The computer says no!

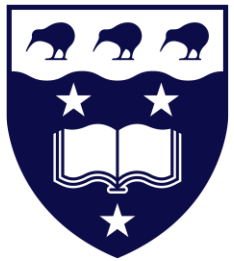
Security as an enabler of the business

Peter Jakowetz, PrivSec Consulting

OWASP Day 2025 - Auckland, NZ

Thank You to Our Sponsors and Hosts!

OWASP NEW ZEALAND
owasp.org.nz



Waipapa Taumata Rau University of Auckland

BASTION SECURITY GROUP

SECURE CODE WARRIOR

safeadvisory.

plexure

SECDIM

Without them, this conference couldn't happen.

Who am I?

- Current Principal consultant and Managing Director at PrivSec
- Ex-electrical engineer, SOC analyst, pentester, auditor, architect, security manager
- CISSP, CCSK, CCSP, PCIP, OSCP, CISA etc
- Have zero design skills ...
- I have a kid that says no a lot, so don't need it at work too

What this talk is about

- Security people like saying no
- Some examples of things that i've seen
- Constraints
- Understanding
- Tradeoffs
- What we can do

No is a two year olds favorite word

- And also security teams
- They are protective!
- They don't like change!
- But sometimes go for the easy answer
- How would you convince your child..?

The role of a security team

- Ensuring things are 'safe and secure'
- Might have a tendency to drift a wee bit out of bounds
- Make sure that we avoid threats to the CIA triad

The role of an individual contributor

- (Dev/ Architect/ Engineer etc)
- You're here to build
- To ship
- Gotta close those Jira tickets
- Trying to meet the functional business need

Security controls

- There are some great basics that we should have in place [read: OWASP Top 10]
- Patching
- Logging
- Strong access management
- Backups
- Strong encryption
- ...

What about when it's not obvious?

- Strong encryption is good!
- SSL scan says website == bad
- Restrict ciphers so only supports strong!
- But what about if the *most* important thing is that your app is available to a wider group of people

NEW ZEALAND / COVID-19

IT expert says My Covid Record app at risk of security breaches

10:56 am on 14 October 2021

Share this

When usability > security

- 'But we don't want MFA'
- Use case might be a really specific set of users: Elderly, minorities, CEOs, ?
- People might not have phones?
- People might not have *work* phones?
- Are there ways that we can work around this?
- Tokens, txt, email, one time passwords, device certs
- What is the information we're actually protecting?

Why do they say no?

- There's typically good intent!
- There might be some context in the background that *you* aren't aware of
- Pre-planning for that future use case you're not aware of
- You're trying to use technology the security team doesn't understand
- The singular security resource is *really* busy and hasn't quite got back to that design yet
- There was a bad experience last year with that technology at the org
- Something that sounds like that technology/ technique was in the news recently

'Don't use the word *Kiosk*'

- Back in 2012 there was a breach at MSD due to a poorly configured kiosk
- People *still* hurt when they hear the word 'kiosk' in Government
- Find a nicer way to talk about that non authenticated PC

Home / New Zealand

MSD shuts Winz kiosks after lax security exposed

APNZ

15 Oct, 2012 12:45 AM 3 mins to read

Resource constraints

- We're in a recession and teams are *tight* whether development or security
- There's going to be an ease in approving things that are 'easy' rather than 'different'
- And things that are 'different' may get immediate friction

Constrained by the process

- Some security teams are constrained by a process, or set of compliance activities
- Credit card processing == PCI
- Health data == HIPPA
- Difficulty often with a misunderstanding of the process/ standard
- Meeting organisational policies
- These sometimes give the opposite problem of actually getting lazy 'yes' rather than no
- ISO27k policy says - patch once a year, so we only patch once a year!

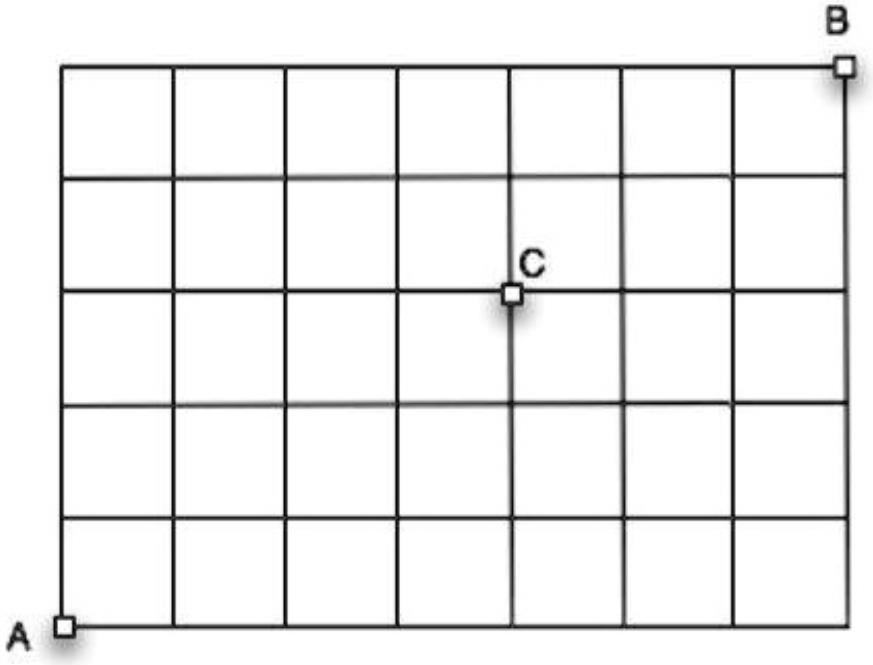
No understanding of the problem space

- How can you share the background of the solution?
- How can you share your understanding of the threat landscape effectively?
- Diagrams are great! But are even better when you can explain them!
- Try to avoid ambiguity
- Admit where you need help for input into controls - 'How have you seen that done before, and what would you recommend we do'.

Lots of paths from A to B

- There often isn't a clear cut answer

8. Consider the paths from A to B as described in the previous problem. How many different paths from A to B go through C?



So you need prod access

- ~~You want prod access so that you can have some more representative data~~
- Seems totally reasonable
- UAT is a dumpsterfire and has terrible quality of data
- But what's the *actual* problem?
- Can UAT be made closer to prod?
- Can you get a redacted extract to a reporting DB?
- Spin up another restricted environment for a few days?
- Get someone to generate some better test data
- Get a limited copy of prod data
- Rather than just giving prod access carte blanche

I want local admin

- EVERYONE want local admin EVERYWHERE
- But what are you actually trying to achieve?
- Is it just to install a single bit of software?
- You just want to get your job done, but the security team are worried that you then install a bunch of unapproved apps/ compromise your device/ ex-fill a bunch of data
- Are there other options? Is there some self-service tooling that can be used?

I want to use this new shiny toy/library/framework/security nightmare

- ~~There are so many great new tools available all the time!~~
- What is its differentiating feature from what was otherwise available?
- Be open to critique
- Be willing to show how *you* have confidence on it
- Does it have some type of reputation?
- Have you thought about the implications and how you could protect yourself from them?
- Is there a low-risk POC you can do to make security comfortable

Compensating controls

- Lots of compliance frameworks (think PCI) allow for the concept of compensating controls
- But make sure they're *actually* useful/ realistic/ address the treats that this thing is opening
- Could be logging, privileged access, a WAF, additional network controls, good config management

Bad example of compensating controls

- Have seen a lot of examples where people try and throw 'the kitchen sink' at problems when trying to explain why they're not doing something
- i.e. We can't patch this - we're going to put logging in place, without specifying logging on *what*.
- Lots of the time, when you actually look at the logs - they're just network logs etc.
- Can you restrict access to the unpatched thing?
- Can you air-gap it?
- Can you explicitly log who's hit the app?
- Are you aware of what the vulns are - and can you monitor if *those* are exploited on an IDS etc?

We need to step outside of this security guardrail

- Sometimes there might be guardrails put in place to do things well (i.e. ORMs)
- But it might not meet functionality needs
- So someone goes and mainlines string concatenation
- And squeely wins!
- How could that be avoided?

Someone else needs to be engaged

- What if there's a common capability that's not in place.
- You *should* have centralised logging in place a lot of the time
- But that common capability isn't available - so how do you work around that?
- There is lots of enterprise tooling/ common capabilities that can help out - password managers, code analysis tools, standard patterns, SIEMs, common infrastructure

POC it

- A really nice way to show a security person you know what you're talking about is to POC it
- Have been working with an architect on a SharePoint deployment
- Some interesting constraints in the org with DLP
- Being able to sit down and show in a POC environment the controls applied == a really easy approval process
- Demonstrate in a safe space rather than hypothetically talking about things

What happens when security says no

- Shadow IT proliferates
- 'Workarounds' are put in
- Bad design decisions are made
- Things become unusable
- An easier to explain, but worse solution can go into production
- Everyone has been to a website with innocuous data where the log in flow is TERRIBLE

Shadow IT

- Every org has a bad history of shadow IT
- Even if you don't think they do... they probably do and you just haven't found it yet!
- AI tools are a great example - 'the business hasn't got around to approving any tools yet' so everyone has just signed up to various tools and are using them without letting anyone know
- It's in a security teams best knowledge to have an open conversation with teams/ the business etc - so that this proliferates less

Can you provide secure alternatives

- From the perspective of the developer/ engineer/ architect
  - Is this the *only* way?
  - Are there alternatives?
- From the perspective of the security team -
  - Are you aware of different tools that could meet the same intent?
  - Is there something you're aware of from a future roadmap for the org etc?
- We all have different knowledge on what's being done more broadly at the organisation
- I.e. not comfortable with #radnewAItool, but ChatGPT is okay, because it has been vetted/ contract in place/ known commodity etc

Complexity can cause trouble

- Internal networks are frequently littered with good intentions (and remote code execution)
- Asset management is *hard*
- Sometimes the 'no' comes from a place of 'we don't believe there is capacity to maintain this solution
- The number of times we do a test on an environment, and the easy ways to get in are through some old unsupported app, or technology no longer maintained
- That was someone's 'yes' one day
- Can you remove complexity with your solution you're trying to push

Sometimes no is lazy

- It avoids understanding the problem the team is facing
- It's making it easier today (for probably a harder tomorrow)

We tell computers what to do, not the other way around

- How do you build in with automation easy ways for things to happen?
- Can you include checks in CI/CD pipelines for allowing faster deployment
- Can additional monitoring be put in place
- What other safeguards can be applied that give confidence without manual intervention?

How to say no effectively

- If it is a 'no' then why?
- What are the *real* risks?
- What are the *real* impacts?
- What are you trying to protect?
- Are there alternatives you're aware of?

The tradeoffs between security and useability

- We're typically always ending up for the same outcome
- There are multiple ways to get ot the outcome though
- What's going to be more sustainable, usable and meet security requirements
- For example PIM/ PAM can be a great solution to 'I need all the access' while giving appropriate oversight/ approvals/ logging etc depending on your specific scenario

Sometimes things need to change

- If you think it's the right decision, and that others are being unreasonable - you can push!
- Innovation does need to happen somewhere - how can you enable that?
- Justify your actions though - and try for some small wins.
- Are there alternatives/ a middle ground you can go to?

Summary - It can be hard!

- Be cognisent of the broader context (i.e. technical landscape)
- Be open to conversation - you'll need to collaborate
- Give yourself time to get it over the line
- Be prepared - design/ poc/ think through threat scenarios yourself
- Compensating controls? What can be done
- Strong business case - what does this allow the business to do
- Are there lessons I can show i've learnt from
- Be open to new/ different ways

Thanks

peter@privsec.nz
https://www.linkedin.com/in/peterjakowetz/

(Thanks Jim R for helping out with presso)