Bill F. Dimmick

Senior Staff Software Engineer

me@billdimmick.com | linkedin.com/in/dimmick | github.com/bdimmick | (919) 414-9634 | Seattle, WA | v2024.03

Summary

WIth 20+ years of professional industry experience, I strive for excellence in solving complicated problems in information security and data privacy with distributed systems and technical leadership. In addition to the technical aspects of my day-today, I also seek to mentor ICs and level up organizations in general.

Work History and Major Accomplishments

Rippling (2024 - Current)	**Product Security** ● SAST and SCA initiatives: Took ownership and led our team's static code and dependency supply chain analysis, including ensuring that rules were correct and properly tested. ● RCE Remediation Project: PDF File Processing - ensured that all PDF rendering and processing from customer-supplied inputs were run in a "clean room" runtime, reducing the chance of an RCE to zero. ● RCE Remediation Project: XML and CSV safe handling - similar to the above, safely removing the use of unsafe libraries with vulnerabilities with safe drop-in replacements to mitigate potential RCE in our services and also protecting customer usage of downloaded data. ● SSRF Remediation Project - Built a replacement for Python's `requests` and `urllib3` libraries to prevent SSRF traversal attacks. ● Bug Bounty and Incident Handling - General partnership with non-security teams to help mitigate their vulnerabilities, including fixing their code and getting them to review the PRs to keep those team on schedule while also improving their security strategy.
Stripe (2022 - 2023)	**SaaS Security** ● Overall reviewer for technical designs written by other engineers, especially around keeping the team's scope to our charter. ● Negotiated agreements between our team and other concerned teams around which internal access groups would be synced to GSuite. ● Mentored engineers on a broad scope of topics, from career growth to when to use certain technologies. **Secure Endpoint Access** ● Used terraform to stand up AWS resources to replace legacy infra with more modern and stable infra. ● Designed a safer replacement for a tunneling system which was sometimes used unsafely by engineers to demo features to third parties. ● Major refactors and cleanup to legacy Go code for VPNs and egress proxies to make the code testable and adding unit tests to assure code quality remained stable.
Twitter (2013 - 2022)	**Cryptographic Key Distribution System (TSS):** ● Identified critical vulnerabilities in the existing system and advocated for its replacement, including a "buy vs. build" document for solution options. ● Technical Lead for the design, ensuring that the technology choices would

be secure and also operate at Twitter scale and availability.
- Threat modeled the design and engaged Infosec to ensure the resulting solution would be as secure as possible.
- Core developer of the system, with 90% code contribution for the production launch of 3 services in Scala on ~100 hosts and a Python daemon on ~50000 hosts across 3 datacenters.
- Created ~50 pages of documentation for end users to easily use the product and for on-calls to deploy and maintain the service while keeping Tier 0 three nines SLAs.
- Gathered feedback post-launch across all teams at Twitter, acting as a project manager to improve availability and user experience.

Service to Service Authentication:
- As Technical Lead, chose mTLS as the solution, obtained buy-in from other stakeholders, and designed the parts of the X.509 SANs which would work for all services at Twitter.
- Guided and code reviewed the implementations for Java, Scala, and Python, ensuring 90% of production services could enable the solution in an easy and secure way.
- Company-wide support and advocacy across all engineering teams at the company, ensuing as timely and smooth adoption as was possible.
- Developed a reporting service that provided detailed information on the state of S2S adoption in all Twitter services, increasing the ability to adopt and giving leadership insight into progress.

Privacy and Data Protection (PDP):
- Technical Lead for the PDP steering committee, providing technical context to non-technical product and legal team members.
- Point of contact for 100+ engineering teams regarding PDP issues, enabling them to find answers quickly to questions and concerns.
- Author of 8 briefs to leadership detailing the technical challenges of PDP to inform how implementation could be streamlined.

Crowdsource Misinformation Detection (Birdwatch):
- Created backend systems which powered pseudo-anonymous aliases for Birdwatch participants to reduce harassment.
- Implemented real-time stream processing for note ratings and impressions, to replace a batch system which lagged by days.
- Reduced the unit tests runtime for every service for Birdwatch from ~50 minutes to ~5 minutes, increasing developer velocity.

Other accomplishments during the time at Twitter:
- Created a new 6-person engineering team (Platform Security) within Infosec dedicated solely to building security systems, including staffing and training.
- Core contributor to the End to End DM Encryption conversation up to the CEO-level.
- Project manager for the Twitter whitehat vulnerability reporting program and front-line responder to discovered security incidents.
- Conducted 50+ technical design and security reviews for teams across the company.
- Participated in 100+ hiring panels and promotion committees.

	● Mentored ~20 engineers (from intern to staff levels) on career growth, system design, project management, and conflict resolution. ● 4+ years on two cross-functional think-tank groups (TAG and Platforum) which guided the technical direction of the company.
Salesforce (2011 - 2013)	**Salesforce Build Automation**: ● Implemented a distributed artifact store for all build artifacts and build logs for every autobuild, replacing the legacy system and stopping multi-day outages for developers. ● Added digital signatures for artifacts which were used to verify which specific check-in, virtual machine, and physical machine related to the artifact, allowing releasing builds to be done with 100% trust. ● Built a dynamic allocator for VMWare and Openstack virtual machines, enabling auto-scaling of the build system, reducing wasted VMs and reducing outages during peak times.. ● Removed a direct dependency for an auto-build instance requiring a “hot” connection to the database server, ensuring a build run would always complete when the database needed to be offline or was experiencing an outage.
AWS (2009 - 2011)	**AWS Elastic Beanstalk:** ● Designed and developed a secured safety monitoring system to self-heal environments and notify customers when they accidentally made their Beanstalk environments unusable via unsafe config changes. ● Implemented how Beanstalk environment deployments would be coordinated to other AWS Services (EC2, ELB, Route53, etc.) in a manner which would meet customer expectations. ● Trained new hires on Amazon's build and deploy systems in their first two weeks, improving overall team efficiency and ability to meet aggressive schedules. ● Documented all deployment and operational procedures to ensure on-call engineers could quickly address issues without impacting customer experience. ● Built a reporting engine used to determine environment launch times and root cause launch failures, enabling the team to build a better product in a data-driven way. ● Replaced synchronous operations with asynchronous workflows on SQS to prevent bottlenecks from directly impacting customer experience of using Beanstalk.
Amazon.com (2006 - 2009)	**Cryptographic Key Distribution System (Odin):** ● Gathered and documented requirements for how cryptographic keys and credentials were being used across major stakeholders across the company. ● As technical lead, documented the ~50 page design, wrote a threat model, and engaged infosec for review to ensure correctness. ● Educated team members without a crypto background on the basics of security: safe key handling, symmetric and asymmetric ciphers, digital signatures. ● Developed the secure host daemon for use on all Amazon hosts in every datacenter by other Amazon systems. ● Developed the distribution nodes, used for distributing materials to the host daemons in a low-latency (1s tp99) high-availability (five nines) manner.

	● Delivered a company-wide Principal's talk to Amazon developers overviewing design, implementation, and best practices of system use. **Company-wide Secure Cloud Storage Standard (Knox):** ● Gathered and formally documented requirements for teams desiring to store sensitive data on S3. ● Implemented the company-wide standard for teams with Java services and supported its use solo by other teams. ● Developed a proxy service that allowed non-Java customers to securely store and access data in the same way. ● Designed a work-around S3's (at the time) 5GB size limitation, which hampered teams with large data requirements. **Amazon.com Co-brand Visa Card team:** ● Integrated JBPM libraries into Amazon systems for managing credit card application workflows to take onboarding external partners from months to days of engineer work. ● Replaced request/response bottlenecks with asynchronous process workflows, resulting in a customer experience on the order ~10s of ms, down from 100+s. ● Promoted and coached on agile development within the team, specifically in the form of Scrum and Test-Driven Development, to speed up development cycle. Other accomplishments during the time at Amazon: ● Authored, managed, and executed the internal AWS Education Course for 2 years, educating hundreds of Amazon engineers to the early systems of AWS (S3, EC2, SQS, etc.) ● Actively educated engineers on scaling Java services, concurrency issues, JVM GC tuning, and code testing and safety. ● Mentored ~5 engineers on software development practices, conflict resolution, and career growth.
Rho, Inc. (2004 - 2006)	**Rho Project Tracker:** ● Gathered and documented requirements from all levels of the company, from company leadership to line statisticians. ● Designed and developed all Java Servlets, Taglibs, JSPs, and Javascript to power the Project Tracker. ● Maintained and optimized the SQL database, ensuring no bottlenecks experienced by users. Other work at Rho included: ● Maintained Rho's single sign-on implementation, including fixing several security flaws in its initial design. ● Developed RhoReviews, presented at the 2005 Society of Clinical Trials and an early cited example of AJAX-based web interfaces for clinical trials.
UNC School of Medicine (2003 - 2004)	**UNC School of Medicine Online Testing System:** ● Designed and implemented an online testing system to replace the traditional overhead projector system, used to improve the testing experience of 300+ first and second year medical students.
Hiddenmind	**Hiddenmind Active Mobility Server:**

(2000 - 2002)	• Created the license key verification libraries and tools used by sales to generate license keys. • Developed a Java language parser for a 10x improvement in product build time.

Skills

Expert	Distributed Systems, Information Security (Cryptography, Threat Modeling) Privacy (GDPR, General), ProgrammingLanguages (Java, Scala, Python) Databases (General SQL, MySQL, Postgres), Source Control (Git)
Adept	Cloud Computing (AWS), Real-Time Stream Processing (Kafka), Agile (Scrum) Programming Languages (Go), ORMs (Hibernate, SQLAlchemy), Linux (RHEL/Centos, Debian)
Basic	Programming Languages (Rust, Javascript/Typescript ,C/C++) Deployment Infrastructure (Docker, Kubernetes,Terraform) Machine Learning (Scalding)

Education

North Carolina State University, BS Computer Science (2000)