MySQL security is more important today than ever. In this Percona eBook, we'll examine MySQL 5.6 security vs ease of use, why it's important to review your current database security measures along with real-world best practices and advice.
# Table of Contents

Chapter 1: MySQL 5.6 security vs ease of use 3
Chapter 2: Database security: How to remove the uncertainty 5
Chapter 3: MySQL Security: Armoring Your Dolphin 8

---

## About Percona

Percona was founded in August 2006 and now employs a global network of experts with a staff of more than 120 people. Our customer list is large and diverse, including Fortune 50 corporations, popular websites, and small startups. We have over 2,000 customers and, although we do not reveal all of their names, chances are we're working with nearly every large MySQL user you've heard about. To put Percona's MySQL expertise to work for you, please contact us.

---

### Contact Us 24 Hours A Day

Is this an emergency? Get immediate assistance from Percona Support 24/7. [Click here](#)

**Skype:** oncall.percona  
**GTalk:** oncall@percona.com  
**AIM (AOL Instant Messenger):** oncallpercona  
**Telephone direct-to-engineer:** +1-877-862-4316 or +44-800-088-5561  
**Telephone to live operator:** +1-888-488-8556  
**Customer portal:** [https://customers.percona.com/](https://customers.percona.com/)

<table>
<thead>
<tr>
<th>Sales North</th>
<th>(888) 316-9775 or 3169775</th>
<th>America</th>
<th>(208) 473-2904</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sales</td>
<td>+44-208-133-0309 (UK)</td>
<td>Europe</td>
<td>0-800-051-8984 (UK)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0-800-181-0665 (GER)</td>
</tr>
<tr>
<td>Training</td>
<td>(855) 55TRAIN or 55TRAIN</td>
<td></td>
<td>(925) 271-5054</td>
</tr>
</tbody>
</table>

---

Copyright © 2006-2014 Percona LLC
MySQL 5.6 security vs ease of use

By Peter Zaitsev

MySQL 5.6 surely changes the game when it comes to security vs. ease of use. Before MySQL 5.6 we would get default MySQL installations that were insecure – the user “root” being created without a password as well as anonymous users with limited access from the local host (though still enough to cause DOS attacks or crash the MySQL server.

There were some exceptions to this rule – such as Debian/Ubuntu install scripts would interactively suggest you to set password for root user if it was not set. Still most users would get a MySQL install with a root account and no password.

This is not the case with MySQL 5.6 when you’re doing fresh MySQL installs! Installing official RPM on CentOS6 I’m getting this:

```
A RANDOM PASSWORD HAS BEEN SET FOR THE MySQL root USER!
You will find that password in '/root/.mysql_secret'.

You must change that password on your first connect,
no other statement but 'SET PASSWORD' will be accepted.
See the manual for the semantics of the 'password expired' flag.

Also, the account for the anonymous user has been removed.

In addition, you can run:
/usr/bin/mysql_secure_installation

which will also give you the option of removing the test database.
This is strongly recommended for production servers.
```

So we’re getting a random password for the root account by default instead of an empty one. Furthermore it is not stored in the root directory my.cnf but separate .mysql_secret file so you need to enter it explicitly to connect to the server for the first time – and it is for good reason, as this is only a temporary password. You can’t really use the MySQL server until you change it:

See next page
As such MySQL will refuse any statements, even ones that do not cause any database content access until you change password with the `SET PASSWORD` command.

If you're looking to keep password you can run:

```sql
mysql> set password=password('MySecurePassword');
Query OK, 0 rows affected (0.00 sec)
```

You also have an option to go back to the old behavior and remove the password for an account *(this is what I do on MySQL running on VirtualBox on my laptop as I keep it for testing only)*

```sql
mysql> set password='';
Query OK, 0 rows affected (0.00 sec)
```

So at least with RPM installs, MySQL 5.6 is getting more secure, but adding a little more effort after installation is worthwhile. I hope this change will make things more secure and will not discourage a lot of users by complicating the install process.
Ah database security… the black sheep of topics and something you would really rather not have to deal with right?

I mean surely all the fanfare and paranoia is reserved for the neck beards with tinfoil hats whom live in their own D.I.Y Faraday cage … that must be it … it just has to be?

No, the hard reality is the world is not rose tinted and “they” are out to get you be it for fun or for profit; from defacements to theft compromising your applications, and more importantly your data is big business. For some these acts are nothing short of sheer entertainment for an otherwise boring evening.

Note I’m avoiding discussing corporate espionage/government spying, as this re-enforces the image of it all being cloak and dagger; admittedly some of it is … the part that’s going to affect you, your business, your livelihood isn’t.

It’s time for that wake-up caffeine infusion and drink the “kool aid” – this is not something you can shun and pretend it’s a monster under the bed/in the closet that doesn’t really exist. Unlike the “bogey man” these threats are real, and have real impacts to peoples livelihood.

F.U.D? The fear part I certainly am wanting to portray here; a level of Fear is healthy; it keeps you alert, the uncertainty and doubt? No these should and will be removed so please allow me to continue.

Removing Uncertainty
As with anything that requires research I’m sure you the reader would carry out proper “Due Diligence.” You wouldn’t want anything to adversely affect your product/business, taking hosting as an example you’ll look at the providers reputation their S.L.A. agreements etc.

What about their Security credentials? PCI / SOX / HIPAA… there’s numerous classifications.

“But I don’t need PCI / SOX / HIPAA!” to this I say you need a compromise of your application/business even less… what’s the harm in asking the provider if they have been through any regulatory compliance? I just don’t get the stigma some people seem to feel when asking a question related to security. Remember when deploying your application you’re building upon your hosting providers infrastructure.

“A foolish man who built his house on sand” - in short if your foundations are not sound you’re opening yourself up to failure and compromise.
Dispelling Doubt
“By failing to prepare you are preparing to fail” – Benjamin Franklin

Let’s look at this a little more with some common misconceptions.

“I don’t really need to look at security, my project business is small and will not be attacked.” I’d liken this statement to saying you don’t need seat belts and air bags because you’re a careful driver; driving at night with no lights on because “I have good night vision.” You have safety and security measures in your everyday life which because they fall as part of the routine are not thought about: locks on doors, car/home/business alarms, cctv, gps locators for phones/cars/tablets/laptops … we need to eliminate this thinking that information security is anything other than a requirement which should form part of our every day “norms”.

“Security is too expensive.” Have you looked at the cost of a compromise of your system, how much is the potential loss of your entire customer base worth to you? …not looking quiet so expensive now is it? Liken an investment in security to an investment in High Availability: you want your application to be “always on” … why do many think “secure” is prohibitively expensive to achieve?

“We simply don’t have the resources to implement security measures.” Yet you have the resources for development, DBA’s, sysadmins? One of the best ways to introduce security into your application is “from the ground up,” so that it becomes part of your general practise – this requires a “state of mind” orientated toward security.

What many fail to realize is assuming your business is successful you already have a state of mind orientated to best practises which work for your business to produce an application/service of value; minor tweaking to this could also introduce a mindset of security.

Remediation - “the action of remedying something, in particular of reversing or stopping environmental damage.”

It’s not going to be a painful or expensive as you may think; the first most powerful step is a minor change of development/sysadmin attitudes to consider security implications of code/services/configurations, let’s bring back the healthy attitude of asking questions...

Do I really need to disable SELinux to get this to work? – the answer is of course no you shouldn’t, this should be the same vein of thought of do I really need to chmod this 777 to make it work?

Does this service really need to be installed? – e.g. bluetoothd doesn’t need to be on your production machines.

We’re adding a user input form, we should really sanitize the input – seems obvious to most now, though this was met with just as much “resistance to change” before it became a best practice standard.
Does MySQL really need to be accessible from everywhere on the internet? – again may seem obvious to most now, though this was and sometimes still met with resistance of “I may need to run queries from home, off my mobile, from the open wifi at the local coffee shop …” (those of a security orientated nature I apologize for this statement and I can sense the cringing now in progress as this is read …).

The above is just a small example of reducing your attack surface. Your attack surface being the potential entry points into your system/network/application which can be potentially attacked.

The thinking behind the need to invest a lot of money into some expensive appliance / magic solution is for the most part misguided and throwing a “security blanket” over or in front of your application isn’t going to be as effective as ensuring consideration of security at every layer of your business.
Q: Do you have a favorite software firewall you recommend that I can run on an EC2 instance in front of my MySQL server?
A: I’d probably just do this with iptables. Any of the other Linux-based software firewall packages are all going to be wrappers around iptables anyway. However, if your MySQL server is already in EC2, you’re going to be better served by Amazon security groups as your front-line firewall. You can also run inside a VPC for some additional isolation.

Q: What do you use for disk encryption in the cloud environment?
A: I might use GPG if I need to encrypt specific files, or just encrypt data in the database via the MySQL functions or in my application, and there’s also no reason why you can’t use dm-crypt/LUKS in the cloud if you’re willing to accept that mounting the device will require some manual intervention, but for the most part, I think disk encryption for servers is pretty useless, because once you enter the key to unlock and mount the volume, anyone that can get access to that machine has full access to the data. I’ve heard of Gazzang being a possible solution here, but I’ve not personally looked at it, so I can’t speak to its performance or suitability.

Q: How much overhead does SSL put on a heavily-loaded MySQL box?
A: That depends on a lot of factors, such as the cipher in use and the kinds of load you’re running. The slowest part of SSL is the connection setup, so you’re going to see a lot less overhead for replication, connection pooling, or long-running transactions than you would from an application with rapidly connects to the database and then disconnects. There’s a graph on the yaSSL site which shows roughly a 25% penalty at 32 threads, but their benchmarks are from 2011 and they were run on someone’s laptop, so I have my doubts as to how well that translates to modern server-grade hardware.

Q: Can MySQL do LDAP/Kerberos login?
A: You can use the PAM authentication plugin (either the commercial one from Oracle or Percona’s open source version) to authenticate against an LDAP database.
About the authors

Peter Zaitsev, Percona's CEO and founder, is arguably the world's foremost expert in MySQL performance and scaling, with a special expertise in hardware and database internals. Peter's work has contributed to dozens of MySQL appliances, storage engines, replication systems, and other technologies. Peter co-authored High Performance MySQL along with two other Percona experts. He is a frequently invited guest at open source conferences, and has been a sell-out speaker at the yearly MySQL User Conference since its inception. Peter currently lives in North Carolina with his wife and their two children.

David Busby is on Percona's Remote DBA team (RDBA) and is the RDBA Security Lead. Prior to his role at Percona, David spent 14 years as a Linux systems admin - generally in different roles: development, network admin, support, DBA and more. A Fedora user and a fan of puppet, he contributes to the EPEL packages for Openstack. David also has an interest in infosec, and so is "generally paranoid about security." He is also familiar with metasploit, sqlmap, john, oclHashCat, and has also written a few Python tools.

Ernest Souhrada joined Percona's consulting team in April 2012, where he is currently a senior architect, bringing many years of diverse experience as a generalist systems architect and engineer. He has held almost every technology role present in the Internet era, from Perl/Java developer to Linux sysadmin, MySQL DBA to Cisco network engineer, security auditor to IT engineering manager. He thrives on and excels at taking on those novel challenges which require creative cross-disciplinary solutions.

Visit Percona's ever-growing MySQL eBook library for more great content at http://www.percona.com/resources/mysql-ebooks