

BERGLER TECHNOLOGY RADAR 2021

Techniques

ADOPT

1. Threat modeling

See our [blog post](#) about this topic.

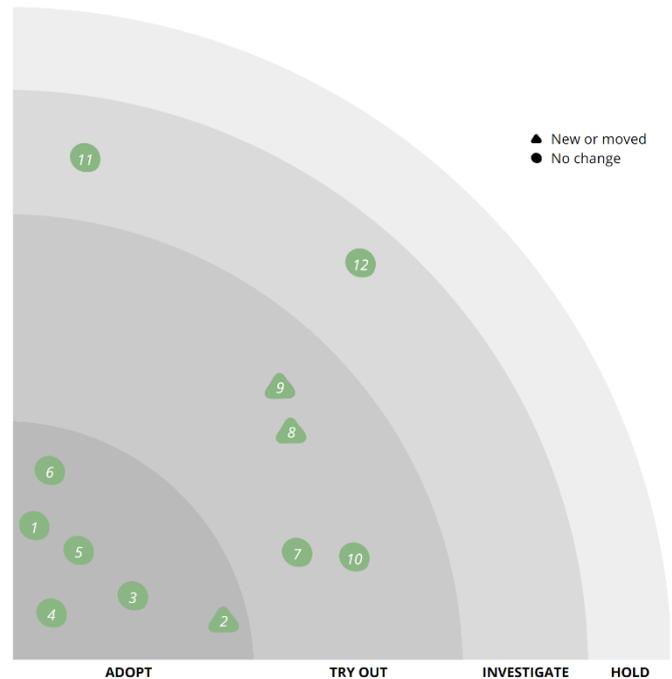
Threat modeling is a process by which potential threats, such as structural vulnerabilities or the absence of appropriate safeguards, can be identified, enumerated, and mitigations can be prioritized. The purpose of threat modeling is to provide defenders with a systematic analysis of what controls or defenses need to be

included, given the nature of the system, the probable attacker's profile, the most likely attack vectors, and the assets most desired by an attacker. Threat modeling answers questions like "Where am I most vulnerable to attack?", "What are the most relevant threats?", and "What do I need to do to safeguard against these threats?". Conceptually, most people incorporate some form of threat modeling in their daily life and don't even realize it. Commuters use threat modeling to consider what might go wrong during the morning drive to work and to take preemptive action to avoid possible accidents. Children engage in threat modeling when determining the best path toward an intended goal while avoiding the playground bully. In a more formal sense, threat modeling has been used to prioritize military defensive preparations since antiquity.

2. Future proof software development

See our [video](#) about this topic.

The world of software development has always been a place of rapid change. However the last decade the rate of change has drastically accelerated. On the one hand Agile processes have led us to ever increasing our rate of releasing. On the other hand developing multi-platform has opened a world of additional tools and practices that developers need to get acquainted with. How do we handle these circumstances? Those



questions is what future proof software development attempts to address.

3. **SOLID**

In object-oriented computer programming, SOLID is a mnemonic acronym for five design principles intended to make software designs more understandable, flexible, and maintainable. The principles are a subset of many principles promoted by American software engineer and instructor Robert C. Martin.

4. **Shift left**

See our [blog post](#) about test driven development.

Shift-left testing is an approach to software testing and system testing in which testing is performed earlier in the lifecycle (i.e. moved left on the project timeline). It is the first half of the maxim "Test early and often." It was coined by Larry Smith in 2001.

5. **Clean code**

See our [blog post](#) related to this topic.

Even bad code can function. But if code isn't clean, it can bring a development organization to its knees. Every year, countless hours and significant resources are lost because of poorly written code. But it doesn't have to be that way. Clean code focusses on principles, patterns, and practices of writing code that is easy to understand and maintain.

6. **Domain driven design**

See our [blog post](#) related to this topic.

Domain-Driven Design is an approach to software development that centers the development on programming a domain model that closely matches the processes and rules of a business domain. The name comes from a 2003 book by Eric Evans that describes the approach through a catalog of patterns. Since then a community of practitioners have further developed the ideas, spawning various other books and training courses. The approach is particularly suited to complex domains, where a lot of often-messy logic needs to be organized.

TRY OUT

7. **Eventstorming**

See our [video](#) related to this topic.

Event storming is a workshop-based method to quickly find out what is happening in the domain of a software program. Compared to other methods it is extremely lightweight and intentionally requires no support by a computer. The result is expressed in sticky

notes on a wide wall. The business process is "stormed out" as a series of domain events which are denoted as orange stickies.

8. Cloud native

See our [video](#) related to this topic.

Cloud-native technologies empower organizations to build and run scalable applications in modern, dynamic environments such as public, private, and hybrid clouds.

Containers, service meshes, microservices, immutable infrastructure, and declarative APIs exemplify this approach. These techniques enable loosely coupled systems that are resilient, manageable, and observable. Combined with robust automation, they allow engineers to make high-impact changes frequently and predictably with minimal toil.

9. Cloud agnostic

From our perspective it is wise to assess if you want a cloud agnostic solution. Vendor locking is not necessarily a problem but assessing up front is advised.

10. Infrastructure as code

Infrastructure as code (IaC) is the process of managing and provisioning computer data centers through machine-readable definition files, rather than physical hardware configuration or interactive configuration tools. The IT infrastructure managed by this process comprises both physical equipment, such as bare-metal servers, as well as virtual machines, and associated configuration resources. The definitions may be in a version control system. It can use either scripts or declarative definitions, rather than manual processes, but the term is more often used to promote declarative approaches. It is advised to dive into this technology, it has been added to trial because of the impact on the organization not because it is not mature.

INVESTIGATE

11. DevSecOps

See our [webinar](#) related to this topic.

DevSecOps is an augmentation of DevOps to allow for security practices to be integrated into the DevOps approach. The traditional centralised security team model must adopt a federated model allowing each delivery team the ability to factor in the correct security controls into their DevOps practices. It is advised to dive into this technology, it has been added to trial because of the impact on the organization not because it is not mature.

12. NoSQL

We see NoSQL as a valuable options which need to be assessed when redesign an

existing or create a new application. Re-evaluating the use of a standard SQL solution cloud be very useful both cost and development wise.

Platforms

ADOPT

13. Azure DevOps

See our [blog post](#) or [video](#) related to this topic

Azure DevOps Server is a Microsoft product that provides version control, reporting, requirements management, project management, automated builds, testing and release management capabilities. It covers the entire application lifecycle, and enables DevOps capabilities.



TRY OUT

14. Kubernetes

See our [video](#) related to this topic.

Kubernetes is an open-source container-orchestration system for automating computer application deployment, scaling, and management. It was originally designed by Google and is now maintained by the Cloud Native Computing Foundation.

15. Docker

See our [video](#) related to this topic.

Docker is a set of platform as a service products that use OS-level virtualization to deliver software in packages called containers. Containers are isolated from one another and bundle their own software, libraries and configuration files; they can communicate with each other through well-defined channels.

16. AKS

See our [video](#) related to this topic.

AKS allows you to quickly deploy a production ready Kubernetes cluster in Azure. It

enables you to use the power of Kubernetes from the Azure cloudplatform and combines the power of Azure services with Kubernetes.

INVESTIGATE

17. Azure Functions

The general direction of application development is cutting up monoliths in smaller part. Azure functions could be the next iteration of this concept.

18. IoT

Internet of things is a hot topic for many years now, we see the maturity level has gone up significantly.

19. Progressive WebApps

See our [blog post](#) or [video](#) related to this topic.

A progressive web application is a type of application software delivered through the web, built using common web technologies including HTML, CSS and JavaScript. It is intended to work on any platform that uses a standards-compliant browser, including both desktop and mobile devices.

Tools

ADAPT

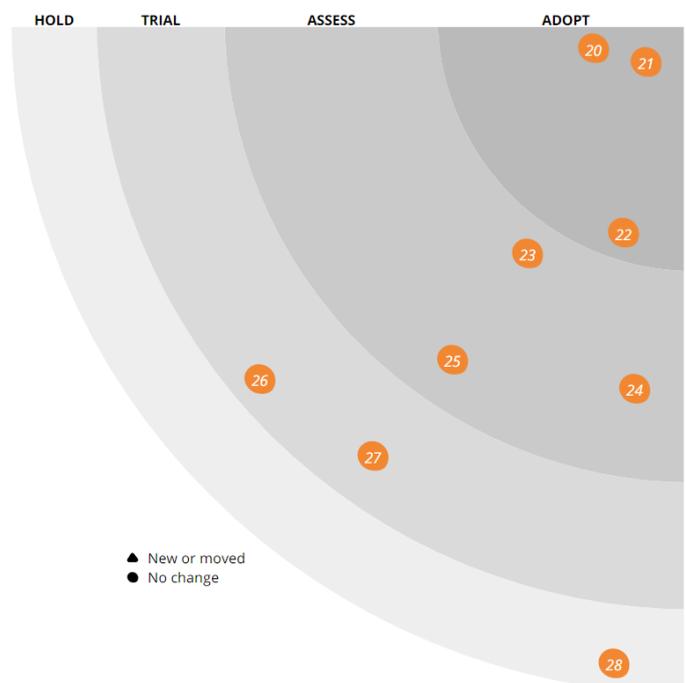
20. Git

See our [video](#) or this [video](#) related to this topic.

Git is a distributed version-control system for tracking changes in any set of files, originally designed for coordinating work among programmers cooperating on source code during software development. Its goals include speed, data integrity, and support for distributed, non-linear workflows.

21. OpenID Connect

See our [video](#) related to this topic.



OpenID Connect is a simple identity layer on top of the OAuth 2.0 protocol, which allows computing clients to verify the identity of an end-user based on the authentication performed by an authorization server, as well as to obtain basic profile information about the end-user in an interoperable and REST-like manner. In technical terms, OpenID Connect specifies a RESTful HTTP API, using JSON as a data format. OpenID Connect allows a range of kinds of clients, including Web-based, mobile, and JavaScript clients, to request and receive information about authenticated sessions and end-users. The specification suite is extensible, supporting optional features such as encryption of identity data, discovery of OpenID Providers, and session management.

22. Automated integrity checks

Whitesource, sonatype Nexus lifecycle, prisma cloud, sonarqube

These tools are lightweight open source security and management solutions, integrated within Microsoft's Azure DevOps Services & Azure DevOps Server (formerly TFS) products. It enables you to detect and remedy vulnerable open source components.

TRY OUT

23. Dependabot

Security and feature wise it could be wise to update you dependencies regularly. Dependabot can do this automated and generate pull requests.

24. Helm

Helm package manager for kubernetes, making adoption and development of kubernetes apps easier for those with little container experience.

25. SignalR

SignalR is around for some time now, but currently mature enough. It can be used for "real-time" communication with the browser.

INVESTIGATE

26. Cosmos DB

Azure Cosmos DB is Microsoft's fast NoSQL database with open APIs for any scale. The service is designed to allow customers to elastically (and independently) scale throughput and storage across any number of geographical regions. Azure Cosmos DB is the first globally distributed database service in the market today to offer comprehensive service level agreements encompassing throughput, latency, availability, and consistency.

27. AKKA.net

Akka.NET offers a way to build your application using actors. The framework solves a

typical concurrency problem. In a regular application if you want concurrency you're have to use things like locks and signals.

HOLD

28. DAPR

Dapr provides many capabilities in its actor runtime including concurrency, state, life-cycle management for actor activation/deactivation and timers and reminders to wake-up actors. Developers can easily implement the actor pattern for microservices running on top of Dapr.

Languages-and-frameworks

ADAPT

29. C# 9

C# 9 is the newest version of the C# language offering support of some new useful features like records. C# 9 is officially only supported on .NET 5 and future versions although some compatibility with older versions is possible.

30. .NET 5

.NET 5 is the newest version of the .NET framework where .NET core and .NET fullframe work merge together to one framework.

31. xUnit

xUnit.net is a free and open-source unit testing tool for the .NET Framework, written by the original author of NUnit. It is licensed under Apache License 2.0 and the source code is available on GitHub. xUnit.net works with Xamarin, ReSharper, CodeRush, and TestDriven.NET.

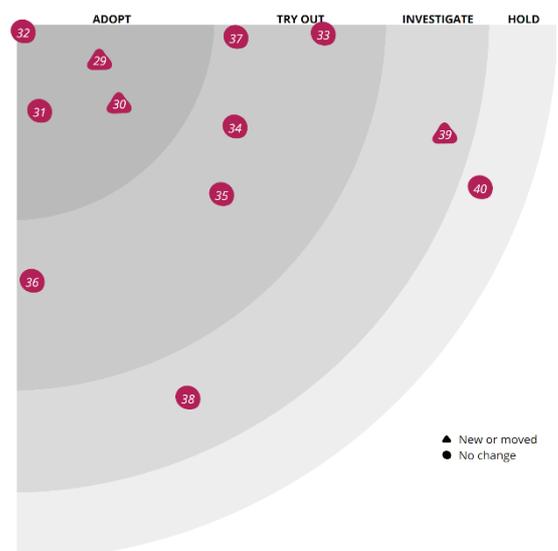
32. Specflow

SpecFlow helps teams bind automation to feature files and share the resulting examples as Living Documentation across the team and stakeholders.

TRY OUT

33. VUE.js

VUE.js surprisingly easy to start with, offers plenty of essential features out of the box,



and offers very good performance.

34. **React.js**

React is an open-source, front end, JavaScript library for building user interfaces or UI components. It is maintained by Facebook and a community of individual developers and companies. React can be used as a base in the development of single-page or mobile applications.

35. **GraphQL**

GraphQL provides an approach to developing web APIs and has been compared and contrasted with REST and other web service architectures. It allows clients to define the structure of the data required, and the same structure of the data is returned from the server, therefore preventing excessively large amounts of data from being returned, but this has implications for how effective web caching of query results can be. The flexibility and richness of the query language also adds complexity that may not be worthwhile for simple APIs. It consists of a type system, query language and execution semantics, static validation, and type introspection.

36. **Python**

See our [blog post](#) related to this topic

Python is an interpreted, high-level and general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant whitespace.

37. **GO**

To learn multiple programming language is a good idea. GO could be a good contender. Go has the same performance as C, but it is much easier to maintain than Java. Without the need for a virtual machine, Go boasts easier maintenance and no warming up period.

INVESTIGATE

38. **Blazor**

Blazor is a .NET framework for creating a single-page application that allows you to create your application using C#, Razor and HTML. The client-side of Blazor uses WebAssembly, which has been shipped with major browser engines, such as Internet Explorer, Chrome, Safari and Firefox. WebAssembly is an alternative to JavaScript in writing client-side code. And it can be converted using many different programming languages. In Blazor's instance, it is converted using C#.

39. **React Native**

React Native is an open-source mobile application framework created by Facebook, Inc.

It is used to develop applications for Android, Android TV, iOS, macOS, tvOS, Web, Windows and UWP by enabling developers to use React's framework along with native platform capabilities.

HOLD

40. **Angular**

See our [video](#) related to this topic

Angular is a TypeScript-based open-source web application framework led by the Angular Team at Google and by a community of individuals and corporations. Angular is a complete rewrite from the same team that built AngularJS. Although Angular is a powerful and mature framework it is more bulky than React and Vue. Also we would like to prevent front-ends from becoming monolithic applications.