this morning, 11.0-RC includes 111 bug fixes and 23 new features. In addition, the user-guide has been updated for 11 as well. As usual, if you find bugs, please report them ASAP since we can only fix things that we know about.
This release also includes the first "official" look at the experimental new Angular-based UI. You'll be given an option to try this out on the Login screen. I just wanted to give you a couple of quick tidbits about this new UI:
It is NOT feature complete, as we have only been working on this for few weeks. While you can use it to do some things, keep this in mind as you "test-drive" it around. The feature complete version is targeted for later this year, most likely the 11.1 or 11.2 time-frame.
It follows (mostly) the same workflow as the legacy UI. This is intentional for now. In order to get us rapidly ported to the new Angular framework, we've decided to try and keep most of the workflow similar for the time being and focus purely on getting the functionality incorporated. Once we have reached the point where all major features are usable in this UI, we will have a chance to do some navel-gazing and re-think workflows of specific sections one at a time. That being said, you are welcome to send in tickets about the new UI and we will be happy to discuss and get to them all in due course.
The current theme will be changing down the road. We are planning to offer multiple themes, allowing you to pick between dark/light or perhaps even user-submitted themes.
Lastly, I wanted to mention support for virtualization. 11.0 now has a VM page, allowing you to spin up your own operating systems on FreeNAS. We are actively working on this functionality, so please, give it a whirl and report issues to the tracker.
Thanks everybody! We look forward to making the 11.X series better than ever!

Kris Moore, Director of Engineering, iXsystems

Source: http://www.freenas.org/blog/freenas-11-0-rc-now-available/

DigitalOcean Releases Free Monitoring Service so Developers Can Easily Optimize Application Performance

DigitalOcean, the cloud for developers, launched a Monitoring service that provides insight into the resource utilization and operational health of every Droplet (cloud server). Developers can collect and visualize metrics in graphs, monitor Droplet performance and receive alerts in one intuitive interface, with no configuration required.

"Our goal is to simplify the complexities of infrastructure by offering a simple and robust platform for developers to easily launch and scale their applications," said Julia Austin, CTO of DigitalOcean. "A Monitoring service is an important feature for developers, and we're thrilled to be able to offer it for free regardless of the number of Droplets. In the coming year, we'll continue to move our Monitoring service forward and introduce new capabilities for high availability, data storage, security and networking to manage larger production workloads."

The Monitoring service measures each Droplet's CPU, memory, disk utilization, disk reads and writes, network traffic and top processes. Metrics are collected at one-minute intervals and the data is retained to enable users to view both up-to-the-minute and historical data. Developers can create alert policies and receive notifications by email or Slack when usage crosses a specified threshold.

Super Humane Co-founder Josh West, a DigitalOcean customer, said: "DigitalOcean's Monitoring service is easy to use, which was our biggest requirement. Having the ability to monitor my infrastructure and create alerts right in my dashboard saves me a lot of time so I can focus on building better applications that improve the lives of people–which ties directly into our company mission. We've built and run every app for our clients on DigitalOcean and we get stellar service for an amazing price."

Enterprise Strategy Group Analyst, Daniel Conde said, "many people start monitoring too late in the game, and it's important to start the process in pre-production and continue it as the workload scale up. The early identification of potential trouble spots will help sift out performance problems before they become critical, and it's a good move for DigitalOcean to include a monitoring service at no extra cost within its developer-oriented cloud."

Ovum Principal Analyst, Roy Illsley said, "as developers move to the cloud and DevOps becomes the normal way of working, the tools that enable the monitoring and management of cloud resources and application performance become a significant requirement. If cloud-based (whether cloud native or just migrated to cloud) applications are going to deliver the correct balance of performance, service quality and efficiency, then the monitoring service must be integrated into the architecture."

Source:
https://www.digitalocean.com/company/press/releases/digitalocean-releases-free-monitoring-service-so-developers-can-easily-optimize-application-performance/

Updated Debian 8: 8.8 Released

The Debian project is pleased to announce the eighth update of its stable distribution Debian 8 (codename jessie). This update mainly adds corrections for security problems to the stable release, along with a few adjustments for serious problems. Security advisories were already published separately and are referenced where available.

Please note that this update does not constitute a new version of Debian 8 but only updates some of the packages included. There is no need to throw away old jessie CDs or DVDs but only to update it via an up-to-date Debian mirror after an installation, to cause any out of date packages to be updated.

Those who frequently install updates from security.debian.org won't have to update many packages since most updates from security.debian.org are included in this update. New installation media and CD and DVD images containing updated packages will be available soon at the regular locations.

Upgrading to this revision online is usually done by pointing the aptitude (or apt) package tool (see the sources.list(5) manual page) to one of Debian's many FTP or HTTP mirrors. A comprehensive list of mirrors is available at: https://www.debian.org/mirror/list

Source: https://www.debian.org/News/2017/20170506

TrueOS Q&A

A few of the TrueOS/Lumina devs held a live Q&A session on Discourse. I've reproduced a few of the questions and responses here, but be sure to check out the full Q&A thread on Discourse. A huge "Thank You!" to all our users and developers who participated!

Current Issues

Q: When will full disk encryption and the FreeBSD bootloader go together?
A: We are still waiting on the patches which add this functionality into the FreeBSD bootloader to get committed upstream in FreeBSD itself.
One of the tickets in phabricator which is still waiting for review/commit is this: https://reviews.freebsd.org/D10512
It references a number of other tickets which seem to imply a large portion of code needs to be refactored first – which is why it is taking a little while.

Q: Will the next installer image have the option to select the graphics driver to be used for installer menu and installation? (As that option was dropped with the last image due to dropping the text-based pre-menu.)
A: Yes, the new installer is intended to be more compatible with a generic fallback driver by default, allowing the user to select a proper drive in the graphical installer. We hope this achieves the same functionality as the text installer by also presenting what we detect as the best option, and allowing the user to override. We also hope this eliminates many of the "I cannot boot the TrueOS installer at all" category of issues.

Q: Is it or will it be possible to take snapshots of specified folders? As far as I've seen, the life preserver only lets me click "snapshot" but not specify so that not all folders are monitored.
A: The life-preserver utility in the background does have the capability to setup snapshot schedules for individual datasets. However, the UI's do not expose that functionality – mainly because if you don't use the "full" snapshot system, then you cannot use the "restore system" function within the TrueOS installer to recover your system later.

Future Work

Q: When will the (Lumina) Window Manager be released?
A: Lumina WM status update:
* Does it build? YES
* Does it run? YES
* Does it manage windows? PARTIALLY
* Does it lock/secure the session? YES
* Does it have a pretty interface? NO (we have not imported the interface features from the current version yet. It just has the background wallpaper stuff right now)

Q: Will you incorporate the Lumina Archiver into Lumina (e.g. easy access via right-click menu with a choice of compression options) like KDE incorporated Ark?
A: Adding special menu options for launching lumina-archiver from the Insight file manager is definitely possible, and would not take too much work. If you make feature request about that on the Lumina github repo, we will try to get that implemented before Lumina 1.3.0 is released.

Q: I asked it before and got the feeling that the answer seemed to be no (but wasn't a definite no to my knowledge), but will "Classic Backup"* ever come back? *aka lumina-archiver takes a snapshot of the home-folder and stores it into a compressed file (or takes it out of such a file) in a way that the user does not have to care about writing writes and so on).
A: We don't have any plans on the "classic" backup ever coming back, and lumina-archiver was written to kind of fill that "gap" in functionality. There were a lot of special hacks in the old "classic backup" system which I really did not want to propagate into the newer systems since they are not "future-proof" solutions.

Source: https://www.trueos.org/blog/trueos-qa/

Among clouds Performance and Reliability is critical

Download syslog-ng Premium Edition product evaluation here

Attend to a free logging tech webinar here

BalaBit IT Security

www.balabit.com

syslog-ng log server

The world's first High-Speed Reliable Logging™ technology

HIGH-SPEED RELIABLE LOGGING

- above 500 000 messages per second
- zero message loss due to the Reliable Log Transfer Protocol™
- trusted log transfer and storage

ZFS Feature Flags

When ZFS was open-sourced by Sun Microsystems (now Oracle) in 2005, it contained a version number, allowing them to add underlying changes to the on-disk structure offering new features, while preventing corruption by old implementations that didn't know how to handle the changes. This simple versioning scheme was strictly copied by the other implementations at the time (mostly FreeBSD). So when Sun added a feature, the version number got incremented, and other implementations copied the code as-is from OpenSolaris.

However after Oracle's acquisition of Sun, in 2010, they closed the source of ZFS, not allowing others to copy new features. Thus, the ZFS developers from Illumos and FreeBSD formed a joint development effort to continue from the last open code, calling it OpenZFS.

In 2012 they decided that, as Oracle would continue developing new features, incrementing the version number, and probably in ways too difficult to implement them compatibly without access to the source code, a solution was needed.

They decided to increment the version number to 5000, in hopes Oracle will never get there, and create a new system for new feature usage and compatibility detection: feature flags.

What are feature flags?

Feature flags are a list of features that can be created by any implementation of ZFS, and can be in three states: disabled, enabled or active. They also have a somewhat undescriptive but schematic name: *feature@<org-name>:<feature>*.

<org-name> is the organization that created the feature, in inverse DNS format, for example, *org.illumos*. *<feature>* is the feature name.

When a feature is disabled, it means that the implementation supports it, but will not use it on the pool (for those that don't know a ZFS pool, it is a collection of disks or other storage media that form a single storage entity).

When it is enabled, it means that the implementation will use it any time, either automatically or by manual user action. But it has not yet done so.

When it is active, it means that the implementation is using it, and changes are made to the underlying on-disk structure.

This way when Oracle's implementation finds ZFS with version number 5000, it will not mount it. Moreover, when feature flags enabled ones (like FreeBSD since 9.2) find it will act accordingly to the feature and its state following these rules:

- If the feature is disabled, it will not be used at all, whether the implementation understands it or not.
- If the feature is enabled and the implementation does not understand it, it will not be used, but ZFS will still work.
- If the feature is enabled and the implementation understands it, it will be used when appropriate, moving its state to active.
- If the feature is active and the implementation does not understand it, it will indicate to the user accordingly and not mount it.
- If the feature is active and the implementation understands it, it will be used as appropriate.

One of the advantages of this system is that any implementation can create a new feature without breaking compatibility or interchangeability of the pool with other implementations, as long as it's not being actively used.

Also, a feature flag can depend on another feature flag. Therefore, when you use one, its dependencies must also be used.

Feature flags explained:

com.delphix:extensible_dataset

This feature expands the underlying on-disk format to allow a more flexible use of its structures by other features. It does nothing by itself except allowing those other features.

It becomes active as soon as any feature depending on it becomes active, and returns to enabled when all users of the depending features have been removed from the pool.

com.delphix:enabled_txg

This feature enables to record the exact moment on which a new feature is enabled. It is only used as a dependency for other features and once it becomes active, it never returns to enable.

com.delphix:async_destroy

When the user destroys a filesystem from the ZFS pool, all of the pool must be traversed to free the blocks it used. Sum a big filesystem and slow storage, and it will take hours. Add a reboot or power outage and the pool won't be mounted until the operation is finished.

This feature adds on-disk data to do that operation asynchronously, in the background, so that the pool can be used while the blocks are being freed.

Also if a reboot happens, the mount is not delayed, and the background operation is restarted where it left.

This feature is only active while a destroy operation is pending, and deactivates itself when it finishes.

com.delphix:empty_bpobj

This feature changes behavior with the references used by snapshots giving a performance increase, and a disk usage reduction in scenarios with a large number of snapshots. Once active, it only returns to enabled when everything created in the pool after activating it is destroyed.

com.joyent:filesystem_limits

This feature allows administrators to set how many filesystems or snapshots can be created under any point in the zpool. Once active, it cannot be deactivated.

org.illumos:lz4_compress

This feature allows setting the lz4 algorithm for compression. It's about 50% faster on writing and 80% faster on reading while giving compressing files about 10% more. The boot pool can use this algorithm, and once this feature is enabled, it becomes active and will never return to enabled.

com.joyent:multi_vdev_crash_dump

This feature allows a dump device to be configured with a pool comprised of multiple vdevs, in any configuration.

com.delphix:spacemap_histogram

This feature enables storing more information about how free space is organized. Once enabled, it will become active when a new free space map is created. Thereafter, it will never return to the enabled state.

com.delphix:bookmarks

This feature allows creation of bookmarks that are like snapshots without holding the data they reference on the disk. They're useful as ZFS send sources. It is active while there are bookmarks in the pool.

com.delphix:hole_birth

This feature improves performance for incremental sends, and receives objects with holes (sections of data that won't get written onto the disk, e.g. blocks filled with zeros), storing information about when the hole was created. It becomes active once enabled.

com.delphix:embedded_data

This feature improves performance and space usage by storing any file that fits in 112 bytes (before or after compression) in the metadata instead of creating a block for it. It becomes active once enabled.

com.delphix:large_blocks

This feature allows using blocks bigger than 128KiB using the recordsize property. It becomes active once a record size is set to anything bigger than that. Moreover, it will only return to enabled once all filesystems that had the recordsize ever increased beyond that are destroyed.

org.illumos:sha512

This feature allows using a truncated version of the SHA-512 hash algorithm as the pool integrity checksum. This hash is about 50% faster on 64-bit hardware than SHA-256, and can be used for deduplication but cannot be used for boot pools. It becomes active once a new entry has been created while it was enabled, and will only return to enabled once all filesystems set to use it are destroyed.

org.illumos:skein

This feature enables using the Skein hash algorithm as the pool integrity checksum. This hash is about 80% faster on 64-bit hardware than SHA-256. It can be used for deduplication but cannot be used for boot pools. It is also salted to prevent hash collision attacks on systems with deduplication. It becomes active once a new entry has been created while it was enabled, and will only return to enabled once all filesystems set to use it are destroyed.

org.illumos:edonr

This feature enables using the Edon-R hash algorithm as the pool integrity checksum. This hash is about 350% faster on 64-bit hardware than SHA-256. It cannot be used for boot pools, and will not be used for deduplication unless verification is enabled.

Furthermore, it is salted to prevent hash collision attacks on systems with deduplication. It becomes active once a new entry has been created while it was enabled, and will only return to enabled once all filesystems set to use it are destroyed.

Recommended features

What features you should enable is a very difficult thing to answer. However, there are some features whose usefulness is beyond doubt.

While you may think you will never destroy filesystems, there comes a day you have to do so. That is when you sit there thinking why you didn't enable *async_destroy* before. Thus, just go and enable it, it will not harm you.

If you use ZFS compression, unless you are using gzip, enable *lz4_compress* and use it. It's faster and better than ZFS's default compression algorithm. Therefore, there is absolutely no reason not to use it.

embedded_data, *hole_birth* and *empty_bpobj* do not harm for being enabled. When needed, just enable them as they give several benefits.

If you ever ZFS send/receive, you should enable bookmarks also.

filesystem_limits, *large_blocks* and *multi_vdev_crash_dump* are very specific. Hence if you don't know whether you require them, then be rest assured you don't.

If you will physically connect the pool to a Linux system, you cannot use *sha512* or *skein*, until ZFS on Linux 0.7.0 gets stable (at the time of this writing, it is not). If not, *skein* gives you enough performance benefits to be a

must-enable, unless you're not using SHA-256 (you are using it only if you have enabled it manually or you are using deduplication).

edonr is not yet supported on FreeBSD 11-CURRENT . So you can forget about it for now on anything but illumos.

If you want to know which feature flags are supported on your system, do *man zpool-features* in a terminal.

Examples

Before using feature flags, you should check whether your ZFS implementation supports them, and which ones are supported. The easiest way to do this is by typing the following command:

root # zpool upgrade -v

This command lists all the supported feature flags as well as the legacy ZFS versions. Then, you can check all the implementations where you use that pool (if it is in a removable media) and choose which ones to enable.

It will also indicate which features are implemented in a read-only compatible way, meaning that when active, the pool can still be accessed to implementations that don't support them, but read-only. If no feature flags are listed, it means that implementation supports none of them.

To enable a feature, you should use:

zpool set feature@<feature_flag>=enabled <pool_name>

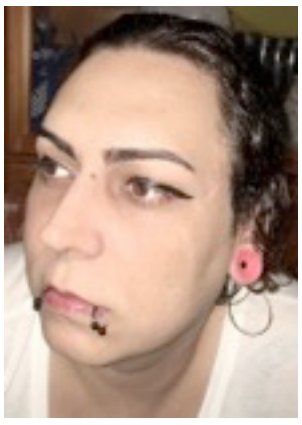
You can omit the organization name. Consequently, enabling the LZ4 compression in a pool called *MyPool* would be:

root # zpool set feature@lz4_compress=enabled MyPool

To see which feature flags are enabled in the pool, type:

root # zpool get all <pool_name>

Supported feature flags will be prepended with a *feature@* and the organization name will be omitted. Unsupported feature flags will be prepended with a *unsupported@,* and depending on where they are marked as *active* or *inactive* , the implementation will not be able to use the pool (or use it in a read-only fashion), or just ignore the unsupported feature, respectively.



About the Author

Natalia Portillo was born in Canary Islands, a transgender girl who is an Independent and an Open-source Developer, a Computer Historian, an Emulator lover, a file system guru, an Apple Certified Macintosh Technician, and a .NET fan.

FREEBSD

SSH Hardening with Google Authenticator and OpenPAM

What is Google Authenticator?

The Google Authenticator project includes implementations of one-time passcode generators for several mobile platforms, as well as a Pluggable Authentication Module (PAM).

Google Authenticator uses the Time-based One-time Password Algorithm (TOTP) which means taking advantages of a key and time to making a secret six digit code. In addition to your password, you'll also need this code to login. The point is you don't need to use the internet; all you need is synchronized time. This technique is called 2-Step Verification.

Two-Factor Authentication Varieties

Two-factor authentication (also known as 2FA) is a method of confirming a user's claimed identity by utilizing a combination of two different components. Two-factor authentication is a type of multi-factor authentication.

There are two types of 2FA:

- **Time-based One-time Password (TOTP).**

In this type of authentication, a verification code is generated by your phone app, a dedicated hardware device, or sent to you via SMS. In this article, we talk about Google authenticator (phone app).

In TOTP, you are not the only one who has the access to the private key. Both the client and the server have access to it too. They know how to create a verification code from it. All we have is a comparison, and this process causes security issues.

- **Universal Second Factor (U2F)**

U2F is an open-authentication standard that enables internet users to securely access any number of online services, with one single device, instantly and with no drivers, or client software needed.U2F was created by Google and Yubico, and is supported from NXP, with the vision to take strong public key crypto to the mass market. U2F uses public key cryptography to verify your

identity. In contrast with TOTP, you are the only one who knows the private key.

Basically, there are two active companies that utilized U2P, SatoshiLabs and Yubico.

SatoshiLabs is private security company which specializes in both hardware and software. All of its projects are related to Bitcoin. It was founded in 2013 and it is headquartered in Prague, Czech Republic. Trezor is a U2F Bitcoin wallet that was made by SatoshiLabs.

Yubico is a private company that was founded in 2007. It has offices in Palo Alto, Seattle, and StockholmYubikey is its USB U2F Token.

You can easily compare the two. Nonetheless, comparison is not a good idea since it depends on the situation.

Sometimes you just can't add a device to your pc Google authenticator by choice.

U2F has not been featured in this article, but we can cover it on the subsequent reads.

What Is Pluggable Authentication Modules (PAM)?

The Pluggable Authentication Modules (PAM) library is a generalized API for authentication-related services which allows a system administrator to add new authentication methods by simply installing new PAM modules, and to modify authentication policies by editing configuration files.

PAM was defined and developed in 1995 by Vipin Samar and Charlie Lai of Sun Microsystems, and has not changed much since. In 1997, the Open Group published the X/Open Single Sign-on (XSSO) preliminary specification which standardized the PAM API and added extensions for a single (or rather integrated) sign-on. At the time of this writing, this specification has not yet been adopted as a standard.

In PAM parlance, the application that uses PAM to authenticate a user is the server, and is identified for configuration purposes by a service name, which is often (but not necessarily) the program's name.

The user requesting authentication is called the applicant, while the user (usually, root) charged with verifying his identity and granting him the requested credentials is called the arbitrator. The server's sequence of operations that goes through to authenticate a user and perform whatever task he requested is known as a PAM transaction. The context within which the server performs the requested task is called a session.

The functionality embodied by PAM is divided into six primitives which have been grouped into four facilities: authentication, account management, session management and password management.

PAM Varieties

There are three common types of PAM:

- **Linux-PAM**: The Linux-PAM used by almost every Linux distributions. However, bear in mind that Linux-PAM is BSD License.

- **OpenPAM**: OpenPAM is a BSD-licensed implementation of PAM used by FreeBSD, NetBSD, DragonFly BSD and OS X (starting with Snow Leopard), and offered as an alternative to Linux PAM in certain Linux distributions. OpenPAM was developed for the FreeBSD Project by Dag-Erling Smørgrav and NAI Labs, the Security Research Division of Network Associates, Inc. under DARPA/SPAWAR contract N66001-01-C-8035 ("CBOSS"), as part of the DARPA CHATS research program.

- **Java™ PAM or JPam**: PAM is basically a standard authentication module supporting Linux and UNIX. JPam acts as a bridge between the Java part and the usual PAM. JPam enables the use of PAM modules or facilities (like *auth*, *account*, *passwd*, *session*, etc.).

- **SolarisPAM**: SolarisPAM is used by Solaris operating system.

How To Add An Extra Layer Of Authentication To FreeBSD With OpenPAM?

OpenPAM was developed for the FreeBSD. /etc/pam.d/ directory contains configuration files for the Pluggable Authentication Modules (PAM) library. Each configuration file details the module chain for a single service, and must be named after that service. If no configuration file is found for a particular service, the /etc/pam.d/other is

used instead. If that file does not exist, /etc/pam.conf is searched for entries matching the specified service or, failing that, the "other" service.

How To Add 2FA On SSH?

First of all, you have to install Google authenticator.

```
#pkg install pam_google_authenticator
```

Then, install QR encoder for ease of use.

```
#pkg install libqrencode
```

QR code (Quick Response Code) is a type of matrix barcode (or two-dimensional barcode) first designed for the automotive industry in Japan. A barcode is a machine-readable optical label that contains information about the item to which it is attached. With QR codes, you can easily transfer your authentication key to your smartphone.

Then, you have to edit **/etc/pam.d/sshd** and add the following line to auth section:

```
auth    required
/usr/local/lib/pam_google_authenticator.so
```

Create a new user called “sara” to test Google authenticator:

Interactively:

```
#adduser sara
```

Manual:

```
#pw useradd -n sara -s /bin/sh -m
```

```
#passwd
```

Now you can easily run Google authenticator and simulate a full login to “sara” username.

```
#su – sara -c google-authenticator
```

You can now see your QR code on screen. With “Authenticator” android app, scan this code and get your verification code on your smartphone.

Google authenticator will ask you a couple of questions:

- Do you want authentication tokens to be time-based? (y/n)
- Do you want me to update your "/root/.google_authenticator" file? (y/n)
- Your chances to notice or even prevent man-in-the-middle attacks (y/n)
- Size of 1:30min to about 4min. Do you want to do so? (y/n)
- Do you want to enable rate-limiting? (y/n)

You can answer all those question with “**y**”.

By default, tokens are good for 30 seconds, and to compensate for possible time-skew between the client and the server, it has been allowed an extra token before and after the current time. If you experience problems with poor time synchronization, you can increase the window from its default. If the computer that you are logging into isn't hardened against brute-force login attempts, you can enable rate-limiting for the authentication module. By default, this limits attackers to no more than three login attempts every 30s.

You must edit the SSH Service configuration file (**/etc/ssh/sshd_config**) , and add these lines for a specific user’s 2FA authentication:

```
Match User sara

     AuthenticationMethods
keyboard-interactive
```

Restart SSH Service:

```
#service sshd restart
```

If “**sara**” is what we used to name the user, create a SSH connection and she will be asked for two factors:

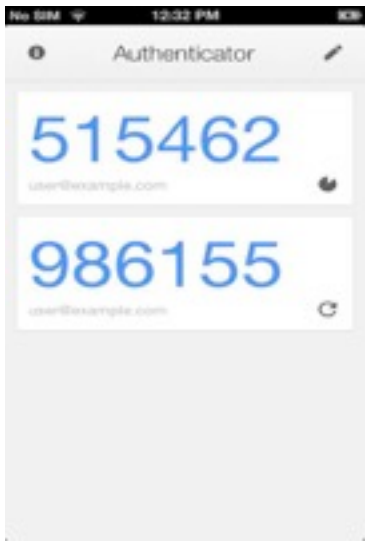
- Password
- Verification code

Also, there are other varieties but it’s not recommended for use:

- publickey + password + verification code
- publickey + verification code, without password

How To Manage Codes Using Your Smartphone.

Google Authenticator is the solution. You can download it from Google play and scan your QR code. Your key will be added to your Google authenticator main page. Every 30 seconds, new 6 digits code will be generated and you will use it as a verification code.



What Is Time Correction For Codes?

Google Authenticator app is Time-Sensitive (Time-based), and if it’s not synced correctly, your code will not work.

To make sure that you have the correct time:

Go to the main menu on the Google Authenticator app.

Click on "Settings".

Click on "Time correction for codes".

Click on "Sync now".

Conclusion

Google Authenticator is compatible with OpenPAM and FreeBSD. You can easily setup 2FA on your FreeBSD machine. With 2FA, the safety of your service is guaranteed and no one can gain access to your shell.

Useful Links

https://www.yubico.com/why-yubico/for-individuals/gmail-for-individuals/

https://www.netbsd.org/docs/guide/en/chap-pam.html

https://www.freebsd.org/doc/en_US.ISO8859-1/articles/pam/

https://trezor.io

http://meetbsd.ir

http://in4bsd.com

About the Author

Abdorrahman Homaei has been working as a software developer since 2000. He has used FreeBSD for more than ten years. He became involved with the meetBSD dot ir, and performed serious training on FreeBSD. He is starting his own company in Feb 2017.

You can visit his site to view his CV: **http://in4bsd.com**

Debugging and Troubleshooting

Debugging/Troubleshooting is a useful skill when you are working in maintaining legacy applications by doing some small incremental changes to an old code base, where the code has been manipulated by so many users over the years, and it is becoming a mess. So the management had decided that the code works as it is, and you are not allowed to change it all over “the right way (tm)”.

In this tutorial, I’m going to represent a real life situation where debugging skills will save us time, headaches and possibly find a solution with a minimal amount of effort.

We are going to debug an old *legacy C application* that acquires plain text files and inserts them to into a database until some new changes made make some dormant “**feature**” available for the users (data is getting truncated and users are complaining, this needs to be fixed ASAP before business closing). In this situation, we will use the classic **"gdb"** debugger.

For debugging, we will use **GDB**. If you don’t have ports collection installed, make sure you have it installed before proceeding. (We need **postgresql** for this tutorial. So, you should use ports if you want an up to- date version of postgresql). Use the following instructions as root:

 # portsnap fetch

Figure 1. Portsnap fetch

When running Portsnap for the first time, extract the snapshot into /usr/ports as follows:

 # portsnap extract

After the first use of ***Portsnap*** has been completed as shown above, /usr/ports can be updated as needed by running:

#portsnap fetch

 # portsnap update

When using fetch, the extract or the update operation may be run consecutively by typing the following command:

 # portsnap fetch update

We require following packages for this tutorial:

postgresql-client

postgresql-server

You could install the 8.4.21 version using pkg[1] as root using the following commands:

pkg install postgresql84-client-8.4.21

pkg install postgresql84-server-8.4.21_1

Or as we have ports already installed, go to the needed software folder and as root, do the usual and install.

For detailed instructions of installing and configuring ***postgresql***, you should read one of the guides from the official site: < https://wiki.postgresql.org/wiki/Detailed_installation_guides#FreeBSD >.

Also, we will use some test data from < http://www.briandunning.com/sample-data/ > to execute the examples in this tutorial. We will need to download the US 500 sample (this one is free). You should rename this file to ***us.csv*** for this tutorial purposes.

Once you have *postgresql* installed, you need to create the ***us table*** using the us.sql.

The Incident (a.k.a Production Down)

You have received an email stating that the current process for adding new clients has stopped working and nobody knows why. All the management knows is that the file must be loaded to the database before business closure or they will have to give the top management team a serious explanation about what happened and how they will avoid this occurrence in future. So to avoid all this stress, you have been selected to fix this problem right away, and before anyone knows what is actually happening (that means you need to work all night as needed).

Therefore, let's act as fast as possible. We don't want to spend our nap time debugging an old application that we really don't want to touch as there is no documentation nor the original programmers are available. The program as far as you know was a product of a joint venture between contractors of different nationalities, and also, the code and comments were written in Spanish.

Let's start by running the program; it takes as a parameter a file name where the client data is:

```
[cneira@trueos] /home/cneira/workshop/Intro/source# ./update_clients ../data/us.csv
digest es :72c8a1214e9cb1e770213b206a07d475b82b30b1
501 registros insertados
[cneira@trueos] /home/cneira/workshop/Intro/source#
```

Figure 1. Trying to update clients using the us.csv file

All seems fine. The program does what it was supposed to do, false alarm again just 30 minutes to go home. Just to be sure, I'll check the table to see if all the data is in there. After all, that is the issue which was reported.

```
[cneira@trueos] /home/cneira/workshop/Intro/source# psql -U root -d cruce
psql (8.4.21)
Type "help" for help.

cruce=# select count (*) from us;
 count
-------
     0
(1 row)
```

Figure 2. Checking if clients present in the us.csv file are in the database

Tough luck, there is an issue here. I'll fetch the source code and fire up **gdb.** I have better things to do than debug old code all night and according to the management, this one must be fixed before the next run as the users are inserting the data manually. The source code fortunately was still on the backups. Thus, I created a minimal "**makefile**" for this. I needed it to compile and let the compiler put all the debugging symbols required in the object file for an "easy" debug session.

```
#This one updates the DB with the clients taken from a plain text file
ALL     :       update_clients
update_clients: update_clients.o
        cc -ggdb update_clients.o -L /usr/local/lib -lpq -o update_clients

update_clients.o : update_clients.c
        cc -ggdb -c update_clients.c  -I /usr/local/include

clean   :
          rm  *.o
          rm  update_clients
```

Figure 3. Simple makefile to start debugging

the "-ggdb" flag is an old **gcc** flag that does the following according to the official manual[2]:

"*When the user specifies -ggdb, GCC normally also uses the value of this macro to select the debugging output format, but with two exceptions. If DWARF2_DEBUGGING_INFO is defined, GCC uses the value DWARF2_DEBUG. Otherwise, if DBX_DEBUGGING_INFO is defined, GCC uses DBX_DEBUG.*"

If we are using clang, this does not matter as you can use the -g flag "Generate complete debug info". Let's compile it and see what happens:

Figure 4. Makefile output

Well, at least it compiles. Otherwise, it could have been worse at this time. Now that the debug symbols are in there, let's try setting some breakpoints to identify the problem at hand. Looking at the source code, the first obvious breakpoint must be set in the insert function call:

Figure 5. Setting first breakpoint

I'll start a debugging session, and then pass a parameter, the us.csv file, to the program as follows:

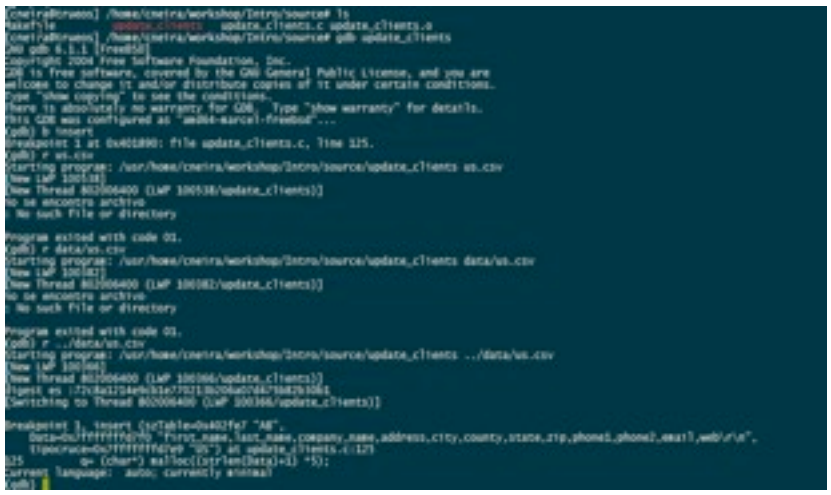


Figure 6. Running gdb using us.csv as parameter to the update _ clients program

To start a debugging session, type the following command:

gdb <program name>

This will take us to a **(gdb)** prompt where we could use all the commands available in the GDB debugger. If we were to debug a running program, we should type:

(gdb) attach <pid of running program>

In doing so, it takes us to the same prompt again. But take note that in this case, it will cause the world to stop for the running program until we let it complete in our debug session. Also if we had a core dump, we could check the stack trace, but in this case we don't have a ***coredump*** to check the stack trace. First, I'll set a breakpoint using the break command and I could just type "b" as a short form. If we need some help with a command, type the following command as usual:

(gdb) help <command>

For example, by typing "**help break",** the screen, Figure 7, will be presented to us:

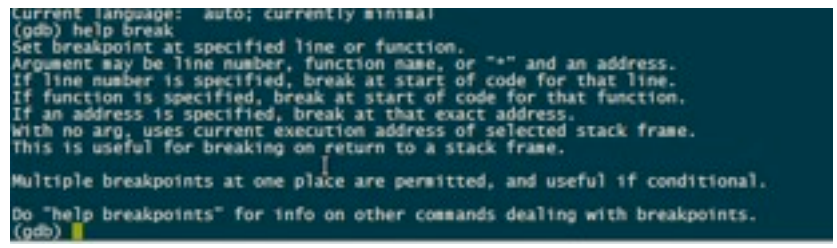


Figure 7. Running gdb help command

Now, I have my breakpoint ready at the insert function. I'll run the program and monitor what will be happening at runtime. I run the program by typing:

(gdb) r ../data/us.csv

Where "r" is the abbreviated form of run.

You could pass the parameters to the program next to the command. In this case, this program only takes one that is the file containing client data (../data/us.csv):

Figure 8. Running gdb stopping at a breakpoint

Since I'm lucky to have the source code, I can use the *win command* to display the source code and the exact line where the execution has stopped as illustrated in Figure 9:

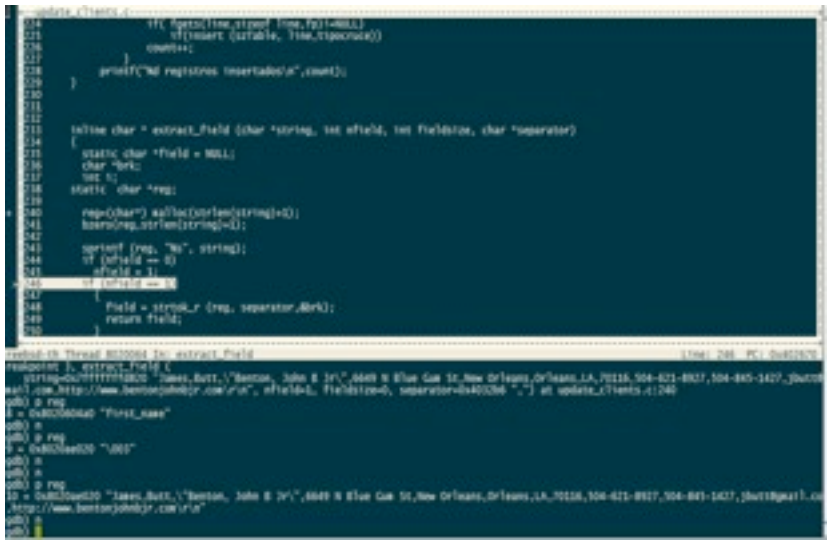


Figure 9. Running gdb using the win command

Then, we need to check the value of the input parameter data to the insert function. To do this, just type:

(gdb) display Data

This command will print every time we hit a breakpoint, the value of the Data variable, as long as the breakpoint is in within the scope of this variable as follows:

Figure 10. Running gdb using display command

All seems **OK** with the data and the functions that make up the sql statement. So, we need to check where the SQL statement is executed. Hence, we will put a break at the query(char*) function, looking at the documentation for the libpq library[3], It seems not enough to check for NULL. Therefore, to know what the database tells us about the result of each transaction, we will use the following functions:

PQresultErrorMessage

Returns the error message associated with the query or an empty string if there was no error. const char *PQresultErrorMessage(PGresult *res);

PQresultStatus

Returns the result status of the query. *PQresult- Status* can return one of the following values:
PGRES_EMPTY_QUERY,
PGRES_COMMAND_OK, /* the query was a command returning no data */
PGRES_TUPLES_OK, /* the query successfully returned tuples */
PGRES_COPY_OUT,
PGRES_COPY_IN,
PGRES_BAD_RESPONSE, /* an unexpected response was received */
PGRES_NONFATAL_ERROR,
PGRES_FATAL_ERROR

Figure 11. Running gdb stopping at a breakpoint

```
void
query (char *qStr)
{

    PQclear (res);

    res = PQexec (conn, qStr);
if( (PQresultStatus(res) != PGRES_COMMAND_OK) )
    {
      fprintf (stderr, "Error: res==NULL\n");
      fprintf (stderr, "Query: %s\n\n", qStr);
      fprintf(stderr, "error <%s>", PQresultErrorMessage(res));
      PQclear (res);
      exit_nicely (conn);
    }
}
```

Figure 12. Code snipet for the Query function

Let's run it again, the result will be as follows:

Figure 13. Running gdb again

Where did that come from? Looks like somebody implemented a function that tries to insert a hash but never worked.

By looking at the file, when it was created, they never got rid of the header from the csv file as shown:

Figure 14. Running gdb query returns error

Let's remove that in the file, and try again. At this instant, we have another problem according to the table of data types shown as follows:

Figure 15. Running gdb SQL error

It makes no sense that "LA" is being considered as it were a zip code.

COLUMN_NAME	TYPE_N...	IS_NULLA...	DECIMAL_DIGI...	COLUMN_...	COLUMN...	REMA...	DATA_...
first_name	varchar	NO	0	2147483647	<null>	<null>	12
last_name	varchar	NO	0	2147483647	<null>	<null>	12
company_name	varchar	NO	0	2147483647	<null>	<null>	12
address	varchar	NO	0	2147483647	<null>	<null>	12
city	varchar	NO	0	2147483647	<null>	<null>	12
county	varchar	NO	0	2147483647	<null>	<null>	12
state	varchar	NO	0	2147483647	<null>	<null>	12
zip	int8	NO	0	19	<null>	<null>	-5
phone1	varchar	NO	0	2147483647	<null>	<null>	12
phone2	varchar	NO	0	2147483647	<null>	<null>	12
email	varchar	NO	0	2147483647	<null>	<null>	12
web	varchar	NO	0	2147483647	<null>	<null>	12

Figure 16. Data types

Then, I'll set a breakpoint in the *extract_field* function when it tries to extract the value for the 7th field. I don't want to wait for all fields to be processed, so I'll set a condition in the breakpoint as follows:

(gdb) b if nfield == 7

Figure 17. extract_field

I'll type c (short form of continue), to resume the program execution. Thereafter, scrutinize every instruction that forms the program, and check the values for the local variables. In this case, I stopped at the *extract_field* function as I'm checking why the "**LA**" value is being considered as a zip code (integer).

Figure 18. checking the values for the local variables

And to display the variable value which I'm interested in during this debug session, just by typing:

display <variable> will do. For example, in this case:

(gdb) display field

(gdb) display nfield

These are variables that exist within the scope of the *extract_field* function. If I don't want to display any one of the values anymore, I just have to type: **undisplay <variable>**.

This is an interesting extraction of the field at position **7** and the value is "**Orleans**". However, according to the data, the value should be "LA". So, ***what is the problem?*** It seems the *extract_field* has a bug since it is off by a field. I'll check the ***backtrace*** to remind me how I got to this point by typing **"bt". This command** shows me the backtrace.

A backtrace is a summary of how your program got where it is. It shows one line per frame, for many frames, starting with the currently executing frame (frame zero), followed by its caller (frame one), and on up the stack.

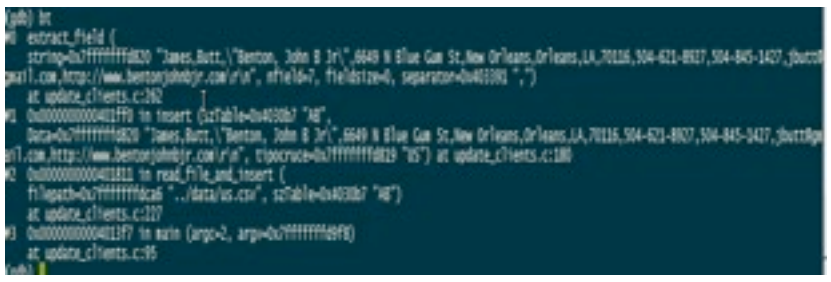


Figure 19. extract_field back trace

There it is, the whole string being tokenize by the extract field function is "James,Butt....". As we see in frame 0, the interesting part is the argument separator = "," and the third field of the string : \"Benton, John B Jr\".

The bug there is a semicolon in the third field causing the "LA" value to be considered as a zip code.

Let's fix the code at runtime. I'll set a breakpoint at line 242 and replace the string \"Benton, John B Jr\" with \"Benton John B Jr"\ and see how it goes.

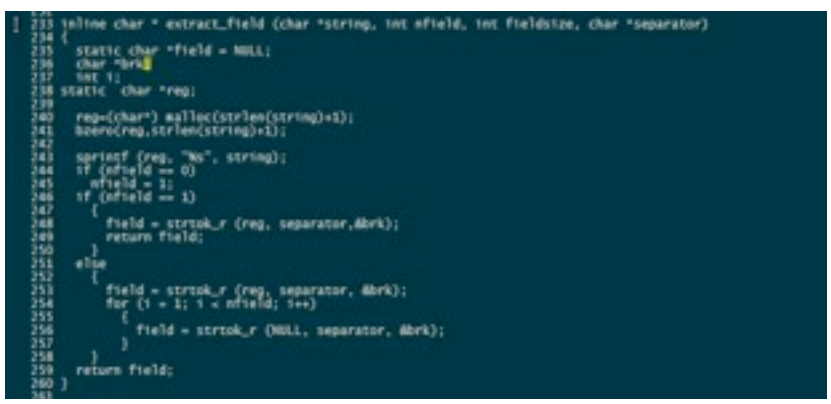


Figure 20. Detecting the error

To set a breakpoint at a specific line of code, use the following commands: as in **(gdb) b <source.c>:<line number>**

(gdb) b update_clientes.c:242.

Now, let's run this again:

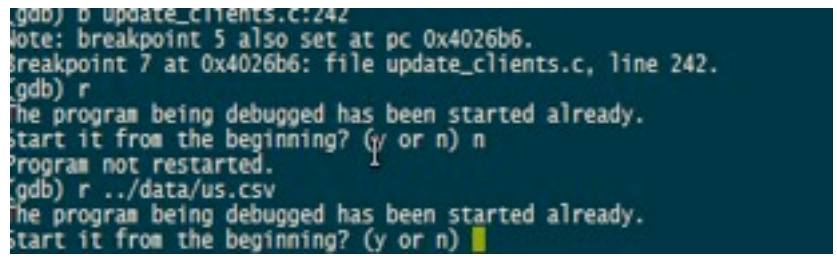


Figure 21. Running again after correcting the error

I had set some breakpoints. I'll delete them by typing d and the breakpoint number. To know which breakpoints, we need to type the following command:

(gdb) i b

That is the short form of breakpoints' information:

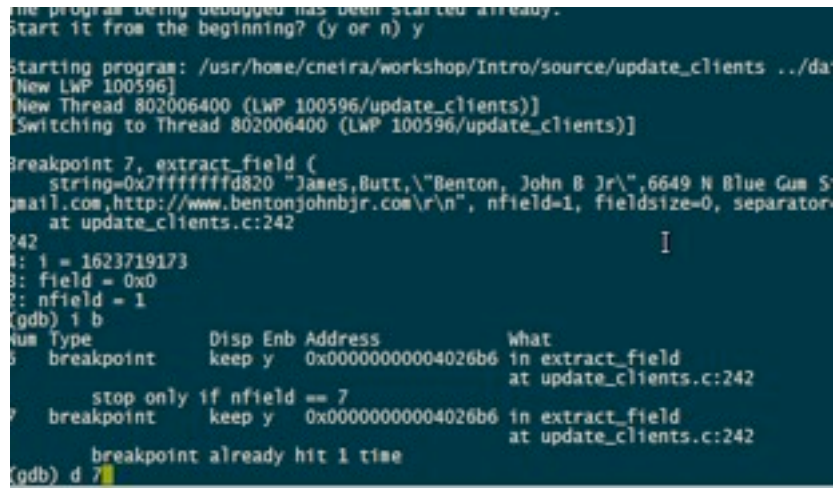


Figure 22. Displaying the breakpoints info.

Now, I'll replace the actual string value with the one I want by typing the following command:

(gdb) set <variable> <value>

Figure 23. Replacing actual string value

I have changed the value, but still it is failing. That is because I picked the wrong breakpoint to change the data. We should have set a breakpoint at line 121 where the string is being passed as a parameter to the function that calls this subroutine. ***Let's fix this and go home!***

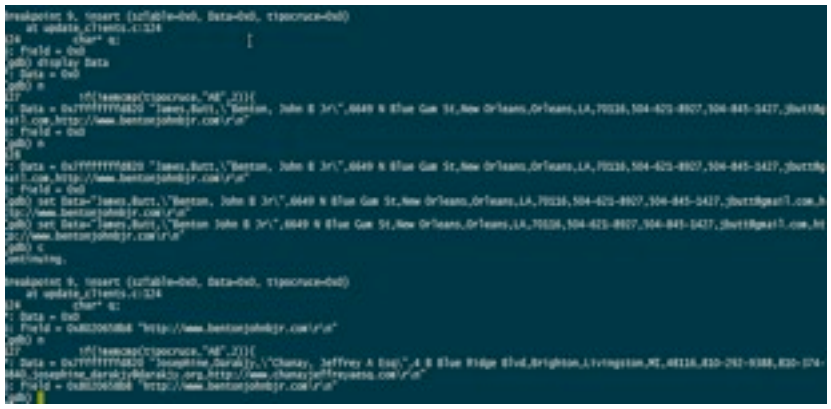


Figure 24. Re-setting the breakpoint

It worked for the record, we have modified (no SQL error at insert), but here comes another register with the same issue as the one we have corrected. At least, we know the fix we applied at the runtime made it work. At this point, it is just a matter of modifying and reprocessing the data.

All this seems easy at this point, but wait! What if we are not allowed to modify the source code to make the necessary changes for this to work out? What if we don't have the source code to debug?

In that scenario, we could use some library interposers and add a wrapper around the function that is causing us trouble. We could also use gdb to debug even if we don't have the source code available.

Conclusions

The GNU debugger (**gdb**) is a powerful tool that gives you an edge over other debuggers when troubleshooting an application. As a developer, you should be proficient when using it so as to enjoy its innumerable advantages.

References

"31.3. Command Execution Functions." *PostgreSQL: Documentation: 9.1: Command Execution Functions*. Web. 08 Aug. 2014. <http://www.postgresql.org/docs/9.1/static/libpq-exec.html>.

"5.5. Using the Ports Collection." *5.5. Using the Ports Collection*. Web. 06 Aug. 2014. <http://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/ports-using.html>.

"All Debuggers - GNU Compiler Collection (GCC) Internals." *All Debuggers - GNU Compiler Collection (GCC) Internals*. Web. 06 Aug. 2014. <https://gcc.gnu.org/onlinedocs/gcc-4.7.4/gccint/All-Debuggers.html>.

"Clang 3.6 Documentation." *Clang Compiler User's Manual —*. Web. 06 Aug. 2014. <http://clang.llvm.org/docs/UsersManual.html#cmdoption-g>.

"Debugging with GDB - Backtrace." *Debugging with GDB - Backtrace*. Web. 10 Aug. 2014. <https://ftp.gnu.org/old-gnu/Manuals/gdb-5.1.1/html_node/gdb_42.html>.

"Detailed Installation Guides." - *PostgreSQL Wiki*. Web. 10 Aug. 2014. <https://wiki.postgresql.org/wiki/Detailed_installation_guides#FreeBSD>.

About the Author

Carlos Antonio Neira Bustos has worked several years as a C/C++ developer and kernel porting and debugging enterprise legacy applications. He is currently employed as a C developer under Z/OS, debugging and troubleshooting legacy applications for a global financial company. Also he is engaged in independent research on affective computing. In his free time, he contributes to the PC-BSD project and enjoys metal detecting.

Dr.WEB®
CureIt!™

EMERGENCY CURING

for Windows workstations and servers

including those running other anti-virus software

FUNCTIONS:

- Cures Windows workstations and servers.
- Verifies the quality of the anti-virus software currently in use.

FEATURES:

- Dr.Web CureIt! doesn't require installation and doesn't conflict with any known anti-virus; consequently there is no need to disable the anti-virus currently in use to check a system with Dr.Web CureIt!.
- Improved self-protection and an enhanced mode for more efficient countermeasures against Windows blockers.
- Dr.Web CureIt! is updated at least once an hour.
- The utility can be launched from removable media including USB storage devices.

LICENSING FEATURES:

The utility is available for free when used for non-business purposes.

Can DevOps Really Be Defined?

DevOps is a way of life for people with the right mindset to embrace the culture of collaboration while scientifically automating the continuous delivery of software features with the rigor and discipline of continuous integration and a passion for continuous testing while using the power of standardized tooling to constantly monitor everything being done. Huh? You say? Well this is a definition that I compiled based upon myriad interpretations that I came across in my interactions with my peers as well as customers, partners and service providers, not to forget my colleagues over social media. The interpretations could be many but the terminology is consistent. A word-cloud around all these interactions surfaces the usual suspects quickly. Let me highlight each aspect that contributes to this definition. And, by the way, feel free to tweak the definition as you deem fit, your comments are welcome!

Culture of the enterprise. “You can use as much DevOps as you want”, was the quote I heard during the Gartner ITXPO Symposium last fall from one of my co-presenters. DevOps is about striking a balance between the desire for agility against the need for stability. The culture of the enterprise at large can strongly influence which way the balance tilts and by how much. Culture is also influenced by market forces, change of leadership and employee behavior. Which brings us to people.

The Mindset of people. It all comes down to the people. There are segments of the workforce who may be very comfortable doing what they have been doing for years together and may be less prone to change. On the other hand, there could be a whole other segment who are all about introducing new paradigms and business models enabled by technology solutions perceived to be cool and enticing. Fundamental concepts like team-play and the willingness to work as a team towards a common goal are best driven by people with the right mindset, a mindset of collaboration.

The Art of Collaboration. Collaboration requires the workforce to reach across the table and put themselves in the shoes of the very individuals they are dealing with. Development teams need to think ahead and take proactive steps to ensure that the management and operations of what they are building is smooth, robust and stable. Operations teams must respect the need for the rapid injection of consumer driven features. Both teams must collaborate and have an open-exchange of information with the common goal of IT meeting the expectations of the business users. And the people concerned with IT must work together to inject the right levels of automation towards this common goal.

The Science of Automation. Automation is not just about using tools to do repeated tasks. The science of doing automation right is all about ensuring that the right processes are being well executed in the first place. Automation of the wrong processes or processes being executed the wrong way only proliferates more problems. The Science of Automation can also be applied to business processes too. Automation must be done in increments across logical subsets of process steps that are part of a continuous engine.

Discipline of Continuous Integration. DevOps is a way of life. It is something that is done in continuum like a smoothly operating engine in constant motion. This spirit of continuity applies to the integration of isolated changes to the larger code base on a daily (or more frequent) basis for a build to be effected successfully

with each change. Active collaboration is a key catalyst to have developers frequently integrate their work with other developers. This promotes early detection of problems, just what the Testing team ordered!

Passion for Continuous Testing. This is a term that I have not encountered. CI/CD. Got that. What about CT? In the spirit of collaboration that is the hallmark of DevOps mindset, testing is everyone's responsibility. To fail-fast, testing must begin at an early stage in the life cycle starting with software requirements, the architecture, design, etc with source code reviews and unit testing by developers to deliver error-free code along with test data sets. With the common goal of delivering a timely and meaningful solution, development and operations must work together to configure the testing environment to be as close to the production environment. And while we are at it, testing is a fine process when it automated. With meaningful automation and relevant test data sets, regression testing can almost become a perfect science -- which is what it would take to address the need for Continuous Delivery.

In the words of one commentator, "continuous delivery is nothing but taking this concept of continuous integration to the next step."Instead of ending at the door of the development lab, continuous integration in DevOps extends to the entire release chain: including QA and operations. The result is that individual releases are far less complex and come out much more frequently.

The actual release frequency varies greatly depending on the company's legacy and goals. For example, one Fortune 100 company improved its release cycle from once a year to once a quarter—a release rate that seems glacial compared to the hundreds of releases an hour achieved by Amazon.

Exactly what gets released varies as well. In some organizations, QA and operations triage potential releases: many go directly to users, some go back to development, and a few simply are not deployed at all. Other companies—Flickr is a notable example—push everything that comes from developers out to users and count on real-time monitoring and rapid remediation to minimize the impact of the rare failure. Need for Continuous Delivery. The concept of Continuous Delivery can be best described using healthy eating habits analogy. All too often, I have heard about eating smaller portions more frequently than large meals spaced wide apart. It would almost seem like Enterprise IT -- and by consequence, the business -- is looking for more frequent and continuous release of new features very fast. To the extent that they are also willing to accept the potential downside of occasional hiccups as long as they are fixed swiftly. The steady stream of new features is a significant shift in mindset that has permeated to the business. Is business going DevOps?

System of Continuous Monitoring. The only way to effectively inject the fail-fast mindset with rapid-fire releases of features is through continuous monitoring across the lifecycle from development to operations. A challenge very often encountered is the proliferation of environments and platforms that need to be monitored. The only way to combat this rising force of technology proliferation is through ruthless standardization of the applications, platforms -- and yes, tools.

The power of standardized tooling. And finally, here we are. Tools. Yes, we absolutely need the tools to do many of the activities discussed above. However, tools are not the first thing to be addressed when it comes to DevOps. Also, standardization of tools goes a long way in simplifying the business of IT while injecting healthy levels of purposeful automation with reusable processes.

About the Author

E.G. Nadhan is an Innovative IT Strategist for Red Hat driving Digital Transformation with emerging technologies and DevOps | @NadhanEG. He has over 25+ years of experience in the IT industry selling, delivering and managing enterprise solutions for global enterprises. As the Chief Technology Strategist for the Central Region at Red Hat, he works with the executive leadership of enterprises to innovatively drive Digital Transformation with a healthy blend of emerging solutions and a DevOps mindset. He also provides thoughtful leadership on various concepts including Cloud, Big Data, Analytics and the Internet of Things (IoT) through multiple channels including industry conferences, Executive Round tables as well as customer specific Executive Briefing sessions.
Specialties: Business Innovation, Technology Innovation, Cloud Computing, DevOps, Open Source, Social Media engagement, and Applications Transformation.

WannaCry / Ransomware

I will try to explain in "a simple way" what happened (and will happen again) with WannaCry / Ransomware / incidents of the last weeks. This is a text for non-technical people or laymen, may be the affected and victims.

All electronic equipment (telephones, tablets, laptops, PCs, Webcams etc.) consist of hardware part(s) and software that controls the physical components.

For instance, a phone is a physical device, but when switched on, it runs an Operating System or a Firmware that allows us to interact with it, interpreting the touches on the screen, buttons and reacts by executing the user's commands.

For those who are using Windows, Windows is the operating system whereas for Apple products, OSX is the operating system. Android is for those using Android phones and to add on this list is Linux. Therefore, all of us use an operating system, with or without our knowledge.

The benefits of using an operating system are that it can be programmed or changed, upgraded to improve its functions or fix errors. If this does not exist, the useful life of a computer would be reduced. In addition, in case of a detected problem, we must replace the physical equipment with a new one.

For example, when someone says, "I have updated my phone", it means that the phone manufacturer sent another O.S. update, and he/she accepted to upgrade/replace the existing phone software.

In addition to the O.S upgrades, there are application programs which run on the operating systems. For example, programs to read and send emails, internet browsers, Facebook clients, WhatsApp, Google maps, etc. All these are some of the useful tools which are supported by your operating system to make your live easier. Besides the O.S, they too require updates.

Therefore, the companies which develop these programs send updates or new versions, exactly the same as the O.S. Manufacturers do. This is done in a bid to patch, update, replace or fix errors, sometimes the updates are an addition of more features.

Having said that, let us now understand the "ugly sad" part about these system upgrades.

All operating systems and all application programs were developed by humans (yes, Coders made up of experienced software professionals who may appear as "semigods". However, they are still humans) who had oversights or did not contemplate on certain critical situations of use when creating these software.

Many times, we have experienced a program closing by itself, show an error or directly hangs the phone and we are forced to restart it. Sometimes it is a program and other times it could be the operating system.

Those who have been using Windows for years will remember the phrase, "restart Windows every two or three days". This action was necessary to clean up the device's memory.

Nothing is perfect, and just as certain batteries exploded because of manufacturing defects, programs aren't the exception, they too fail if something in particular occurs (certain conditions or combinations of actions).

This is not new. As a matter of fact, for many years, companies build sites where you can report any errors to them and they will try to fix it. In some cases, they even pay the informants for their mistakes.

In the case of the proprietary companies i.e. Microsoft, Apple and many others, they usually try to cover or hide the errors since they charged for a product that has failures, each one represents an error or a problem for its brand, an increase of repair cost, a fall of their image, etc.

As these companies earn money through the sale of their products, it is better for them that their devices do not have so many defects. There is a cost associated with fixing those problems that in the end reduce their profits in millions. In addition to selling the right to use a program, they are still the owners and cannot evade taking a share of the blame when a program or a physical component fails.

In the GPL / Linux Open world, bugs are better well received because they help to improve the product we are all using. We are looking for better quality products. Here, there is no cost of use or license. Thus, each error is fixed because its central feature is to be a good APP, and maintain the quality of each program.

Now to an Even Darker Part

As has been reported by Edward Snowden, Chelsea Manning, Richard Stallman, Julian Assange, Wikileaks, Anonymous and many other professionals / technology sites, the U.S. Government (and certainly others) has also purchased and internally developed programs to "Use" the errors arising from O.S. and programs in their benefit. Instead of reporting it to Microsoft, Apple and other companies, they keep the knowledge of "if I do this and this, I can have access to the remote Windows computer". This is called "exploiting the vulnerability" of a component, or may be intrusion too.

There is a market of programs that take advantage of the mistakes made by other programs. There exist big (and bad) companies that sell these intrusion programs in millions of Euros. Also, groups of dark hackers in the world and even the governments are key players in this market. This insecurity is a big business because it allows to have control over other teams „without your permissions”, hang them or destroy/disable their equipment.

There is evidence that the NSA and other government offices around the world have purchased such programs from law firms, whether legally or not. They have succeeded in developing the so-called "cybernetic weapons".

For example, suppose one country in the Middle East uses Windows on their computers and wants to make enriched Uranium.

The United States does not like this idea and decides to send them a virus / trojan / malware program to infect their PCs. The detrimental effect on those computers with Windows as their O.S. could be that they restart every hour, or their date/time setting is changed permanently or a blue screen appears with a "hungup" message.

That has happened before and is called cyber-attack. Search the term ”Stuxnet” on the internet and see the extent of damage caused by this malicious computer worm on Iran’s nuclear program.

At the same time, there may be groups of hackers paid by China, Russia, Germany, UK or "freelancers" who are mandated to develop a program to sabotage certain others teams’ industrial systems (or all of them) or everybody so as to make money, lower stock market shares etc. The economical or technological damage is for some players while the benefit for others.

I think there are three motivations for this type of action, political/govern motivations, personal/economic or simply to seek fame/recognition for a while.

For economic reasons, sometimes "the dark forces" pay more and faster than "the good ones". There is no data collected by the cybercriminals of WannaCry. However, it is much more than what they earned receiving payments in Bitcoins than they would have by sending the errors to Microsoft for their settlement.

If these "bad players" are not being supported by governments, then any danger they pose is often called cyber terrorism. This is like placing a bomb in a mall to generate fear among shoppers.

If the governments support them and the action is pretty the same, it is called National Security.

Nationalist, religious, or moral beliefs can make all these groups to be seen as either good or bad. I will choose not to express my opinion on this matter.

Origin of the incident of 12 May

A few months ago (who knows when), a group of hackers was able to obtain a copy of all the viruses / trojans / malware programs from the NSA (its cyber-armament repositories). The parallel action would be to enter a

complex investigation of the contagious diseases, and to steal the samples of all the viruses they have in storage (in many cases without a vaccine).

Those who had these viruses kept them for investigation, prepare a vaccine or for a biological attack, preparing a bacteriological attacks to others.

That group of hackers offered to make public all the viruses they had found, and requested a ransom to sell/deliver them on the internet, charging Bitcoins in return. They began to release parts of what they had found.

And at one point of time, they made public all the information of the viruses they had stolen from the NSA.

Perhaps, they themselves or other hacking groups, obtained those viruses, mixed them with other diseases, and sent them worldwide through many emails, generally as email attachments, "photo links" / pdfs or links to infected websites. It seems that they managed to reach Telefónica in Spain and Latin America, British health systems, banks and businesses of all kinds. And like biological viruses, these computer viruses infected hundreds of thousands of PCs in the world, be it in small companies, homes, universities and development teams, they were all affected.

In this case, they added a component to ask for money from the infected PC owners or organizations. The virus infected computers with Windows O.S, encrypted certain files (blocks access to the victims' data), and asks for money from its owner to decrypt it. This type of malicious software is called a RansomWare (The same scenario is featured in a series called MrRobot). A premonition that we all saw and couldn't stop it.

The Attack

Under certain conditions, all the infected computers (maybe months ago) were "activated" and began to spread the virus to their neighbors. The epidemic was triggered and several companies sent their employees to the streets without their equipment, to prevent the spread of the virus to other equipment at home, like a quarantine measure.

The program also encrypts the files and presents a screen to request a "rescue" of about 300 USD. For the acquaintances that accepted their offer and sent the money, they were sent passwords to dencrypt all their files.

However, the total number of infected P.C's will never be known because it has proved almost impossible to estimate the number.

Many people have reinstalled Windows, recovered files, paid for the vaccine, and there is no report which comprehensively accounts the findings of this incident.

This happened to Windows computers OS, but it can also happen to Apple, Linux products, and worst of all is that it could happen again.

Thanks to the ethical hackers and security companies who understood what was happening, tried solutions and also informed Microsoft. They could stop the virus and prepare vaccines for it.

Global security researchers and consultants have verified that in this case, several components of the virus were in the armament, stolen from the NSA but with a mix of other programs that were used to encrypt files and request for money. A direct relation was verified between the NSA and WannaCry.

Who is Responsible?

For political, religious, human, or social reasons, the NSA can be accused of building, paying, and storage of a virus sample for their benefit, without notifying Microsoft (in this case) of the vulnerabilities. They obtained a benefit to attacks others.

We can also hold Microsoft to account for its proprietary policies and obscurantism in how to manage its security, its proprietary products and for "allowing" errors to be made without fixing their products. Or, these errors were not fixed immediately which created a loophole.

We can as well accuse the group of "bad" hackers who stole the bacteriological / computer equipment from the NSA. This is a robbery on the premises of a government entity that is dedicated to "Security".

Moreover, the people and companies who used Windows O.S and trusted their security and patch systems also share a 2% of the blame. Surely, they have heard hundreds of times the problems/vulnerabilities associated with an Operating System, how could they trust that nothing would happen to them? May be they did not create backups, they did not install an antivirus (something that would not be a solution like this case), they did not keep their products updated, etc. Okay, their actions could be insignificant but they are responsible too.

Can This Happen Again?

Yes, definitely. It can happen again. So, the question is when. Can it happen to Apple products? Yes, perfectly.

And using Linux? Yes, it could happen. It's not a perfect world, and no O.S. is ever immune from attacks.

Statistics confirm that 70% of computers have Windows, 25% for Apple and 5% for Linux distributions. Personally, those numbers represent the percentages that the attack could happen again.

Groups that are looking for an economic gain or governments seeking a strategic/economic benefit can return to something similar.

I think until today, Government agencies are looking for new "tools" to gain access to other computers, mobile phones, IoT, Servers etc. It is a never ending story.

Is This The End of Technology As We Know It?

I guess no. We still have to promote the cooperative work, leave proprietary software and trust more in EFF/GPL, build better defenses, and be prepared for a "worst wave" in the future.

Security managers not only have to understand the world but also to seek ways on how to solve future issues more efficiently instead of kicking the balls off the court.

About the Author

Daniel Cialdella Converti has been working as a DBA during the last 22 years, the last 4 for a big company based in Switzerland and UK, on Linux and Windows, and different Engines. He has been servicing countries in America and E.U. on two principal paths, maintaining what they have and moving the company to the latest technologies and products. Linux and I.T. lover.

Static Sites Alongside Dokku on Digital Ocean

I host most of my dynamic sites with Dokku on Digital Ocean, through their one-click install.

Dokku is basically a self-hosted Heroku, letting you run all your Sinatra, Rails, Phoenix or what-have-you apps containerized with little hassle.

Also, I also have some static sites though, I like this blog.

You can get static sites on Dokku. However, it feels a bit bloated to me, adding some magic files and then building a container when you could just deploy by copying the files. A simple static site takes some 15 seconds to deploy via Dokku, vs. 1 second via rsync or scp.

So instead, I hosted them alongside Dokku, using the same Nginx that fronts Dokku.

Find a place on the server

If you set up Dokku with Digital Ocean's one-click installer, you will have a dokku user home directory (/home/dokku) and a root user home directory (/root).

Neither seemed appropriate to store the static sites. Dokku makes assumptions about files stored in the dokku home directory. And Nginx (the www-data user it runs as) can't access stuff under /root. Never mind any security implications of using the root account for this.

So, I created a new user named static:

```
adduser static
```

Enter some password (and store it away) when prompted. Thereafter, accept the defaults for the other fields.

Then, I add the static files somewhere under that user's home directory, like */home/static/sites/my-site.com*.

Configure Nginx

Dokku adds some stuff to Nginx – we can too, without conflict.

Thanks to a suggestion by Mikkel Malmberg, we can set up a single piece of Nginx configuration that should cover most static sites.

Create a /etc/nginx/conf.d/static_sites.conf file (incidentally, right next to Dokku's dokku.conf):

```
/etc/nginx/conf.d/static_sites.conf
```

```
server {
  server_name ~^(?<domain>.+)$;
  root /home/static/sites/$domain;

  access_log /var/log/nginx/$domain-static-
access.log;

  # error_log can't contain variables, so
we'll have to share: http://
serverfault.com/a/644898
  error_log /var/log/nginx/static-
error.log;
}
```

If you like, you can verify that there are no errors in the configuration:

```
sudo nginx -t -c /etc/nginx/nginx.conf
```

Then, restart Nginx so that it picks up this addition:

```
sudo service nginx restart
```

This configuration uses regular expression server names to automatically map any hostname to /home/static/sites/<hostname>. So in the spirit of Dokku, adding new sites requires a minimum of setup. Just point the DNS to the server, add the static files to a directory named like the hostname, and you're done.

If you need a custom Nginx configuration for one site, add a file to /etc/nginx/sites-enabled with that configuration and restart Nginx like above. Nginx is smart enough to prefer exact hostname matches when they exist, and fall back otherwise.

About the Author

Henrik Nyh is a Swedish web developer and he blogs at The Pug Automatic. He works at Auctionet.com. Contact him at @henrik or henrik@nyh.se.

Using Docker Swarm with MySQL Database

Docker Swarm, a cluster manager for Docker hosts, exposes a pool of hosts as a single "virtual" host. Docker Swarm is used to deploy Docker containers on a distributed Cluster instead of a single host. In an earlier tutorial, we discussed about creating a Docker Swarm Cluster for Oracle Database. In this tutorial, we shall run MySQL in a Docker Swarm Cluster. Swarm was introduced in an earlier tutorial, and has the following components: Swarm Cluster master, Swarm manager, and the Node/s in the Swarm Cluster. The Swarm manager may be installed on any node. We shall install it on the same machine as the Swarm master then run Docker client also from the Swarm master node. As in the earlier tutorial, we will create a Docker Swarm cluster of 3 nodes to run MySQL Database Docker image on the cluster.

You will learn ...

Setting the Environment
Creating a Docker Swarm Cluster
Starting the Docker Swarm Manager
Starting Docker Swarm Agents
Listing the Nodes in the Swarm Cluster
Getting Information about the Swarm Cluster
Running MySQL Database Docker Image on the Swarm Cluster
Listing the Docker Containers in the Swarm Cluster
Creating a MySQL Database Table
Listing Logs for the Swarm Cluster

More
sdjournal.org/using-docker-swarm-mysql-database/

OPEN SOURCE BLOG PRESENTATION

Sapere Aude

Vitaly Repin

http://vrepin.org/

Vitaly still enlightens us on the belief that knowledge can help us to improve our life.He uses his blog to spread the knowledge and inspire readers to use the information contained therein. However, his preference to use photo and video sharing services in his blog should not be confused, his primary goal is still to share knowledge and NOT to share his life moments.

Can you tell our readers about yourself and your projects?

I am 37 years old Software Engineer from Finland. My professional career has been connected with FOSS, which begun from my stay at the university. I continued to develop GNU Linux-based solutions in Open Source Software Operations - Nokia division which delivered Linux-based internet tablets and later even Linux-based Nokia phones.

Currently I am an independent IT consultant. I am working on several projects now. Most of my time is spent on the FOSS Governance project for one of the biggest Nordic financial companies. But I also keep my hands dirty with real

programming tasks in the field of big data analytics - ElasticSearch and Bro are my main tools. As well as C/C++, Perl and bash of course. I like cycling and sailing. I try to spend my vacations not on a beach but by exploring the beauty of the nature by MTB or sailing boat.

How you first got involved with blogging?

I have participated in different forums and newsgroups and felt that some of the problem solutions we managed to find there shall not be lost in the forum threads. This is how I came to an idea to start a blog and use a how-to-style context there. Then I started to also add the notes from my personal life on the blog. My blog is in three languages - Russian (my native language), Swedish (the language I aim to master as a second native) and English. Most of the technical content is in English as it is targeted to the international audience. Posts which are specific to Nordic countries are in Swedish and I currently have only one post in Russian with translation of the interesting historical article from Finnish newspaper.

What's the best thing a blogger can give to his readers?

Good posts can save the most valuable resource in our lifetime. Readers can use the experience of the blogger instead of them going through the same painful experiences.

I am trying to have a clear line between technical and non-technical content in my blog. My articles about Logstash will never have passages about my trips to The Alps. And vice versa - my travel notes will never be mixed with Jekyll howtos.

The motto of my blog is "Sapere aude!". This is Horace's phrase which can be translated as "Dare to be wise!" and is strongly associated with the Age of Enlightenment. I share Enlightenment's believe that knowledge can help us to improve our life and I use my blog to spread the knowledge and inspire readers to use it. The primary goal of my blog is NOT to share the moments of my life. For that I prefer to use photo and video sharing services.

Everyone has a favorite/least favorite post. Name yours and why?

I published my favorite post not in the blog but in the Maemo forum during my OSSO times. The title was "Mail for Exchange (MfE). Blame me here, pls". It became a huge success and I got a lot of connections with the users of a Nokia Linux phone because of that post. It was so amazing to feel the connection to the community and get a direct feedback from the real end-users! And it was a great communicational practice as well - sometimes I had to control my emotions talking with angry non-techy users.

Least favourite post, hmm, I can not remember any specific one but I learnt that more emotions I have about the subject, the worse the essay I would create. This is why I try to stay calm while writing. Also, it is really easy when you write about technical things. However, it is not that easy when you address the philosophical agenda of the free and open source movement.

What do you think about Open Source projects?

I am a fellow of FSFE (Free Software Foundation Europe). As you can imagine, I am very positive towards Open Source projects. In my opinion they are beneficial for the participants as this offers a great opportunity to develop both social and engineering skills. Additionally, the society as a whole also benefits from the free and Open Source projects.

I am under impression that your readers are very well aware of the free and open source movement but if they need to explain the FOSS phenomena to the wider audience I can recommend a (gratis) online course "Road to the Free Digital Society" (http://digitalfree.info/) - a project I am proud to be part of.

What is your advice to anyone who wants to advance their programming knowledge?

Study the theory from the books, online courses and (last but not the least) your university lectures and workshops. Practice the knowledge you got there in real life projects. Learn how to collaborate with other engineers through participation in open source projects.

Always keep an eye on new technologies. Be ready to learn new things all the time. Programming is about understanding and understanding requires constant learning. Develop your learning skills.

And probably an unexpected advice from an engineer - study humanities! Don't be focused exclusively on software engineering. Humanities will help you to develop critical thinking, argumentation skills and to understand a "big picture" of the environment you and your solutions operate in.

What is your favorite OS and why?

Of course my favorite OS is Maemo/MeeGo as I had the privilege to be part of the team developing it. However, it has recently discontinued. So, the real answer for today is GNU Linux.

I use Gentoo Linux distribution at home and Debian-based distros for servers if I have a choice. Gentoo Linux makes it easy to play with the sources and different compile-time features. Gentoo community is also very friendly and professional.

Debian GNU/Linux is really easy to setup and maintain - I can focus on the specific tasks which really require my attention.

What is the future of UNIX in general? What do you think?

First let us agree on the common terminology. I understand UNIX as a huge family of OSes. With GNU Linux and FreeBSD included. So, when the purist uses the words "UNIX-like systems" I use the simpler form - UNIX.

I do not see the opportunity for UNIX to disappear in any foreseeable future. I believe it will be alive forever. The real question should be about the quality of its life. Will it be COBOL-like, TeX-like or Web-like? So far I think it goes in the 3rd way. It flourishes because of the constant changes. The changes which are caused by the real need of its users. We all know that UNIX is user friendly, it's just selective about who its friends are. Open Source developers are traditionally great friends of UNIX.

In short - I am very optimistic about UNIX's future. It will be great. "It will be absolutely fantastic", as Mr. Trump would probably say.

Do you have any specific goals for the rest of this year?

My technical goal is to get a better understanding of machine learning systems and big data. Currently, I am playing with different ideas from this field.

I try to keep work-life balance and also invest my time into studying humanities, Swedish language and living social life. I have set specific goals on this front in the beginning of the year and still have time left to achieve them. "The French Revolution" MOOC is saved as a bookmark in my Mozilla Firefox and summer tickets to my beloved Swedish-speaking Åland islands are already in my Mutt mailbox. Looking forward for the summer adventures!

Thank you

How to Incorporate External Utility Scripts Into Logstash Pipeline

Logstash is a great tool to process the logs and extract valuable data from them. There are many useful Logstash filter plugins which make it easy to process the raw log data. However, sometimes external utilities are required to process the data in a more complicated way than existing filter plugins can.
It's possible to code your own filter plugin in Ruby. But what do you do when you already have the filter implemented in some other programming language and want to reuse it in Logstash?
In this case it's easier to communicate with this external filter from Logstash. This article demonstrates the simplest way of incorporating external applications into the Logstash pipeline:

- Logstash launches an external program and delivers the input data to it through command line arguments and stdin.
- External program writes results to stdout in any format understood by Logstash filters (e.g., JSON).
- Logstash parses output of the external program and continues to handle it in the pipeline.

It's needless to say that it is not the very best approach regarding performance. E.g., if startup time of the external application is significant, you may consider to launch this application once (as a daemon/service) and communicate with it using ØMQ or another high-performance message queue.
A detailed explanation and usage example are stated below.

Launching the external program

We will use ruby filter to launch external application and capture its output:

```
filter {
    # <...> <- More filters are above
    # Launching external script to make a deeper analysis
    if [file_path] =~ /.+/ {
       ruby {
        code => 'require "open3"
                 file_path = event.get("file_path")
                 cmd =  "/opt/bin/my_filter.py -f #{file_path}"
                 stdin, stdout, stderr = Open3.popen3(cmd)
                 event.set("process_result", stdout.read)
                 err = stderr.read
                 if err.to_s.empty?
                   filter_matched(event)
                 else
                   event.set("ext_script_err_msg", err)
                 end'
           remove_field => ["file_path"]
       }
    }
    # Parsing of the process_result is here (see the next section)
 }
```

Note:

- External application /opt/bin/my_filter.py is launched only if file_path field is not empty. This field shall be extracted earlier in the filter pipeline. It's value (#{file_path}) is used in the command line to launch external filter.
- stdin handle is accessible for our tiny ruby script and it can be used to send more data to the external program (/opt/bin/my_filter.py).
- If application stderr is not empty, filter is not considered to be successful and stderr content is recorded in the ext_script_err_msg field.
- If processing was successful, the output of the external program is recorded in the process_result field and the file_path field is removed
- This config has been tested with logstash 5.3.0.

Parsing output of the external program (JSON)

The easiest way to deliver the data back to Logstash is to use one of the structured data formats understood by Logstash filters: JSON, XML or the more old-fashioned key-value (kv).
Example with JSON:

```
if [process_result] =~ /.+/ {
      json {
         source => "process_result"
         remove_field => [ "process_result" ]
       }
   }
```

Note:

- Field process_result holds the output of the external application and is supposed to be in JSON format.
- If parsing was successful JSON fields are becoming event fields and intermediate field process_result is removed.

Several words about exec output plugin

If you only need to launch external utility upon any matched Logstash event, you may consider to use simpler approach – exec output plugin

Interview with Daniel Cialdella Converti

Can you tell our readers about yourself and your role nowadays?

During the last four and half years, I have been working as DBA for a big company based in Switzerland and UK, on different engines. I have been servicing countries in E.U. on two principal paths, maintaining what we have and ensuring the company adopts the latest technologies and products.

How you first got involved in programming?

I started with a Texas Instruments TI-99/4A playing games, typing them first and playing them later. Thereafter, I developed a keen interest in Dbase II, Clipper, Fox and Ansi SQL, until now. In different projects or applications, Arduino, NodeJs, Bash, MongoDB etc. were some of the programming tools I used. For the operating systems, I started with UNIX, then D.O.S., Mac OS, Windows, and later revert back to Linux/Unix. Currently, four databases engines and two operating systems comprise my daily programming fields. Store procedures, triggers, views, queries (in some different flavors of SQL language) are my everyday tasks, and this is "programming" too.

While having a wide field of expertise, you put noticeably more emphasis on MySQL. Why?

MySQL/MariaDB is my principal engine, and it's where we are recording some tremendous improvements. For the last four and half years, it has proved to be the most important engine and there is still a long way to go. It's not yet finished, we are just beginning.

You are contributing to the Open-Source project. Could you tell our readers which one, and what is your role?

I was part of Linux Groups where I got the opportunity to write for Magazines and work as a trainer in Linux/MySQL/Apache and other products. I not only taught Linux in institutes based in Madrid but also was among the beta testers for some applications, and was the leader/coordinator in some of them. All related to APP using GPL software only. Sometimes soft, other times a combination of hard and soft.

What is your the most interesting IT issues, and why?

As an I.T. person at work, I'm always resolving issues every day which are related to people's projects. This sounds obvious but recover/restore/mistakes are the big ones. I think 70% of my time I'm a fireman and 30% an architect.

What tools do you use most often, and why?

I use Linux and Bash databases tools programmed within and from other companies. 80% are handmade and customized for our needs. Also, I have to maintain/admin Win+SQL Srv. These add up to my daily tools too.

What was the most difficult and challenging implementation you've done so far? Could you share some details with us?

I think the greatest task so far was to build an architect/environment for an app in the internet for 1 million end users, which supports both online and batch processing. At that risky age, building that App was an uphill task. For the functionality of the App, the project demanded some servers, lots of components to work together, securing very sensitive data of payment transactions and customers' details, all in one. It would be the central app for a big company, and replace the (very) old one which had been in existence. I had to think, define, develop, implement, test, optimize, and document the project. Those were some three strenuous years.

What future do you see for Open-Source systems?

Open-Source systems are "the logical future" in the I.T. world. The central servers, the core of the internet, the processes running in the backgrounds will have to be GPL. I hope for this, and push for it to come to a realization.

Anyway, for now, the end-users will have to use easy and graphical tools since they are not technical experts, and sometimes can't understand the relevant facts about what they are using.

O.S. / GPL is not only a nice term to add to our tool but also it's a community, a dream, a wish, and it sometimes prepares to add more professionals on it. 90% of Startups use O.S. software and Linux due to cost implications. That's a good sign.

I started teaching in 1999 and became a part-time consultant for different projects. Additionally, for the last four and a half years, I served as a Linux teacher trying to teach the magic of freedom to my students.

During the past 15 years, we have time and again heard about the "big brother", "the black hands", "the govern monitoring", and during the last 7 years, we verified that it was true from the leaks. The cyber weapons of NSA are the last proof that we don't have to trust commercial companies or governments because their interest is not in the promotion of a good APP. One of them is into selling and the other in hacking them all.

In my mind, I think commercial products will exist which will be managed by very big companies trying to maintain their kingdom, their profits, their big engines and working for themselves. For the few, I think it will yield them an incredible amount of profit.

And a big community in the world will rule the O.S. arena by working for "all"; trying to generate a better tool, app or service, living for coding and earning enough to have a good life. Sometimes wanting a good code or service instead of being only a millionaire(code first, earn money later).

Do you have any specific goals for the rest of this year?

Yes, the official ones are to complete a migration of 100 servers to a new product (off course GPL), and collaborate in the reinstallation of 200+ Linux servers.

The unofficial ones are finishing the BOT for Trading (new version), certify in MariaDB and help communities in the implementation of technology.

Do you have any untold thoughts that you want to share with the readers?

I'm not sure about "telling them". If I do, they would not be untold anymore.

I did things anonymously to avoid the relation between what I have to think working for a big company and what I want to do as a human. I write what I think with an alias so as not to be associated with my professional life or work. Sometimes companies and communities are not on the same side.

What’s the best advice you can give to programmers?

As a programmer, try to be part of every project you come across. Be collaborative, be a nice person, enjoy it, learn and share what you think, and be an open-minded I.T. individual. Finally, try to find the best solution and not the trendier one. If you take heed of three advices, at least, you will be a great person.

Thank you

ServerU

Rack-mount networking server

Designed for BSD and Linux Systems

DESIGNED FOR FreeBSD | DESIGNED FOR GNU / Linux | DESIGNED FOR PRO apps FreeBSD Enterprise Appliance | DESIGNED FOR pf Sense

Designed. Certified. Supported

Up to **5.5Gbit/s** routing power!

KEY FEATURES

- 6 NICs w/ Intel igb(4) driver w/ bypass
- Hand-picked server chipsets
- Netmap Ready (FreeBSD & pfSense)
- Up to 14 Gigabit expansion ports
- Up to 4x10GbE SFP+ expansion

PERFECT FOR

- BGP & OSPF routing
- Firewall & UTM Security Appliances
- Intrusion Detection & WAF
- CDN & Web Cache / Proxy
- E-mail Server & SMTP Filtering

contactus@serveru.us | www.serveru.us
8001 NW 64th St. Miami, LF 33166 | +1 (305) 421-9956

COLUMN

The latest worldwide WannaCry malware attack crippled the British National Health Service for hours, delaying non-essential operations and shutting down accident and emergency departments. Now that the inevitable finger pointing exercise has begun, who should be held responsible?

by Rob Somerville

So far, we have been very fortunate with the latest malware attack in that nobody has died or suffered major injury or harm. At least that is the case at the time of writing, but only time will tell. It was a close call though, with critical devices such as automated blood banks and MRI scanners being out of action, and doctors having to resort to pencil and paper and not being able to access patient records. So a genuine “Well done and Thank you” is in order to all the technical, administrative and medical staff who went the extra mile in dealing with this incident efficiently. As to the perpetrators of this wicked deed, my only hope is that someday karma will catch up with them in the form of a dearth of anesthesia, a power outage, and the only implement available to perform life-saving surgery by removing your gangrenous internal organs being a rusty, blunt screwdriver. Of course, with a loose handle.

Maybe that is a bit harsh. It is still unknown if this was a deliberate attack by a state actor (North Korea is currently in the frame for this), or just pure unintended consequences by a script kiddie, or something in-between. The broad scope of the attack, affecting more than 230,000 computers in over 150 countries will make unwinding the Gordian knot of how exactly WannaCry entered the NHS network - complex. What is clear though is the toxic mix of legacy kit, commercial greed, underfunding and cognitive dissonance played a major part.

Government and medical systems are complex beasts. Much has been mentioned about systems running Microsoft XP, and to date, it is not clear what percentage (if any) of these systems were standard desktop systems. XP is still supported from a security standpoint both as a desktop and an embedded system, albeit at an outrageous cost to the licensee. It has been suggested that Microsoft is justified in deliberately ramping up the cost to force the market to move on to later, more secure versions of the Operating System. This might be acceptable where the scenario is not mission critical and funds are available, but where lives depend on the device and the cost and time of re-engineering the back-end device is counted in years (MRI scanners are complex and fickle beasts) , this is not always feasible. MS seems to have learned their lesson from the debacle of almost 20 years ago, where their reputation took a hammering from the failure of NT on Aegis missile cruiser USS Yorktown that caused the propulsion systems on the ship to fail. At least they released a free of charge patch to this malware fairly quickly. Maybe the sight of horses running out through the stable door prompted their change of heart. This brings me to the whole issue of patch management. As an ex-government employee, I can tell you it is an absolute nightmare. A lot of government IT departments run in fire-fighting mode, the lack of resources, documentation and the sheer complexity of the organisation (never mind the systems or the politics) is probably the birthplace of the phrase “It is easier to seek forgiveness than approval”. Patch a system up with the latest standards and no-one ever cares. Break a system in the process, and there will be mud-slinging, investigations and censure. Trying to get an approval, or at least a degree of support from senior management is well-nigh impossible. Your request will be pushed into the long grass, ignored, or you will be classed as a troublemaker for your efforts. Patching in such an environment is not just the case of running a few commands on a terminal. Due to the level of integration between the systems, the chances of breaking something further downstream is quite high, especially if these systems have not been upgraded for a while. So while it is easy to blame system administrators for not keeping systems totally up to date, sometimes it is well-nigh impossible. Often the adage of “If it ‘aint’ broke, don’t fix it” and a pair of crossed fingers is your only hope.

Commercial greed by vendors monetising “security as a service” really is the pits. Software is either fit for purpose or it is not. If bugs are found, these should be squashed and the upgrades released free of charge. If need be, the cost be built

into the initial product over a fixed lifetime cycle. The industry needs to focus less on new features and the competitiveness of the marketplace and more on inherent stability, security and the quality of code. Like Typhoid Mary, your code could be carrying something nasty. The IT professional realises that there are “Unknown unknowns” and does his or her best to engineer these out. However, he or she realizes that nothing is perfect and there is always an element of risk. Unfortunately, due to the accountants, lawyers and marketers, this has become a casino of opportunity.

Underfunding has become the watchword as IT departments are looked upon as a cost. Like the tread on a tyre, so often the excuse is we don’t have the budget for a new set, but just wait until Mr. Traffic Policeman catches you with a defective one. A small fine. Go on a wet road, brake hard, and you might not get out of the scenario alive. An ounce of prevention costs less than a ton of cure, but sadly many organizations are too short-sighted to see this. IT security is not cheap. It requires not only long term but a continual investment. Like a toner cartridge, it is a consumable. What stopped the criminals six months ago is a lot less effective – or indeed ineffective - today. Resting on your laurels is a recipe for failure.

Cognitive dissonance (That great big disconnect between the IT folk in the basement and the executives in the stratosphere) is at the root of the issue. Blinded by the men in shiny suits, the marketing blurb, their detachment from grepping an Apache access log in real time, IT security is just another pain. Like the IT department, technology is not a playing field. It is more like an ocean, turbulent one minute; calm the next, and very deep. The risks are different from what occurs on land, and you need a good crew if you are not going to run aground. Often, management prefers to sail under a flag of convenience (outsourcing) rather than have their crew. It allows for plausible deniability when really the best solution is engagement and understanding. Until senior management engages with those that inhabit the dark spaces below decks, understand the issues and act upon them, poor communication, mistrust, and frustration will continue on both sides.

In the final analysis, cognitive dissonance is the biggest Achilles heel so far. Until organizations take IT and security matters seriously, there will be a time that someone dies. Be it in the medical, nuclear or automotive sectors. It is not a question of “if” - it is just a matter of when.

MAGAZINE BSD

Editor in Chief:
Ewa Dudzic
ewa@bsdmag.org
www.bsdmag.org

Contributing:
Natalia Portillo, E.G Nadhan, Daniel Cialdella Converti, Vitaly Repin, Henrik Nyh, Renan Dias, Rob Somerville, Hubert Feyrer, Kalin Staykov, Manuel Daza, Abdorrahman Homaei, Amit Chugh, Mohamed Farag, Bob Cromwell, David Rodriguez, Carlos Antonio Neira Bustos, Antonio Francesco Gentile, Randy Remirez, Vishal Lambe, Mikhail Zakharov, Pedro Giffuni, David Carlier, Albert Hui, Marcus Shmitt, Aryeh Friedman

Top Betatesters & Proofreaders:
Daniel Cialdella Converti, Eric De La Cruz Lugo, Radjis Mahangoe, Daniel LaFlamme, Steven Wierckx, Denise Ebery, Eric Geissinger, Luca Ferrari, Imad Soltani, Olaoluwa Omokanwaye, Radjis Mahangoe, Katherine Dizon and Mark VonFange.

Special Thanks:
Denise Ebery
Annie Zhang
Katherine Dizon

Senior Consultant/Publisher:
Paweł Marciniak

Publisher:
Hakin9 Media SK,
02-676 Warsaw, Poland Postepu 17D Poland
worldwide publishing editors@bsdmag.org

Hakin9 Media SK is looking for partners from all over the world. If you are interested in cooperation with us, please contact us via e-mail: editors@bsdmag.org

IS AFFORDABLE FLASH STORAGE OUT OF REACH?

NOT ANYMORE!

IXSYSTEMS DELIVERS A FLASH ARRAY FOR UNDER $10,000

Introducing FreeNAS® Certified Flash. A high performance all-flash array at the cost of spinning disk.

KEY ADVANTAGES

- 10TB of all-flash storage for less than $10,000
- Unifies SAN/NAS for block and file workloads
- Runs FreeNAS, the world's #1 software-defined storage solution
- OpenZFS ensures data integrity
- Scales to 100TB in 2U
- Perfectly suited for Virtualization, Databases, Analytics, HPC, and M&E
- Performance-oriented design provides maximum throughput/IOPs and lowest latency
- Maximizes ROI via high-density SSD technology and inline data reduction

The all-flash datacenter is now within reach. Deploy a FreeNAS Certified Flash array today from iXsystems and take advantage of all the benefits flash delivers.

For more information, visit iXsystems.com/FreeNAS-Certified-Servers today.

iXsystems™

Developing Java EE Applications on Cloud

What you will learn...

- How to use RAD to create Java EE applications.
- Connect RAD to a PureApplication.
- Create a Cloud application in RAD.
- Publish the cloud application onto PureApplication.
- Use the Virtual Application Builder in PureApplication to build the Virtual Application Pattern topology.
- Deploy the Virtual Application Pattern from RAD to the private cloud.

What you should know...

- Database and JPA concepts.
- Basic Java EE knowledge.
- Basic concepts of cloud computing.