International Journal of Engineering Applied Sciences and Technology, 2024
Vol. 9, Issue 03, ISSN No. 2455-2143, Pages 136-145
Published Online July 2024 in IJEAST (http://www.ijeast.com)

SECURITY AND PERFORMANCE ENHANCEMENT USING SHORT-LIVED CERTIFICATES. A COMPREHENSIVE ANALYSIS

Manish Sinha
Senior Software Engineer
Facebook (Meta), San Jose, California, USA

Abstract—**Short-lived certificates (SLC) are a comparatively new approach to improving software systems' security, performance, and effectiveness. A typical certificate validity can range anywhere between 12 months to multiple years. Short-lived certificates dramatically reduce the validity period to months, sometimes days, and, for some situations, even hours. By reducing the validity, they minimize the scope of damages if the private key is ever compromised. This paper examines the benefits of short-lived certificates, which include but are not limited to improved security posture and reduced reliance on certificate revocation infrastructure and mechanisms. We will approach the automation and infrastructure changes that made this new approach practical. We will explore methods of implementing such a system at scale. We will analyze the trade-offs between security, reliability, performance, and operational complexity. We will additionally cover the network and infrastructure changes that might be necessary for a reliable implementation of Short-lived certificates infrastructure.**

Keywords—**Cryptography, X509 Certificates, Security Best Practices, Public Key Infrastructure**

I. INTRODUCTION

We rely heavily on Transport Layer Security (TLS), a successor to Secure Socket Layer (SSL). TLS is built on the foundation of Digital Certificates (X509), which are used for two primary objectives: Encrypting data in flight and validating the identity of the communicating entities[1]. The digital certificates have an issue and an expiration date. Once the digital certificate has been issued, it has to be deployed to the server to handle the communication on behalf of the entity. This deployment process is intrusive, and mistakes can cause outages. In the past, digital certificates were issued for 12 months and sometimes more than a year[2]. Due to increasing threats, short-lived digital certificates are appearing. These certificates have significantly shorter validity periods – sometimes months or even days[3].

The adoption of SLC has been a culture and mindset change and marks a major shift in the digital trust and security industry. If a private key is compromised, the attacker runs against the clock before the certificates expire and the key becomes worthless[4]. This improves security, and we rely less on complex infrastructure like Certificate Revocation List (CRL)[5]. In small devices with limited bandwidth or devices behind a firewall that cannot access the Internet, CRL is of limited to no use.

What changed

Short-lived certificates (SLC) are not new, as they fundamentally reduce the validity period. What changed in the last decade is better automation, practices, and tools to manage digital certificates. This has made widespread adoption of SLC much more feasible[6]. Let's Encrypt, a popular free digital certificate issuer, has a policy of issuing certificates with only 90 days of validity. They provide no exception to this policy[7]. It has forced its users to develop robust certificate deployment, rotation infrastructure, and policies. This has further increased the cultural adoption of the concept of short-lived certificates.

Performance impact

The problem still remains, as certificate issuance is much more computationally expensive than validation. This means we run into performance problems as we scale up our SLC infrastructure. Now, we can see that Security and Performance are at odds. All hope is not lost, and we can explore approaches to reduce the performance hit.

Reliability Impact

Short-lived certificates cost not only performance but also reliability. Increased certificate rotations can cause outages if the process is not managed with care[8]. Such outages can lead to major service disruptions and negatively affect the company's revenue. Many companies have legacy infrastructure, and their behavior is not well-known or documented.

Objective

We will provide a comprehensive analysis of short-lived certificates, examining their benefits, challenges, and impact on the future of web security. We will take an example of a real implementation of such a system. With a critical eye, we hope this paper contributes to the evolution of security standards and best practices.

II. CONCEPTS

A. Certificate Revocation List and Online Certificate Status Protocol –

Traditional certificate checking primarily involves two methods – Certificate Revocation List (CRL) and Online Certificate Status Protocol (OCSP). They work using different techniques. CRL is a list of certificates marked as revoked and maintained by Certificate Authorities (CA)[9]. Due to its size, this file can be broken into smaller files. The list is static and is regularly updated by each CA[10]. The client has to download and parse the certificates. OCSP, on the other hand, is a protocol where the client can request to validate the revocation state. Since the client only has to check for one certificate instead of downloading the entire list, OCSP is faster and less burdensome on the client's resources[11].There is an extension protocol called OCSP Stapling, where the server is responsible for fetching the certificate's revocation state from the OCSP server and stapling it to the response sent to the client[12].

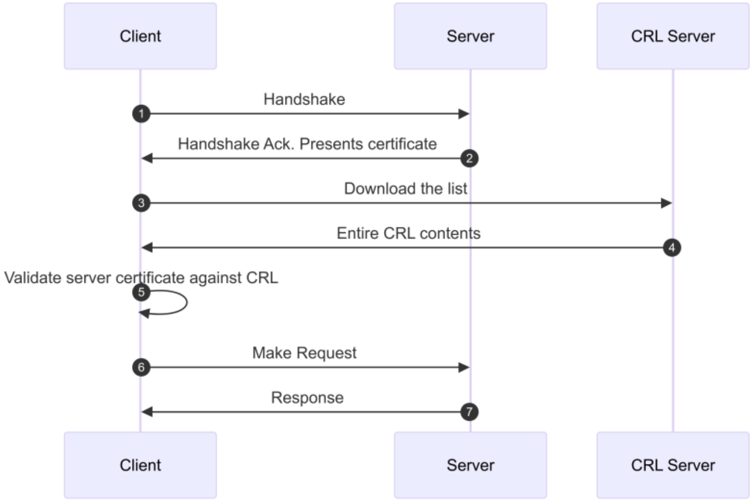


Fig 1: How Certificate Revocation List works

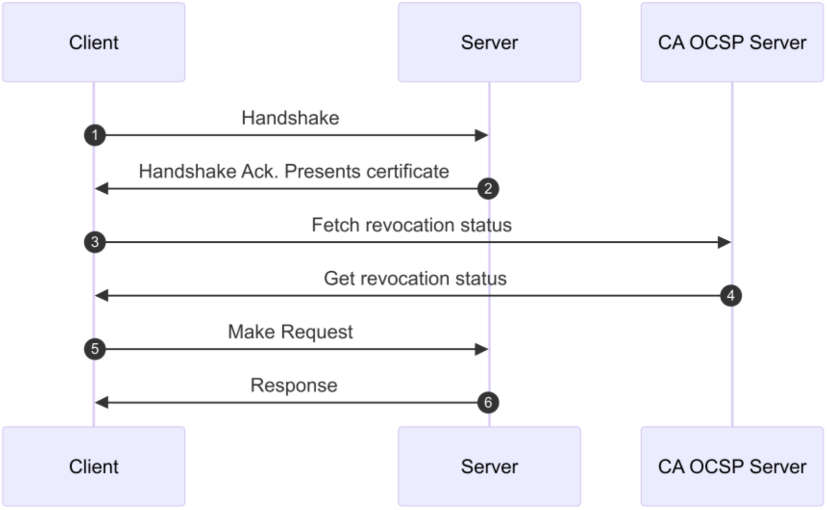


Fig 2: How OCSP works

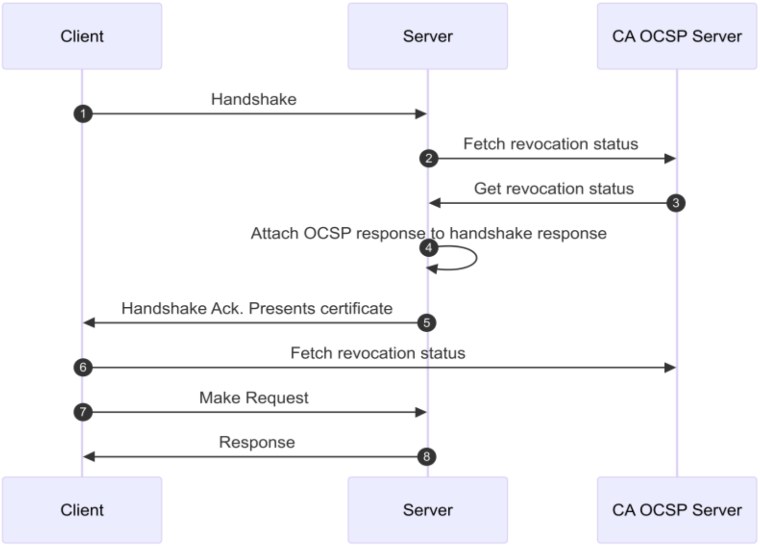

Client
Server
CA OCSP Server
1 Handshake
2 Fetch revocation status
3 Get revocation status
4 Attach OCSP response to handshake response
5 Handshake Ack. Presents certificate
6 Fetch revocation status
7 Make Request
8 Response
Client
Server
CA OCSP Server

Fig 3: How OCSP Stapling works

B. Working around CRL and OCSP –

Short-lived certificates are fundamentally an alternative to CRL and OCSP; since the certificate is short-lived, we can avoid verifying it[13]. This is true since checking for CRL and OCSP lies primarily on the client or browser. Since such an operation is expensive, some clients might skip such verification. SLC functionally relieves the client from the responsibility of certificate verification.

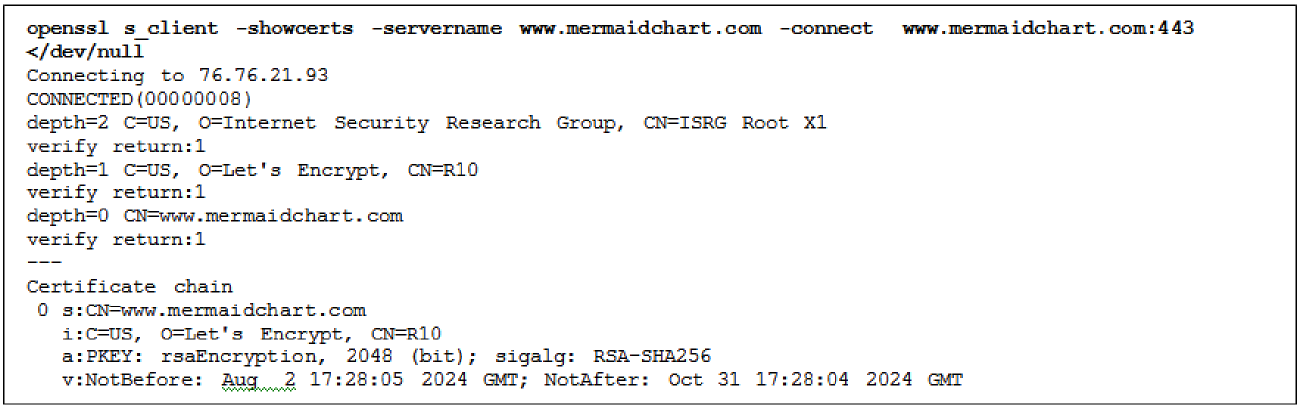
SLC also improves performance since CRL causes a performance hit, and OCSP does not mandate HTTPS, which means it is open to man-in-the-middle attacks when HTTP is used instead of HTTPS[14].

C. Certificate Validity Period –

Let’s Encrypt pioneered in making shorter validity certificates a commonplace occurrence. Despite their 90-day validity period, which is much shorter than the typical 365-day validity period of most Certificate Authorities, it is still too long for a typical short-lived certificate. The 90-day validity would still require CRL and OCSP since this duration is enough for most attackers to successfully exploit major breaches to either the private key or a similar heartbleed-style attack[15].

The question remains: What is the best duration for a short-lived certificate? Unfortunately, there is no correct answer. The correct answer might be anywhere from 8 hours to multiple days. In corporate settings, the computers assigned to the employees can authenticate using certificates[16], which should ideally be as short-lived as possible because accounts attached to humans have much more expansive permissions. There can be more than one certificate issued to an entity (the employee computer in this case), where a different certificate is used to access more critical services. Suppose the employee is accessing customer or employee data. In that case, those servers should only accept a certificate with a validity of 4 hours, whereas servers that handle regular day-to-day work can have a validity of a few days.

We can always use openssl command to get the certificate validity for any website.



```
openssl s_client -showcerts -servername www.mermaidchart.com -connect  www.mermaidchart.com:443
</dev/null
Connecting to 76.76.21.93
CONNECTED(00000008)
depth=2 C=US, O=Internet Security Research Group, CN=ISRG Root X1
verify return:1
depth=1 C=US, O=Let's Encrypt, CN=R10
verify return:1
depth=0 CN=www.mermaidchart.com
verify return:1
---
Certificate chain
 0 s:CN=www.mermaidchart.com
   i:C=US, O=Let's Encrypt, CN=R10
   a:PKEY: rsaEncryption, 2048 (bit); sigalg: RSA-SHA256
   v:NotBefore: Aug  2 17:28:05 2024 GMT; NotAfter: Oct 31 17:28:04 2024 GMT
```

Fig 4: Checking the validity of a digital certificate

The certificate was issued on August 2nd and expires on October 31st, which is exactly 90 days.

D. Certificate Authority–

The choice of certificate authority is critical in deciding to issue short-lived certificates. Not all Certificate Authorities provide the necessary support and tooling to achieve this result. DigiCert and Let's Encrypt are the two providers that actively support such infrastructure and guide users[17].

E. Intermediate Certificate Authority (ICA)–

The intermediate certificate authority forms a bridge in the chain of trust from the root certificate authority to the end-user certificate using Public Key Infrastructure. There can be more than one intermediate certificate in a chain of trust [18]. The root CA is offline and is kept highly secure using proper physical security. The root certificates do not sign the end-user certificates for the simple reason that it increases the chances of the private key getting compromised. Instead, root certificates sign intermediate certificates, validating their authenticity and duration.

ICA relieves the burden of certificate management, as we can have more than one ICA signed by a single root CA. In the case of a compromise, it is much easier to revoke an ICA rather than a root CA. Revocation of a single ICA would not affect other ICAs signed by the same root CA [19]. ICAs can help reduce the operational burden of using a single CA in an organization. An ICA can sign child ICAs for each department, which makes the top-level ICA more secure.

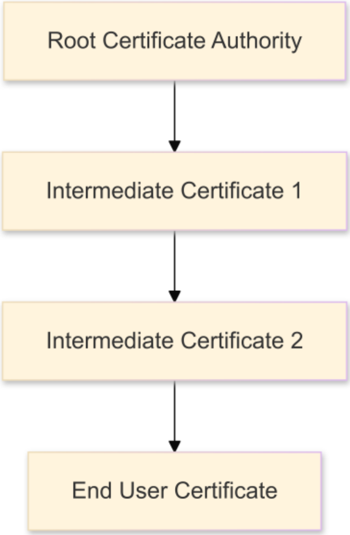

Root Certificate Authority
Intermediate Certificate 1
Intermediate Certificate 2
End User Certificate

Fig 5: An example of a certificate chain of trust

III. CHALLENGES

***A.* Inadequate Automation Infrastructure –**

While we might consider short-lived certificates as the best practice, providing us with a reduced window for attack, they do come with their own set of problems, especially in terms of automation infrastructure. Inadequate internal automation infrastructure is a glaring problem when using this approach. If your organization needs more automation infrastructure, it can lead to service disruption, deteriorating operational overhead, and possibly introducing security vulnerabilities[20].

Due to the lack of automation infrastructure, the organization might try to manually renew certificates at scale. This is fraught with issues since manual renewals can introduce mistakes and are very expensive.

The issues with inadequate automation infrastructure continue beyond there. It can lead to delayed or missed renewals, potentially causing critical services to become unavailable by customers who cannot access them due to untrusted or expired certificates being rejected by the clients. The systems can contain heterogeneous pieces, including external off-the-shelf solutions[21].

***B.* Inadequate computational resources to issue as scale –**

The growth of short-lived certificates comes at a high cost. Issuing new certificates at scale puts a lot of demand on the computational resources in the certificate management systems. This can put a lot of stress on existing infrastructure, which needs to be carefully designed to not become a single point of failure[22].

Traditional certificate issuance systems can quickly overwhelm and cause processing delays and a backlog, calling operational efficiency into question. This issue can be severe for large organizations that manage tens of thousands of certificates, including client certificates. Such inefficiencies, which might seem small, can slowly compound and cause an outage[23].

To address this issue broadly, organizations should invest in scalable, horizontally scalable computational resources by adding hardware, optimizing algorithms, and using well-known techniques to avoid thundering herd problems [24].

C. Inadequate monitoring solutions –

It's not enough to have adequate infrastructure and computing capacity. A big chunk of work has to go into ensuring that our monitoring and observability capabilities are top-tier. We must run production tests to ensure each endpoint presents the correct certificate and matches our expectations. The expectation keeps changing since the certificate itself keeps getting updated. With massive amounts of updates in metadata, it is possible for distributed databases that hold such information to lose consistency. Even if we miss a single certificate expiry, it can lead to an outage, the severity of which can be simple, like the client cannot access server resources, or severe enough that we are no longer able to access customer contact information and the sales department comes to a halt[25].

The monitoring solution that we need should be real-time and able to handle large amounts of data, churning analytics, and presenting real-time validation data for either manual observation or automated systems to identify and flag anomalies.

D. Clock Skew –

Clock Skew is usually not a big problem until there is a problem with time synchronization across different servers. It can be an elusive problem, difficult to debug, and requires the engineers to be even aware of it. It can be surfaced by monitoring solutions and providing enough information on their failures for a skilled engineer to identify the problem[26].

IV. APPROACHES AND SOLUTION

A. Start with a high-level design –

Before we create an automation, compute, and monitoring solution to issue short-lived certificates at scale, we need to start with a high-level design of the entire infrastructure. We need to start with the non-technical requirements, sort them, and make judgment calls on their relative importance. Non-technical requirements lead to technical requirements, and even at this stage, we need to understand the scope and scale adequately. Once we have the design, we must enforce the implementation to align with the design and regularly assess whether we are deviating from the expected [27].

Every good design requires proper assumptions—both business and technical. Business assumptions could relate to capabilities that focus on humans and their expectations, whereas technical assumptions relate to the capabilities of automation systems. The assumptions should be documented and reassessed to ensure they still meet the organization's changing needs[28].

Care should be taken to abstract the implementation details properly, and only the relevant aspects should be surfaced. The specifics that surface should be contextual. The executives need to focus on the overall availability reliability and satisfaction score. The Directors or Senior Managers need to focus on cost efficiency. Engineers must focus on data like error rate and latency aggregation metrics.

B. Build incrementally –

Instead of solving all the problems simultaneously, starting small and building incrementally is vital. The first order of business is to create a Minimum Viable Product (MVP)[29], a stripped-down, bare-basic product that can contain bugs or poor code quality but focuses on delivering a use case. When demoed, we can get feedback immediately from the relevant stakeholders. This ensures our understanding of the high-level design aligns with the stakeholders' understanding[30].

During MVP implementation, we encountered some technical blockers. We would have to revisit our design and the assumptions that accompany it. We would then have to rework the plan to ensure the implementation continues to match what we have decided. This is broadly called "course correction" [31] and is industry jargon.

With the platform maturing in the later stages of development, additional features and integrations can be added, such as monitoring certificate validity from internal and external networks. As we see, this feature is optional and should be left for later[32]. We should first focus on the critical and basic functionality and build out the infrastructure in a layered manner.

C. Staged Rollouts –

One issue we have to worry about is the "Thundering Herd problem" [33]. When we flip the switch on a new feature or action, all the consumers try to consume a service, which causes a Denial-Of-Service.

The rollout of the new infrastructure configuration or even the renewal of certificates should be done in batches, each of unequal size, in a staged, structured manner[34]. When a new configuration is deployed on the infrastructure, it risks causing an outage. In such a case, new certificate renewals and deployment should happen exponentially. In this case, we artificially limit the number of certificates renewed in the next cohort. If it goes well, we will increase the number of certificates in the next cohort until we reach the desired batch size[35].

D. Have a robust testing stage –

It should be self-evident that we should not be testing production. It is fraught with issues, causing outages and introducing security risks. We need a robust testing strategy and infrastructure to validate our changes and ensure they behave as expected[36]. Well-designed testing environments closely mimic the production environment, with noticeable differences – no customer or personally identifying data should be present in the testing environment. We should be using dummy or synthetic data.

The testing stage should be able to run unit tests and integration tests first to catch the issue as soon as possible. Once those tests pass, the code is built as an artifact and deployed to the staging environment. At this point, we should be running another suite of tests that mimic actual customer behavior[37].This is very similar to black-box testing.

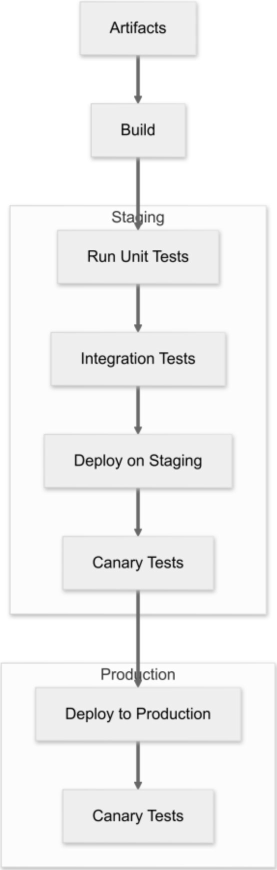


Fig 6: A typical Continuous Integration Stage with Staging

E. **Secure the private key –**

The entire short-lived certificate infrastructure and concept are as secure as the safety of private keys. Private keys are the weakest link in this infrastructure, and we must take the utmost care to keep them from falling into the hands of malicious actors[38]. This discussion covers three private keys: the root CA Private Key, the ICA Private Key, and the end-user Private Key.

The root CA Private Key is the most critical of them all, as it can affect all the certificates directly and indirectly issued by it. On the other side of the risk, end-user Private Key leaks only affect that specific user.

The best way to store Private Keys is offline, away from any machine with internet access. Hardware Security Modules (HSM)[39] come in handy. They are highly specialized hardware designed to safeguard cryptographic keys in a tamper-resistant hardware module. They offer features like auditing, access control, and physical security. HSM is useful for storing Root CA and ICA Private Key.

The root CA Private Key should be kept in HSM and disconnected from the internet. ICA Private Key should be kept in HSM, which is connected to a computer in its network and can only be accessed from another web server, which sits in the public network and is accessible to clients requesting a new certificate. The end-user certificate private key should be kept in HSM and accessed from a web server like Apache[40].

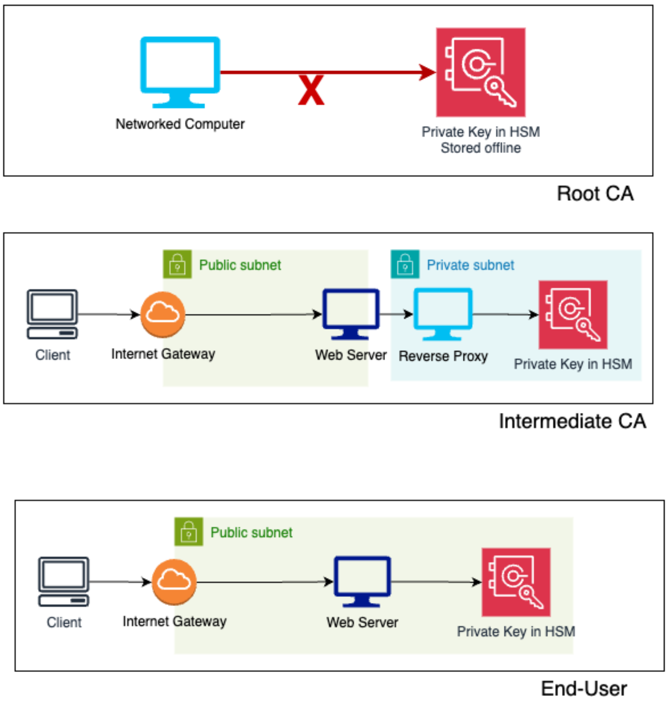


Fig 7: How to store and secure Private Keys using HSM

F. Regular rotation of the CA –

It's crucial to regularly rotate the Root Certificate Authority and Intermediate Certificate Authority for issuing certificates. It reduces the surface area of attacks when the private key has been leaked unknowingly[41].

The frequency of key rotation depends on multiple factors, including the number of end-user certificates generated, the duration of usage of Root CA or ICA, and the broader appetite for security risk in the organization. Such rotations can be disruptive, so some organizations might rotate less frequently than others. It's important to strike a delicate balance between operational efficiency and security. Industry best practices are to rotate Root CA between 10-15 years and Intermediate CA between 2-5 years [42].

The root CA and ICA rotation process needs careful planning and execution, with enough notice given to external customers and internal teams that depend on the certificates. The process involves generating a new key pair, safely storing the private key in HSM, securely distributing the public key, and revoking existing public keys [43].

G. Robust Monitoring –

A robust monitoring system is important to running such a system at scale. It will help us identify and mitigate risks, ensuring the integrity of digital certificate management infrastructure.

Two key components of this monitoring system are centralized logging and auditing capabilities. These systems aggregate logs from various systems, including end-user machines, certificate validation canaries, and certificate issuance systems. We need auditing capabilities to check if our certificate management infrastructure has been compromised. We must audit each action where the Private Key for Root CA and ICA is loaded into memory. Centralized logging gives organizations a holistic view of all activities and helps with anomaly detection[44]. We can use this data for alarms, take automated actions for some threats, and alert a human operator for alarms attached to critical issues.

Apart from automated monitoring, we need to perform regular audits. These involve understanding the context of the data and looking at the logs with a different eye, with the kind of intelligence computers are not currently capable of or are too unreliable to perform. We should commit to regular reassessments of infrastructure security posture and make recommendations for improvement[45].

H. Documented Operational Plan –

The operational plan is a roadmap that outlines the processes, procedures, and responsibilities of managing such an infrastructure. A RACI matrix should exist [46]for Responsibility, Accountability, Consultation, and Information. The list of the team's individuals or roles is on the other axis. This matrix needs a checkmark where the roles and actions align.

The Operational Plan promotes standardization and best practices across the organization. It contains a summary and detailed instructions for each action and responsibility. Documenting them reduces the knowledge gap and information siloing. Delineating responsibilities ensures people do not step on each other's toes but work together cohesively.

I. Standardized Incident Response Plan –

An Incident Response Plan is required to deal with security incidents related to certificates—either with certificates themselves or with the certification management infrastructure. It outlines the steps and actions during a breach, specifying who must be involved and the communication method[47]. It should have documented steps of bringing people together, going as specific as how to start a conference call and invite people to it. During the incident, there was not enough time for the on-call to determine how to start a conference call.

Swift coordination between the people involved is of the utmost importance. If there is chaos, time is lost, and the dangers of damage increase. There should be a method to escalate the issue[48] and a proper system to track all the related actions, logs, and metrics. The Incident Response Plan should also specify how to classify the severity level of the incident.

The Incident Response Plan should allow organizations to learn from their mistakes. It should promote transparency and encourage a no-blame culture. There should be steps on mitigation and how post-incident reviews are conducted[49].

V. CONCLUSION

Short-lived certificates have significantly been developed in the last decade, and their limited duration simplifies the historical complexities of certificate revocation infrastructure. Reducing the window of validity reduces the window of vulnerability, thereby improving the security posture. This marks a change in thinking and willingness to adopt a new method focusing on the ever-increasing security risk.

The transition to short-lived certificates has been fraught with pushbacks in the past. Browsers and Certificate Authorities were always reluctant to reduce the validity period. However, with recent improvements in automation, knowledge, and the adoption of an open mindset, short-lived certificates are seeing a surge in usage.

In the coming years, short-lived certificates will play an increasingly important role in safeguarding online communication and personal data. This will catalyze the widespread adoption of the 90-day expiration for website digital certificates that Let's Encrypt has pushed so hard for.

Not having to rely primarily on certificate revocation infrastructure is instrumental in the performance improvements of the web pages, which can be felt by customers too.

VI. REFERENCE

[1] IBM. (2024, January 31). IBM MQ 9.3. https://www.ibm.com/docs/en/ibm-mq/9.3?topic=tls-how-provides-identification-authentication-confidentiality-integrity

[2] Apple Support (2023, August 21). About upcoming limits on trusted certificates - Apple Support. https://support.apple.com/en-us/102028

[3] Swientek, T. (2024, February 25). What are short-lived certificates? Teleport. https://goteleport.com/learn/what-are-short-lived-certificates/

[4] Patil, K. (2023, August 18). Seven ways short-lived certificates help reinforce security. AppViewX. https://appviewx.com/blogs/seven-ways-short-lived-certificates-help-reinforce-security/

[5] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., & Polk, W. (2008). Internet X. 509 public key infrastructure certificate and certificate revocation list (CRL) profile (No. rfc5280).(pg 13)

[6] Steven Hall (2024, May 28)The evolution of automating certificate management. https://www.globalsign.com/en/blog/evolution-of-automating-certificate-management

[7] Josh Aas (2015, November 9)Why ninety-day lifetimes for certificates? https://letsencrypt.org/2015/11/09/why-90-days.html

[8] "Palanisamy, M. ". (2024, February 16). "Seven Ways to Navigate Certificate Management Complexity." RSA Conference. https://www.rsaconference.com/library/blog/seven-ways-to-navigate-certificate-management-complexity

[9] Rahul Awati, Michael Cobb (n.d.) Certificate Revocation List (CRL) https://www.techtarget.com/searchsecurity/definition/Certificate-Revocation-List

[10] CRLs. (n.d.). https://docs.digicert.com/en/digicert-one/ca-manager/ca-manager-walkthrough/crls.html

[11] J. L. Munoz, J. Forne and J. C. Castro, "Evaluation of certificate revocation policies: OCSP vs. Overissued-CRL," Proceedings. 13th International Workshop on Database and Expert Systems Applications, Aix-en-Provence, France, 2002, pp. 511-515

[12] What is OCSP Stapling? (n.d.). https://knowledge.digicert.com/quovadis/ssl-certificates/ssl-general-topics/what-is-ocsp-stapling

[13] Topalovic, E., Saeta, B., Huang, L. S., Jackson, C., & Boneh, D. (2012, May). Towards short-lived certificates. Web. (pg 6)

[14] Are federally operated Certificate Revocation Services (CRL, OCSP) also required to move to HTTPS? (n.d.). cio.gov.https.cio.gov/guide/#are-federally-operated-certificate-revocation-services-crl-ocsp-also-required-to-move-to-https

[15] Aaron Gable (2022, September 7) A new life for certificate revocation lists. https://letsencrypt.org/2022/09/07/new-life-for-crls

[16] Instasafe Marketing. (2024, March 11). What is Client Certificate Authentication? | InstaSafe. Zero Trust Blog. https://instasafe.com/blog/what-is-client-certificate-authentication/

[17] Nelson, M. (2023 June 14). Why Short-Lived certificates & automation can be beneficial. DigiCert. https://www.digicert.com/blog/short-lived-certificates-and-automation

[18] Wazan, A. S., Laborde, R., Barrère, F., & Benzekri, A. (2017). A formal model of trust for calculating the quality of X.509 certificate. Security and Communication Networks, 2017, (pg 1-18)

[19] Liang, J., Jiang, J., Duan, H., Li, K., Wan, T., & Wu, J. (2014). When HTTPS meets CDN: A case of authentication in delegated service. In 2014 IEEE Symposium on Security and Privacy (pp. 67-82)

[20] SSLTrust. (n.d.). A guide to Intermediate Certificates. SSLTrust. https://www.ssltrust.com/help/setup-guides/intermediate-certificates-guide

[21] Chung, T., Lok, J., Chandrasekaran, B., Choffnes, D., Levin, D., Maggs, B. M., ... & Wilson, C. (2018). Is the Web Ready for OCSP Must-Staple?. In Proceedings of the Internet Measurement Conference 2018 (pp. 105-118).

[22] Stark, E., Huang, L. S., Israni, D., Jackson, C., & Boneh, D. (2012). The case for prefetching and prevalidating TLS server certificates. In Proceedings of the 2012 Network and Distributed System Security Symposium (pg 10)

[23] Liu, Y., Tome, W., Zhang, L., Choffnes, D., Levin, D., Maggs, B., ... & Wilson, C. (2015). An end-to-end measurement of certificate revocation in the web's PKI. In Proceedings of the 2015 Internet Measurement Conference (pp. 183-196).

[24] Szalachowski, P., Matsumoto, S., & Perrig, A. (2016). PoliCert: Secure and flexible TLS certificate management. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (pp. 406-417).

[25] D. Kumar et al., "Tracking Certificate Misissuance in the Wild," 2018 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 2018, (pp. 785-798), doi: 10.1109/SP.2018.00015.

[26] K. Arvind, "Probabilistic clock synchronization in distributed systems," in IEEE Transactions on Parallel and Distributed Systems, vol. 5, no. 5, (pp. 474-487), May 1994

[27] M. Sefika, A. Sane and R. H. Campbell, "Monitoring compliance of a software system with its high-level design models," Proceedings of IEEE 18th International Conference on Software Engineering, Berlin, Germany, 1996, (pp. 387-396)

[28] Mamun, M. A. A., & Hansson, J. (2011). Review and challenges of assumptions in software development. In Second Analytic Virtual Integration of Cyber-Physical Systems Workshop (AVICPS). (pp 3-4)

[29] Moogk, D. R. (2012). Minimum viable product and the importance of experimentation in technology startups. Technology Innovation Management Review, 2(3).

[30] Olsen, D. (2015). The lean product playbook: How to innovate with minimum viable products and rapid customer feedback. John Wiley & Sons. (pp. 4-7)

[31] Lortie, J., Cox, K. C., Kelly, S., & Bolivar, T. (2022). Two-factor learning for launch: How entrepreneurs can increase the probability of positive responses to their minimum viable products. Entrepreneurship Education and Pedagogy, 5(4), (pp 523-546)

[32] Graham, D. R. (1989). Incremental development: review of nonmonolithic life-cycle development models. Information and Software Technology, 31(1), (pp 7-20)

[33] Matt Godbot (n.d.) IO Completion Ports https://xania.org/200807/iocp

[34] Behyad Ebadifar (2023 September 26) A developer’s guide to staged rollouts for the Amazon Appstore https://developer.amazon.com/apps-and-games/blogs/2023/09/guide-to-staged-rollouts

[35] Smith, A., & Johnson, B. (2021). Phased deployment strategies for enterprise-wide security implementations. International Conference on Cybersecurity Practices, (pp 45-58)

[36] Humble, J., & Farley, D. (2010). Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation. Addison-Wesley Professional. (pp 83-101)

[37] Dustin, E., Rashka, J., & Paul, J. (1999). Automated Software Testing: Introduction, Management, and Performance. Addison-Wesley Professional. (pp 336-334)

[38] Adams, C., & Lloyd, S. (2002). Understanding PKI: concepts, standards, and deployment considerations. Addison-Wesley Professional. (pg 89)

[39] Mavrovouniotis, S., & Ganley, M. (2013). Hardware security modules. In Secure Smart Embedded Devices, Platforms and Applications (pp. 383-405). New York, NY: Springer New York.

[40] Apache HTTP Server Luna HSM - Integration Guide | Thales. (n.d.). https://cpl.thalesgroup.com/resources/encryption/apache-http-server-luna-integration-guide

[41] Barker, E., & Roginsky, A. (2019). Transitioning the use of cryptographic algorithms and key lengths. NIST Special Publication 800-131A Rev. 2. National Institute of Standards and Technology.

[42] CA/Browser Forum. (2021). Baseline Requirements for the Issuance and Management of Publicly-Trusted Certificates, v1.7.6. CA/Browser Forum.

[43] Chokhani, S., Ford, W., Sabett, R., Merrill, C., & Wu, S. (2003). RFC 3647: Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework. Internet Engineering Task Force (IETF). (pg 33)

[44] Hsu, C. W., & Slagell, A. J. (2015). Scalable log visualization for the analysis of digital certificate authority activity. In Proceedings of the 2015 ACM Workshop on Visualization for Cyber Security (pp. 1-8).

[45] NIST. (2018). Framework for Improving Critical Infrastructure Cybersecurity, Version 1.1. National Institute of Standards and Technology. (pg 20)

[46] Miranda, D. (2024, June 4). What is a RACI chart? How this project management tool can boost your productivity. Forbes Advisor. https://www.forbes.com/advisor/business/raci-chart/

[47] Cichonski, P., Millar, T., Grance, T., & Scarfone, K. (2012). Computer security incident handling guide (NIST Special Publication 800-61 Revision 2). National Institute of Standards and Technology. (pp 13-18)

[48] European Union Agency for Cybersecurity. (2019). Good practices in incident management. ENISA.https://www.enisa.europa.eu/publications/best-practices-for-cyber-crisis-management

[49] Ahmad, A., Maynard, S. B., & Shanks, G. (2015). A case analysis of information systems and security incident responses. International Journal of Information Management, 35(6), (pp 717-723)