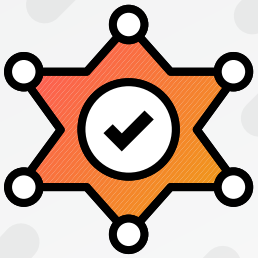

The DevOps Security Checklist

A PROJECT BY sqreen

Culture

SERIES BAZ

❑ Cover your ass

It is not a question of “if” but “when”. Evaluate your risks, prepare a proper action plan in case of a breach and communicate properly after the fact.

https://blog.serverdensity.com/how-to-write-a-postmortem/
https://codeascraft.com/2012/05/22/blameless-postmortems/
https://blog.sqreen.io/cybersecurity-risk-assessment-for-startup-cto/

POST SERIES B

❑ Follow an onboarding / offboarding checklist

This checklist should contain a list of all the steps you need to enforce when an employee, contractor, intern, etc... joins your company. A similar list can also be used when the someone is leaving your team.

https://www.rippling.com/
https://about.gitlab.com/handbook/general-onboarding/
https://about.gitlab.com/handbook/offboarding/

SERIES B

❑ Gamify security and train employees on a regular basis

Humans are the weakest links in the security chain. DevOps contribute to the security awareness of all the employees in a company. By explaining how an attacker could infiltrate your company, you will increase the awareness and thus minimize the chance of a hack. Don’t forget fishing and spear-fishing attacks.

https://www.schneier.com/blog/archives/2008/03/the_security_mi_1.html
http://www.govtech.com/blogs/lohrmann-on-cybersecurity/Ten-Recommendations-for-Security-Awareness-Programs.html
http://lifehacker.com/5933296/how-can-i-protect-against-hackers-who-use-sneaky-social-engineering-techniques-to-get-into-my-accounts

SERIES A

❑ Stay on top of best practices

DevOps is an ever-changing landscape. Ensure that you stay up to date in terms of new technologies, vulnerabilities or best practices.

https://aws.amazon.com/whitepapers/architecting-for-the-aws-cloud-best-practices/
https://docs.microsoft.com/en-us/azure/security/azure-security-network-security-best-practices
http://webopsweekly.com/

SERIES A

❑ Understand the risk

The cost of breaches is drastically increasing and security should be taken seriously inside an organization. DevOps engineers should play an important role in advocating for better security practices.

http://www.nttcomsecurity.com/us/uploads/documentdatabase/US_Report_Risk_Value_Public_Approved_v2.pdf
http://fortune.com/2016/06/15/data-breach-cost-study-ibm/
https://www.troyhunt.com/the-emergence-of-historical-mega-breaches/

Code

POST SERIES B

❑ Don't implement your own crypto

The problem with cryptography is, that you don’t know you are wrong until you are hacked. So don’t do your own crypto. Use standards instead.

https://en.wikipedia.org/wiki/Bcrypt
http://crypto.stackexchange.com/questions/43272/why-is-writing-your-own-encryption-discouraged
https://download.libsodium.org/doc/
https://blogs.dropbox.com/tech/2016/09/how-dropbox-securely-stores-your-passwords/

SERIES A

❑ Ensure you are using security headers

Modern browsers support a set of headers dedicated to block certain types of attacks. Make sure you properly implemented all security headers. Don't forget about the CSP.

https://securityheaders.io/
https://myheaders.sqreen.io/
https://blog.appcanary.com/2017/http-security-headers.html

POST SERIES B

❑ Go hack yourself

If your company doesn't have yet a structured security team, help create a multidisciplinary Red Team to stress your application and infrastructure. Providing an easy environment for the Red Team to attack the application should be part of the scope of DevOps.

http://www.devsecops.org/blog/2015/12/10/red-team-pwning-the-hearts-and-minds-one-ticket-at-a-time

SERIES B

❑ Integrate security scanners in your CI pipeline

Integrate a Dynamic Application Security Testing (DAST) tool in your CI, but just like SAST be aware of the high number of false positives.

http://www.arachni-scanner.com/
https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project
https://www.acunetix.com/vulnerability-scanner/

SERIES A

❑ Keep your dependencies up to date

Third-party libraries can put your application at risk. Make sure you track your vulnerable packages and update them regularly.

https://www.sqreen.io/

https://pyup.io/
https://snyk.io/
https://nodesource.com/products/certified-modules

POST SERIES B

❑ Protect your CI/CD tools like your product

Your continuous deployment pipeline is the backbone of your IT. Security should be checked at each step. Your CI builds should fail if you detect a security vulnerability. Store your CI configuration for traceability and audit.

https://wiki.jenkins-ci.org/display/JENKINS/JobConfigHistory+Plugin
https://www.slideshare.net/kponiatowski/if-cicd-teams-have-time-for-security-so-do-you

SERIES A

❑ Run Security tests on your code

Static Application Security Testing (SAST) is an easy and fast way to find security vulnerabilities in your code. You can enforce SAST security checks in your CI, but be aware of the high number of false positives that can frustrate developers.

https://www.owasp.org/index.php/Source_Code_Analysis_Tools
https://github.com/mre/awesome-static-analysis
https://docs.travis-ci.com/user/coverity-scan

Infrastructure

SERIES B

❑ Automatically configure & update your servers

An automated configuration management tool helps you ensure that your servers are updated and secured.

Chef: https://learn.chef.io/tutorials/
Puppet: https://www.digitalocean.com/community/tutorials/how-to-install-puppet-4-in-a-master-agent-setup-on-ubuntu-14-04

Ansible: http://docs.ansible.com/ansible/intro_getting_started.html
Salt: https://docs.saltstack.com/en/latest/topics/tutorials/walkthrough.html

---

SERIES A

❑ Backup regularly

Your data is likely to be your business's most precious asset. Be sure not to lose it. Implement proper backups and check for backup integrity.

MongoDB Backup: https://docs.mongodb.com/manual/core/backups/
Postgresql: https://www.postgresql.org/docs/current/static/backup.html
Linux: http://www.tecmint.com/linux-system-backup-tools/
https://www.dataone.org/best-practices/ensure-integrity-and-accessibility-when-making-backups-data

---

SERIES A

❑ Check your SSL / TLS configurations

Use free tools to scan your infrastructure regularly and make sure the SSL configurations are correct.

https://observatory.mozilla.org/
https://www.ssllabs.com/
https://diogomonica.com/2015/12/29/from-double-f-to-double-a/

SERIES A

❑ Control access on your cloud providers

The best way to protect your services (database, file storage) is to not use passwords at all. Use the built-in Identity and Access Management (IAM) functions to securely control access to your resources.

http://docs.aws.amazon.com/IAM/latest/UserGuide/introduction.html
https://cloud.google.com/compute/docs/access/create-enable-service-accounts-for-instances

POST SERIES B

❑ Encrypt all the things

SSL performance problems are a myth and you don't have any good reasons not to use SSL on all your public services.

https://letsencrypt.org/
https://certbot.eff.org/
https://www.digitalocean.com/community/tutorials/how-to-secure-nginx-with-let-s-encrypt-on-ubuntu-14-04
https://www.digitalocean.com/community/tutorials/how-to-secure-apache-with-let-s-encrypt-on-ubuntu-14-04

POST SERIES B

❑ Harden SSH configurations

SSH is the defacto remote login mechanism on Linux environments. It's also the de facto penetration vector for hackers. Make sure you have proper SSH configurations.

https://devops.profitbricks.com/tutorials/secure-the-ssh-server-on-ubuntu/
https://www.digitalocean.com/community/tutorials/how-to-set-up-ssh-keys–2

SERIES B

❑ Keep your containers protected

Use Docker (or Kubernetes), and ensure that they are patched and secure. Use tools to automatically update and scan your containers for security vulnerabilities.

https://www.docker.com/docker-security
https://docs.docker.com/docker-cloud/builds/image-scan/
https://jpetazzo.github.io/2015/05/27/docker-images-vulnerabilities/
https://www.slideshare.net/MichaelCherny/security-best-practices-for-kubernetes-deployment

SERIES A

❑ Log all the things

Infrastructure logs and application logs are your most precious allies for investigating a data breach. Make sure your logs are stored somewhere safe and central. Also make sure you whitelist- or blacklist-specific incoming data to avoid storing personally identifiable information (PII) data.

https://qbox.io/blog/welcome-to-the-elk-stack-elasticsearch-logstash-kibana
https://www.loggly.com/

POST SERIES B

❑ Manage secrets with dedicated tools and vaults

When you need to store cryptographic secrets (other than database password, TLS certificate, ...) and perform encryption with them, you should use dedicated tools. This way the cryptographic secret never leaves the tool and you get auditing features.

https://www.vaultproject.io/
https://github.com/square/keywhiz
https://aws.amazon.com/cloudhsm/
https://aws.amazon.com/kms/

SERIES B

❑ Store encrypted passwords in your configuration management

Storing passwords (like databases ones) can be done on a dedicated database with restricted access. An other solution is to store them encrypted in your Source Code Management (SCM) system. That way, you just need the master key to decrypt them.

Chef: https://github.com/chef/chef-vault
Puppet: https://puppet.com/blog/encrypt-your-data-using-hiera-eyaml
Salt: https://docs.saltstack.com/en/latest/ref/renderers/all/salt.renderers.gpg.html
Ansible: http://docs.ansible.com/ansible/playbooks_vault.html

SERIES A

❑ Upgrade your servers regularly

Server packages and libraries are often updated when security vulnerabilities are found. You should update them as soon as a security vulnerability is found.

https://www.ubuntu.com/usn/
https://help.ubuntu.com/community/AutomaticSecurityUpdates
https://access.redhat.com/security/vulnerabilities

POST SERIES B

❑ Use an immutable infrastructure

Use immutable infrastructures to avoid having to manage and update your servers.

https://martinfowler.com/bliki/ImmutableServer.html
https://hackernoon.com/configuration-management-is-an-antipattern-e677e34be64c#.n68b1i3eo

Protection

SERIES A

❑ Don't store credit card information (if you don't need to)

Use third-party services to store credit card information to avoid having to manage and protect them.

https://stripe.com/
https://www.braintreepayments.com
https://www.pcisecuritystandards.org/pdfs/pciscc_ten_common_myths.pdf
https://medium.com/@folsen/accepting-payments-is-getting-harder-1b2f342e4ea#.897akko4q

POST SERIES B

❑ Enforce Two-factor authentication (2FA)

Enforce 2FA on all the services used (whenever possible).

https://duo.com/
https://auth0.com/
https://nakedsecurity.sophos.com/2016/08/18/nists-new-password-rules-what-you-need-to-know/

POST SERIES B

❑ Ensure Compliance with Relevant Industry Standards

Comply to standards to ensure you follow industry best practices and answer your customer needs. But simple compliance will never protect your apps.

https://cloudsecurityalliance.org/
https://en.wikipedia.org/wiki/ISO/IEC_27001:2013
https://en.wikipedia.org/wiki/Payment_Card_Industry_Data_Security_Standard

SERIES B

❑ Have a public bug bounty program

A bug bounty program will allow external hackers to report vulnerabilities. Most of the bug bounties program set rewards in place. You need security aware people inside your development teams to evaluate any reports you receive.

https://www.tripwire.com/state-of-security/vulnerability-management/launching-an-efficient-and-cost-effective-bug-bounty-program/
https://www.hackerone.com/
https://bountyfactory.io/en/index.html

SERIES A

❑ Have a public security policy

This is a page on your corporate website describing how you plan to respond to external security reports. You should advise you support responsible disclosure. Keep in mind that most of the reports that you receive probably won't be relevant.

https://www.intercom.com/security
https://www.zendesk.com/product/zendesk-security/
https://www.apple.com/support/security/

SERIES B

❑ Protect against Denial Of Service (DoS)

DoS attacks are meant to break your application and make it unavailable to your customers. Use a specific service to protect your app against Distributed Denial Of Service attacks.

https://www.akamai.com/
https://www.cloudflare.com/ddos/
https://www.ovh.com/us/news/articles/a1171.protection-anti-ddos-service-standard

SERIES A

❑ Protect your applications against breaches

Detect and block attacks in real-time using a protection solution. All the OWASP top-10 vulnerabilities (SQL injections, NoSQL injections, cross-site scripting attacks, code/command injections, etc.) are covered.

https://www.sqreen.io/
https://en.wikipedia.org/wiki/Web_application_firewall

SERIES A

❑ Protect your servers and infrastructure

Your servers will be scanned in order to fingerprint your application and locate open services, misconfiguration, etc. You can setup tools to keep these scanners away from your servers.

https://www.digitalocean.com/community/tutorials/how-to-protect-ssh-with-fail2ban-on-ubuntu-14-04

SERIES A

❑ Protect your users against account takeovers

Account takeovers or brute force attacks are easy to setup. You should make sure your users are protected against account takeovers.

https://www.sqreen.io/
https://www.owasp.org/index.php/Blocking_Brute_Force_Attacks
https://security.stackexchange.com/questions/94432/should-i-implement-incorrect-password-delay-in-a-website-or-a-webservice

Monitoring

SERIES A

❑ Audit your infrastructure on a regular basis

With cloud providers, it's easy to start instances and forget about them. You will need to create and maintain a list of your assets (servers, network devices, services exposed etc...), and review it regularly to determine if you still need them, keep them up to date, and ensure that they benefit from your latest deployments.

http://docs.aws.amazon.com/general/latest/gr/aws-security-audit-guide.html
http://searchenterpriselinux.techtarget.com/tip/Creating-an-inventory-with-nmap-network-scanning
https://github.com/Netflix/security_monkey

SERIES A

❑ Check that TLS certificates are not set to expire

You should be using TLS certificates. It can be a hassle to configure and monitor but don't forget to renew them!

https://www.ssllabs.com/
https://serverlesscode.com/post/ssl-expiration-alerts-with-lambda/

SERIES A

❑ Detect insider threats

The most important attacks will come from insider threats. Those can be users or attackers gaining access to privileged user accounts. Make sure you monitor your users to detect attackers before an attack happens.

https://www.sqreen.io/

SERIES B

❑ Get notified when your app is under attack

You will be attacked. Make sure you have a monitoring system in place that will detect security events targeting your application before it's too late. Knowing when your application is starting to get massively scanned is key to stop more advanced attacks.

https://www.linode.com/docs/security/using-fail2ban-for-security#email-alerts
https://www.sqreen.io/
http://alerta.io/

SERIES A

❑ Monitor third party vendors

You're likely to use third party products to manage your servers / payrolls / logs or even just social media. Just like you're likely to be hacked, they can be too. Make sure you follow the news and react immediately after a breach.

https://haveibeenpwned.com/
https://twitter.com/SecurityNewsbot

SERIES B

❑ Monitor your authorizations

Be proactive and be alerted when authorizations or keys binary are changed in your production.

http://techblog.netflix.com/2017/03/netflix-security-monkey-on-google-cloud.html
https://cloudsploit.com/events

http://ossec.github.io/
https://security.stackexchange.com/a/19386

SERIES A

❑ Monitor your DNS expiration date

Just like TLS certificates, DNS can expire. Make sure you monitor your DNS expiration automatically.
https://github.com/glensc/monitoring-plugin-check_domain