8 DOS & DON’TS OF SECURITY ARCHITECTURE

We asked three experts about what makes this tech field tick.

November 22, 2019

Written by Mae Rice

Cybersecurity spending had a record year in 2018, exceeding $100 billion. But that doesn’t mean everyone is optimally secured.

Besides the fact that security is expensive, there’s also a workflow component. What has typically been a late-stage consideration in the tech production process — as per industry parlance, “bolted on” at the end — is moving to the forefront as products and features are conceptualized and designed with security firmly in mind. Instead of bolted on, it's built in. And there’s a relatively new term for this process that’s catching on: security architecture.

Some would call it that, anyway; the definition remains fairly fluid. But while security architecture also can be interpreted broadly — as, say, all the resources and protocols that allow engineers to build safe new products, or the way in which a given security system is structured — it’s still closely tied to built in security. As a last-minute add-on, security often ends up hamstrung by budget constraints and technical challenges.

“When you're bolting on, you may be handcuffed to certain things,” Munya Kanaventi, senior director of security engineering at Everbridge, told Built In.

One example: in AWS, Amazon’s cloud service, S3 buckets — a common type of data container — can be easily encrypted during setup by checking a box. Once data is uploaded, however, encrypting isn’t so simple.

“You actually have to export the data out of the bucket, re-encrypt, set the encryption tag and then add the data back,” Kanaventi said. “If I've moved 10 terabytes of data to the [bucket] and then you tell me, ‘Oh, you have to move that back so we can check this button and start encrypting and then move it back,’ I've pretty much just tripled the cost of transport.”

Security architecture minimizes such needless expenses. **Here are some dos and don’ts of implementing it, according to three experts (including Kanaventi).**

DO BE PRAGMATIC.

PAT CABLE

Director of Platform Security at Threat Stack

Often, folks focus on making the most secure thing ever, when the reality is that you're never 100% secure. And there’s a trade off. You could spend months putting together something that is the most secure, but if you go out of business, it doesn't help you. So you need to know where to draw that line. I find that a lot of times, security folks will push for the most secure thing when 80% of that would be accomplishable in a shorter period of time and you can work towards that extra 20% over the longer term.

MUNYA KANAVENTI

Senior Director of Security Engineering at Everbridge

It's important to balance maintaining security or managing risks with usability, and you have to balance all that with the appetite of the business. You have to understand what the business needs and what you need to do in order to achieve it.

PROF. STUART MADNICK

John Norris Maguire Professor of Information Technology and Professor of Engineering Systems at the Massachusetts Institute of Technology

Sometimes people in security don't understand how [users] are going to behave. Unfortunately, a lot of things that may be security ideas at the very abstract level cause real problems, if you look about how they actually play out.

For example, at many organizations you must change your password every 90 days. Well, one of my colleagues at MIT, Professor Catherine Tucker, wrote about this one organization she studied where you had to change your password every 30 days. And of course, the passwords have to be complex passwords.

It turned out when they put that policy in, the net effect was to make the organization 30 percent less secure than it was before. Now how did that happen? Well, if you've got to change your password every 30 days, what do you do? You make your password XYZblahblahblah1, then

XYZblahblahblah2, then XYZblahblahblah3. There's no other way to keep track of it. So the quality of the passwords actually became more breakable.

DO THREAT MODEL.

KANAVENTI: Our security architecture comes with a concept of what's called threat modeling. It's the concept of initially understanding the threats you're working against — the ability to say, what is the threat? When does it happen? How does it happen? Threat modeling allows you to appropriately select controls. An example of that is people that live in neighborhoods where there's statistically a lot of shootings. Maybe most people don't go out after dark. They could wear a bulletproof vest to water the lawn in the morning, and get a false sense of meeting the objective, because then if they do get shot they don't die. But that's not really the appropriate control.

One good way to model the threats is with the STRIDE model. STRIDE is an acronym for the five basic threat categories:

- **S**poofing, or being able to look like someone else — look like a different user, look like you're coming from a different IP address, those things.
- **T**ampering, which is how easy it is for people to tamper with data.
- **R**epudiation — how easily users can take an action in your system, and pretend they didn't.
- **I**nformation disclosure, which is basically the opposite of privacy.
- **D**enial of service — how easily can a malicious actor shut down the service you're building?
- **E**levation of privilege is the last one. If a user elevates their privilege in my system, I want to make sure that you are authorized to do so.

MADNICK: The model I typically rely on is often referred to as the National Institute of Standards and Technology framework. And it involves five elements: identify [vulnerabilities in your system], protect [those vulnerabilities], detect [attacks], respond [to attacks] and recover [after attacks]. It's not exactly about architecture — architecture is how you organize your system, and the NIST model is about the things your system needs to do. But of course, there is a relationship between them.

Detection, especially, is key. If someone were to break in, how quickly can you realize it? Cyber attacks, particularly in the US and Western Europe typically have been going on for over 200 days before they're detected. And in the Asia Pacific region, I've heard numbers closer to 400 days. You want to catch attacks more quickly than that.

DON'T INTERCONNECT ALL YOUR SYSTEMS.

MADNICK: One important thing is isolating or segmenting parts of the system, so that whether an insider or an outsider breaks into the system or misuses the system, the damage is limited. They only can damage the section they're in. It's often referred to as "zero trust" — you don't give any user the power to cause problems that permeate throughout the organization.

Unfortunately, that's not the way many systems have historically been developed, because often you don't know ahead of time exactly what you want to do. Making it possible for anybody to access anything gives you maximum flexibility. That sounds great if flexibility is your most important thing, but that flexibility comes with a big liability.

I don't think this ever happened as an actual cyber attack, but it’s been demonstrated that on a number of airplanes, tapping into the entertainment system at your seat would allow you to control the engines on the plane. It was all designed as one big communication system. Whether it's the pilot talking to the engine or you talking to the TV set, it's all over the same phone line, if you will. Segmentation just was not a priority. But should the entertainment system be connected to the engine control system?

DO KNOW YOUR STUFF.

KANAVENTI: Security architect is one of the only roles in a technical room that has to know enough about everything to garner everyone's respect. What I mean by that is, when you go talk to the networking guys, they know networking very well; they've been doing it their entire career. Then you go talk to the database guys and the same applies. You have to talk to all of these different teams and explain to them why they potentially need to do things differently, and earn their respect. It requires a high level of breadth and depth. You probably have to have five to eight years of experience in the tech sector, minimum.

DON’T COPY ANOTHER COMPANY’S SECURITY ARCHITECTURE.

CABLE: Every organization has some sort of security knowledge, but each company is going to have a different structure for that depending on what they do. I always joke that Twitter for pets is going to have a very different security model than a company like Threat Stack, where we provide a security solution to other companies. For us, the security and privacy needs really center around ensuring that our customer's data is secure and ensuring that our engineers don't accidentally put that at risk.

DO CONSIDER YOUR AFFILIATES' SECURITY.

MADNICK: Remember that break-in at Target in 2014? The way the break-in took place was not directly through Target. Target had an air-conditioning maintenance company they worked with, and in order to schedule things and so on, the company had access to the Target corporate computer. The attacker broke into the air conditioning maintenance company — most likely a relatively modest size organization that didn't have a particularly strong security and broke into their system. Then they were able to migrate over to the Target system.

We talk a lot about this in our research — it’s what we call a third-party vulnerability. Once upon a time these things were simpler, but in the interconnected internet age, system A and system B — which used to be separate systems — talk to each other, and the systems of company X and company Y talk to each other. So where is the boundary line between your system and your vendors’ systems? It’s really important to think about.

DO THINK LIKE A HACKER (A BIT).

CABLE: It's important to be able to sit back and have that mindset of, "Hey, this is what happens when somebody tries to break this thing." It's not necessarily going in with a mindset of, "Oh, I want to hack into this," though that does help. It's just understanding the parameters of what you're building, and how it could go wrong.

KANAVENTI: I don't think actual hackers are good security architects, necessarily, but I think validation is a part of the architecture. An example being, if your architecture says, "You should use a firewall here." Okay. Did they put in a firewall? Yes they did. Did they leave every port open on that firewall? Yes they did. So it's there, but it's not effective. You have to measure the effectiveness of the controls at the end, just to validate. That's where a hacker mindset could come in handy.

MADNICK: I think that it's very important to think like a hacker in security architecture. Most people tend to be optimists, and when they design systems, they design them to behave the way you intend them to behave. But if you look at most attacks, if you look at them in retrospect, almost all of them are somewhat amusing. Something bizarre was being done that no one ever thought of — like setting an alarm clock software to 25 o'clock.

DO COMMUNICATE AND COLLABORATE WITH OTHER DEPARTMENTS (ESPECIALLY ENGINEERING!).

CABLE: I think folks really focus too much on tools versus "Hey, what can we do to make sure that engineering and security get along well?" When you talk about traditional cybersecurity, where security comes at the end of the build process, there can be a lot of animosity and headbutting between engineers and security. Engineers are saying, "Oh, I didn't know this was a requirement, now we have to go back and change everything!"

That's not what you want. If you work on security architecture, you have to work very closely with engineering to make sure that when they want to do something new, they come to you for input at the beginning of the process. I have a friend in cybersecurity who had a $350 a month candy budget, and he says it was the best money they ever spent, because it meant that the engineers came over to talk. Those side conversations really helped build up trust that when engineers approached security with a new project, it wouldn't just turn into the department of "no." It was more the department of "yes, and."

MADNICK: When I talk about architecture, I don't think of it just in terms of the software architecture, but also the architecture of the organization. How is the organization structured? In most organizations, if you were to ask an arbitrary person, a receptionist or an accountant, "Who's responsible for cybersecurity?" they'd say, "Well, those guys in IT." But these are the people — the administrators, the accountants — who are leaving the doors open for the bad guys to get in. My view is that

everybody in an organization has a role to play in cybersecurity, and if they don't understand that then all of the organization’s vulnerabilities are out there to be taken advantage of.

Images via Shutterstock and interviewees. Responses have been condensed and edited.

Live your purpose. Work your passion.