WHITEPAPER

2018 CLOUD SECURITY AND COMPLIANCE CHECKLIST

Make This Year's Audit Just Another Day

www.cloudcheckr.com

Cloud Checkr

MAKE THIS YEAR'S AUDIT JUST ANOTHER DAY

A new year, 2018, is upon us, and with it comes another set of audits. There are new regulations to follow and old regulations that still require compliance. Whether this is your company's first audit or tenth, there is always room to improve, and there are ways to make it run smoother and ultimately, to keep you out of political and technical hot water.

If you are viewing the auditor as your enemy, **think again**. Auditors come in to ensure you are properly safeguarding information, that you don't get yourself into a bind. They are there to verify you aren't exposed. Hiding skeletons in the darkest closet in your data center is only going to keep them there and later they will haunt you, not the auditor. The more you view the audit as an opportunity to shine and work with the auditing team, the more they will be willing to help you pass it and succeed in safeguarding data.

If your company is going through a new audit and you don't know what "you don't know," it can be beneficial to solicit help. There are many consulting organizations that will audit you, for a fee, prior to the real audit. They can point out your shortcomings and help you build a Plan of Action and Milestones (POAM). They may open your eyes on actions that cannot be completed by the time the auditors arrive. There is good news here, though. If you have your plan in place and share it with the auditors, they are typically receptive. There are circumstances where you may have a new company you're trying to secure, or you've inherited a mess from the previous security manager. Auditors have seen it all, and the best approach is to **be honest and be prepared**.

What to Expect

Audit types vary, but what is audited often overlaps. For instance, an ISO auditor is going to want to see where you're compliant, whereas as a SOC 2 Type 2 auditor is going to hunt for where you're not. Both auditors are going to look for access controls, logging, encryption, and the like. If your company has been audited before, chances are you have a record of the findings, and the auditors will **expect to see that they have been resolved**, or at the very least, that there has been significant progress toward resolution.

Auditors will send a large set of controls they want verified, often well before they arrive. These control items will be referred to here as **artifacts**. Artifacts can be proof of backup verification, or a privileged user access review, or what VPN ciphers are used, and anything in between. The list will be daunting, so if your company hasn't built an artifact library from previous audits, there will be **a lot** of information to gather. Always verify with your auditor how they want artifacts captured. Some artifacts may be documented controls or procedures you can share with them (such as with a Confluence or Sharepoint repository). The proof can vary depending on the audit. For instance, a SOC 2 Type 1 audit is fine with screenshots of settings. A SOC 2 Type 2 wants to verify (by seeing) that a configuration is set.

Auditors have limited time to spend verifying your artifacts and findings, and as you have spent a great deal of time getting them information, they have to pore over all of it. It may sound counterintuitive, but you should get as much to them as you can early and be as prepped for their visit as possible. If you make them wait for data or stall in providing them with certain artifacts, or in other words, make it difficult for them, they will return the favor—guaranteed.

Have a Pulse on Your Compliance Checklist

You passed your HIPAA audit 10 months ago. You get an email from the auditors saying that they will be requesting the artifacts within the next few weeks. Are you still compliant? Maybe. How can you be sure? If you manually researched all the answers, chances are that there was change, and even if there wasn't, how can you be sure without verifying? This can become a major time sink for you and your resources.

You could take inventory of your auditable items every few months. Even if this is effective, it is still time consuming, and you still only know your status directly after the audit. Luckily, there are tools on the market that can help you achieve continuous compliance. CloudCheckr has native tools that help you to monitor dozens of controls across multiple cloud platforms so you can **know** you are compliant by looking at a dashboard or **be notified** when a control changes. Demonstrating this level of control over your environment will gain a lot of favor from the auditors.

Using a tool like CloudCheckr gives you the ability to run reports and plan for improvement. Some audits, like HIPAA, expect to see you continually improving your security posture each year. If you're constantly chasing artifacts and aren't sure what is still compliant, it is difficult to improve your security posture efficiently. Since several controls overlap in audit types, you can use these dashboards and reports to help you perform each audit without producing redundant screenshots and verification checklists.

Use these tools to keep a scoreboard for your compliance. Have your scoreboard line up with your annual budget for security and show your auditors that your engineers are continuously working to improve the security posture of your environment.

Have Evidence Available

Prepare a controlled, easy-to-access place to manage artifacts. There are many tools on the market. Pick one and organize your artifacts by the request ID. For instance, if "Verify HTTPS is used on Web Servers" is HIPAA SEC1.0.5, then your document or screenshots pertaining to that request should be named HIPAA SEC1.0.5-NGINXConfig or HIPAA SEC105_Findings. **Make it easy** for your auditors to validate your compliance. Don't make them ask you twice or hunt for the answers. If they are unsuccessful in finding your proof, they may just mark the item unresolved. Build an accountability spreadsheet to track who owns the artifact, where it is located, when it was last updated, when it was delivered to the auditor, etc. Once you receive the massive list of requests, it will become difficult to track if you don't methodically account for all actions.

Evidence checklist:

- Keep track of artifacts. You will reuse them. Auditors may request them again later.
- Store documentation in a place that leverages access control and revisions.
- Name your evidence based on the control/request/article, etc. Make it easy for you and the auditor to match up your compliance.
- Get as much data as possible to the auditor before they get on site. It will impress them and set the precedent that security and compliance are high priorities!
- Use a progress tracking sheet, or a "legend," for the audit. Don't rely on emails and status reports to track progress. Don't expect the auditors to do it for you! It is their job to **verify** your data but it is **your job to get it to them**.

Keep Hardening

Now let's dig into the weeds a bit. The real trick to technical compliance is automation and predictable architecture. Without them, you'd have to verify technical controls each time (and who wants to do that?).

Always Install Security Patches

Always install security patches; it is worth saying twice. Whether you're using Windows, Linux, VMWare, or containers, you should apply **all security-related patches** and have an effective way to **verify they are completed**, since auditors will want to see it. If you don't have it yet, use patch repositories that you can control, such as SCCM, a local Yum/Apt/Zypper repo, or AWS Patch Compliance and Patch Groups. That way you can approve incoming patches once and have servers check in locally. Security updates should always be approved as soon as they are available. Patch updates should be automated to deploy monthly at the very minimum and burnt into your images prior to launch. This gives you a predictable benchmark and makes it much faster to achieve an approved security state. The last thing you want is to have autoscaling groups bring up base systems that have to patch before joining load balancers … only 45 minutes left and three reboots until they are ready to serve traffic!

There are numerous guides available for you to evaluate your operating systems and applications. Most are free. Here are a few examples of industry standards:

- Defense Information Systems Agency (DISA) Security Technical Implementation Guides (STIGs) for OSes. There are also STIGs for Active Directory, Group Policy, Web servers, Acrobat, databases—there is a huge list of hardening articles. Find the catalog here.
- Center for Internet Security (CIS) benchmark guides. There are operating system and new technology benchmarks as in Docker, Kubernetes, and Palo Alto Network devices. These benchmarks are free and are located here. There is also a CIS evaluation tool that you can purchase.

These guides are great at providing information about the hardening items and further reading, but using tools makes it much easier to scan and automate remediation of vulnerabilities. A word of warning: **Do not** use tools to blindly remediate vulnerabilities. This can and likely will lead to inoperable systems. Always remediate in a test environment first and rerun integration tests afterward (or perform regression testing).

There are multiple paid tools on the market to scan and remediate systems, and a good free utility for Linux-based systems is OpenSCAP. Microsoft has some basic free tools; a good list can be found here. There are more robust paid products, but ideally, you want to ensure compliance rather than fix once and create an image. This is where configuration tools such as Chef or Puppet come in as strong contenders.

Consider this scenario: Your checklist includes disabling root login to Linux-based systems. An admin logs in remotely, sudos to root, then changes his or her favorite server to allow root access. That machine is now vulnerable and out of compliance, and you don't know it. In contrast, if you take a requirement such as "disable root login in sshd" and create that test and configure it in a puppet module, you can now test functionality, test that it works, and the agent will replace the setting should the admin undo it. This, of course, isn't bulletproof, but it is a much better CI/CD model of ensuring security hardening and compliance.

Once your operating system hardening audit is on track, move to the network. If you're working with Infrastructure as Code, you're in luck. Most can evaluate compliance, and Terraform is an example. Firewalls and network ACLs should be allowing least-privilege access, especially from exposed zones (such as DMZs, or public-facing cloud subnets) to internal service and data layers. Nothing will get an auditor hungry like seeing a network rule with "Allow All." If you aren't using Infrastructure as Code, at a minimum there should be a quarterly review of firewall and network ACL policies and a strict change-control process for updating traffic flow. Auditors will expect this to be a heavily guarded control.

You're on your way: OSes are hardening, the network is automated through code, vulnerability scans are part of the image-building process, integration tests are checking functionality post-remediation. Now what? It's time to throw away the keys. For all non-data-bearing servers, lock down all ports except service ports, remove SSH keys, scramble the administrator password, completely wall off the server. It becomes a black-box service, reducing the attack and exploit vector considerably. Your logs should be shipping and servers should be throwaway so if there is an issue, you can destroy and let another come up in its place. There should be no need to log in and troubleshoot or retrieve information. For data-bearing nodes (databases, legacy file services) maintain least privilege and heavy monitoring. If you can get to this stage of a distributed environment, auditors will love you for it.

Hardening checklist:

- Install all security patches.
- Have an easy way to show patches installed.
- Scan your servers for vulnerabilities (at least quarterly).
- Remediate vulnerabilities within a reasonable timeframe in an automated way, e.g., criticals within 96 hours, highs within 2 weeks, mediums within 60 days, lows within 90 days, and use Puppet or the configuration management tool of choice. Unresolved criticals and highs will set off big alarms with the auditors.
- Bake hardening and patches into images. When servers come up, they should be security- and compliance-"ready."
- Build integration testing into your security if you don't already have it, lest your regression testing be painful.
- Allow least privileges on ports in firewalls, network ACLs, security groups, iptables/firewalld, Windows Advanced Firewalls, and the like. Use Infrastructure as Code if and whenever possible.
- Use industry guides to help you harden
- Don't shoot for 100 percent up front but make reasonable progress. This is what auditors expect to see.
- Use tools to help you scan and harden.
- Keep a compliance dashboard.

Log More, Retain as Required

Auditors want proof. Screenshots suffice for some things, but most often it is the logs that provide evidence. Auditors want to see that the appropriate logs are being shipped and archived, that they are protected, and that they are maintained for the required time (which varies by control). Each control will specify the minimum; always check the documentation.

Most importantly, when your application touches sensitive data, such as PII, ePHI, or PHI, **ensure that access is logged**. Wherever you choose to store your sensitive data, **ensure that it is accredited for the control**. For example, if you are placing ePHI in Amazon S3, ensure that Amazon S3 is certified for HIPAA (it is). Ensure that all mechanisms that retrieve sensitive data are using encryption. Better yet, encrypt everywhere. Don't leave a control or system to chance by not configuring encryption. If you are handling sensitive data and are subject to a control such as SOC or HIPAA, and you aren't using encryption at rest/in transit for the entire control of that data, it will be a finding. Such a finding is bad, but it is even worse if an attacker was able to obtain the data or it was accidentally exposed, which would be a reportable offense. If the data is multi-tenancy (i.e., customers share it), this would be reportable to ALL the customers.

Logging checklist:

- Use Rsyslog, Windows Event Log Forwarding, third-party tool, etc. for log shipping, and use a method to ship logs securely for analysis, storage, and archiving.
- Retain logs for minimum control requirements (often 1-7 years).
- Ensure that storage of logs with sensitive data is encrypted (this includes backups!).
- Ensure that access to sensitive data is logged.

Encryption checklist:

- Use hardware encryption for encryption at rest. This will reduce the impact on performance. Cycle the key at least annually. If hardware encryption isn't available, encrypt disks with software (and expect a performance hit).
- For encryption in transit, ensure that HTTPS or SSL is used with medium-strength ciphers at a minimum (over 128 bits) and strong hashes. Only terminate encryption at the point of processing.
- Safeguard all private keys for certificates and public keys.
- Encrypt data in databases if you can handle the performance loss. It's an extra layer of protection.
- Encrypt backups with AES-256 or stronger encryption.
- Encrypt stored files (think S3) with AES-256 or stronger encryption.
- Use VPN tunnels with at least AES-256 or stronger encryption.

Review Access

Every audit will have elements of access review. If your business manages sensitive data, the auditors will be really focused on access to that data. There should be a ticketing and authorization process to provision access to any system that touches sensitive data. That list of users and privileged users should be reviewed quarterly for every system.

If you have a system to provision access to customer credit card data (think PCI), an auditor is going to expect to see how the access was requested, who authorized it, when it was authorized, when it was reviewed, and if it was revoked when the employee no longer needed it. Whether the system is automated (better), or completely manual (still okay in most cases), the auditor will expect to see this. If your environment isn't heavily federated, the sprawl of disparate systems and controls can become overwhelming. In these cases, it's best to use an application owner for each system who reviews access quarterly. Don't let the sprawl turn into accidental access, or worse, unauthorized access.

Review checklist:

› Assign application owners and audit quarterly.
› Build automation to provision and revoke access.
› Strictly track and control access to sensitive data (and be able to show it).

SUMMARY

The auditors are going to show up. The better prepared you are, the smoother the audit will be. They aren't there to help you; they are there for the industry to ensure you are safeguarding customers' information. They are there to catch you before an attacker does, so treat them as part of your safeguarding team. Compliance is tightening in 2018, and newly introduced regulations such as GDPR or MiFID will bring additional challenges to organizations. Get ahead of your security demands by improving your security posture. This will please the auditors and keep your organization out of a security article.

Need CloudCheckr for your organization? Learn more at **www.cloudcheckr.com**.

Cloud Checkr

342 N GOODMAN ST,
ROCHESTER, NY 14607

1-833-CLDCHCK

www.cloudcheckr.com