Insecurity in Information Technology

OWASP
The Open Web Application Security Project

Tanya Janca
Tanya.Janca@owasp.org
OWASP Ottawa Chapter Leader
OWASP DevSlop Project Leader
@SheHacksPurple

OWASP
The Open Web Application Security Project

The “I’m Qualified” Slide

Tanya Janca: Application security evangelist, web application penetration tester and vulnerability assessor, trainer, public speaker, ethical hacker, OWASP Ottawa chapter leader, OWASP DevSlop project leader, effective altruist, software developer since the late 90’s.

I’m extremely concerned about application security, and you should be too. Let me tell you more.

OWASP
The Open Web Application Security Project

Thesis:

People make stupid decisions when they feel insecure.

Conflict between security and development makes some people (employees) feel insecure.

When this happens, insecure software is often the result.

Changes are required to solve this problem.

OWASP
The Open Web Application Security Project

Warning:

This is not a talk about “feelings”.

This talk is about the bottom line, getting the job done, and making software secure.

However, it might get uncomfortable at times.

Try not to feel too defensive.

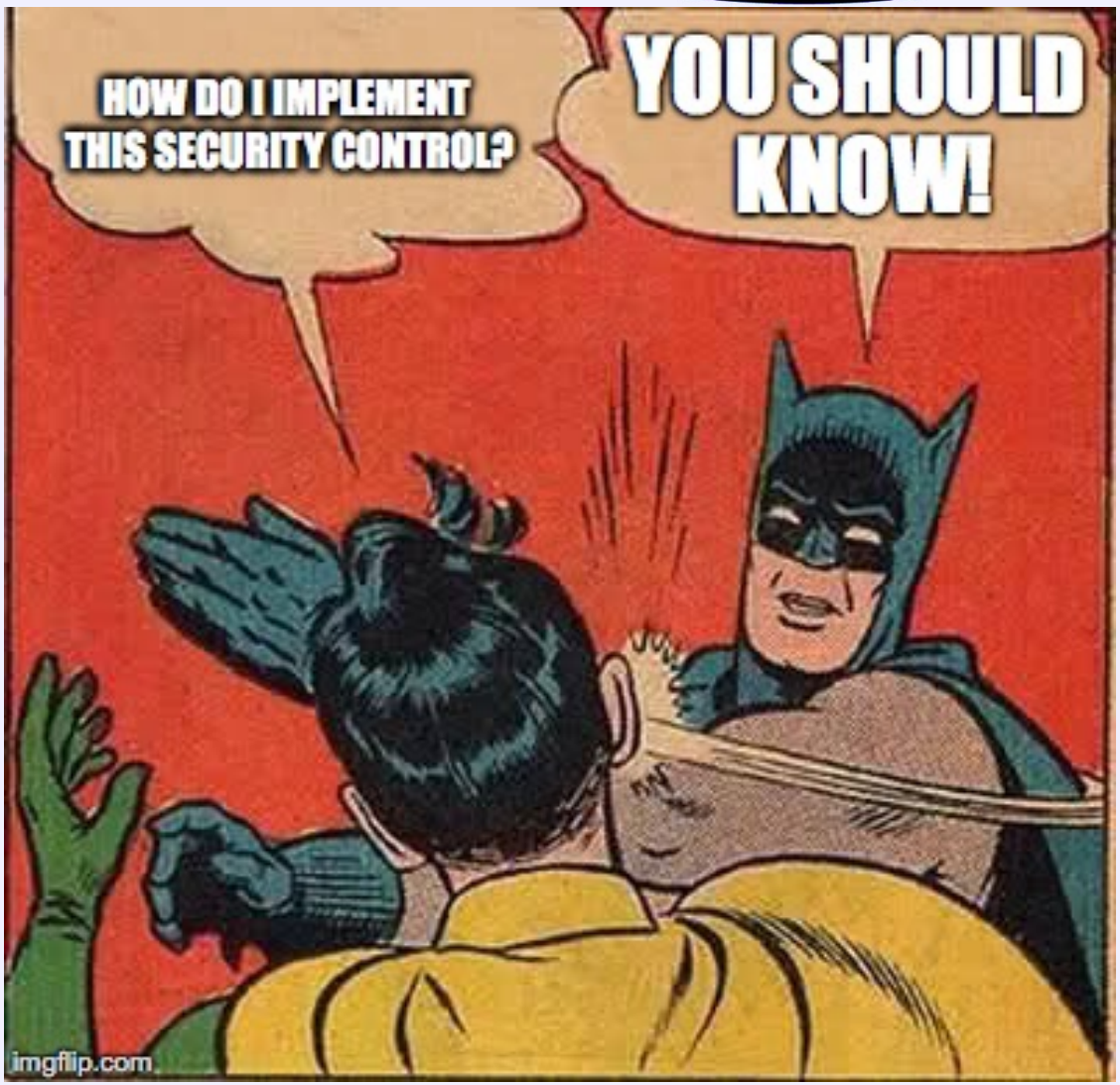
OWASP
The Open Web Application Security Project

The Fine Print:

Although I relay all examples in the first person, as though I was there, these are not all my personal examples. They are from multiple members of the InfoSec community.

Talk Outline:

- Problem(s)
- Solution(s)



OWASP
The Open Web Application Security Project

Problem:

The way many IT shops are run can create feelings of job insecurity in employees.

OWASP
The Open Web Application Security Project

Security testing is performed so late in the game that developers

1) Don’t fix anything
2) Resent security for presenting last minute challenges

Developers

OWASP
The Open Web Application Security Project

- Do security testing without the security team, making them feel unrequired.
- Don’t cooperate with the security team to enable security testing, claiming they have no time/implying security is low priority.
- Ignore security advice from security reports and do not implement any or most of the mitigations.
- Do not take/ask for advice from the security team.

Security

OWASP
The Open Web Application Security Project

The security team quotes policies at developers, or sends them an unvalidated 500 page report.

Security Team

OWASP
The Open Web Application Security Project

- Does not give **usable** security guidance to the developers when asked.
- Acts or is seen as a gate, slowing down the SDLC.
- Adds project requirements without explanation, “because security”.
- When revealing issues, they make developers feel incompetent.

All of this creates the feeling of insecurity about people's jobs and how to do them well.

This leads to:

- deviant behaviour,
- moral disengagement,
- reduced job involvement,
- risk taking behavior, and
- reduction of organizational citizenship behavior (positive workplace activity and involvement).

OWASP
The Open Web Application Security Project

All of this bad behavior leads to insecure software.

Is everyone on board with this? Questions?
Are we on the same page? More examples?

OWASP
The Open Web Application Security Project

OWASP
The Open Web Application Security Project

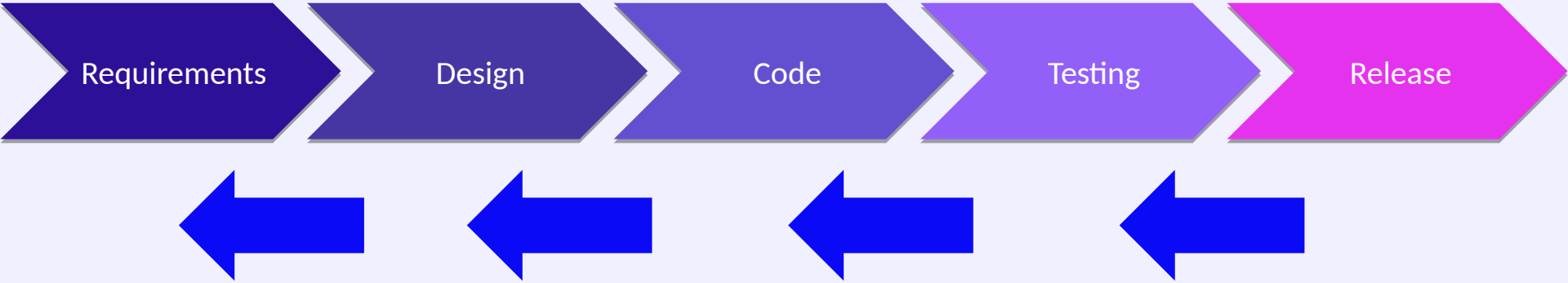
The Plan:

1. Support dev and sec team with processes, training, and resources so they can confidently get the job done.
2. Repair relationship.
3. Do not accept ‘bad’ behavior anymore.

OWASP
The Open Web Application Security Project

OWASP
The Open Web Application Security Project

Pushing Left, Like a boss!

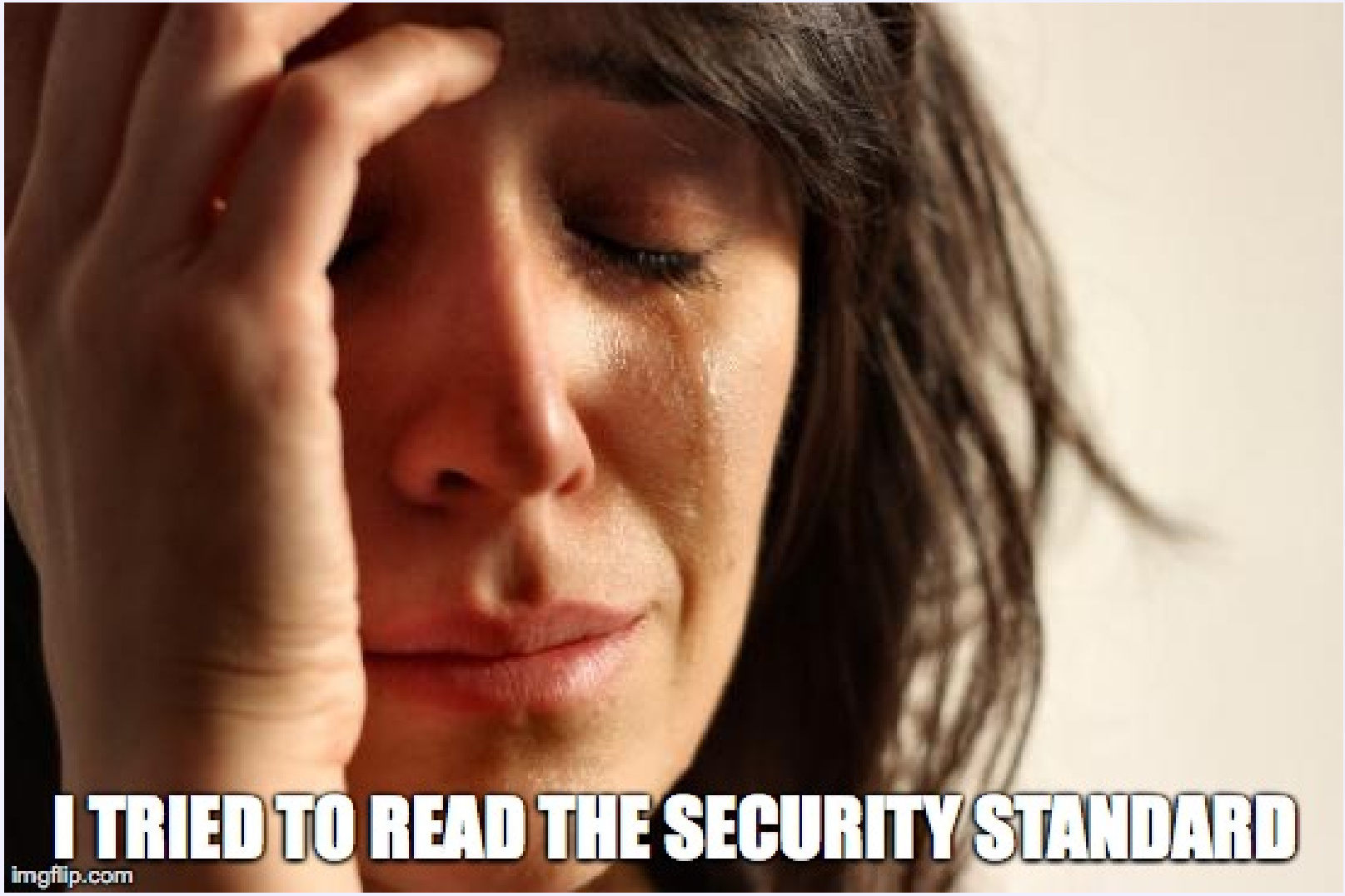


Start Security Earlier!

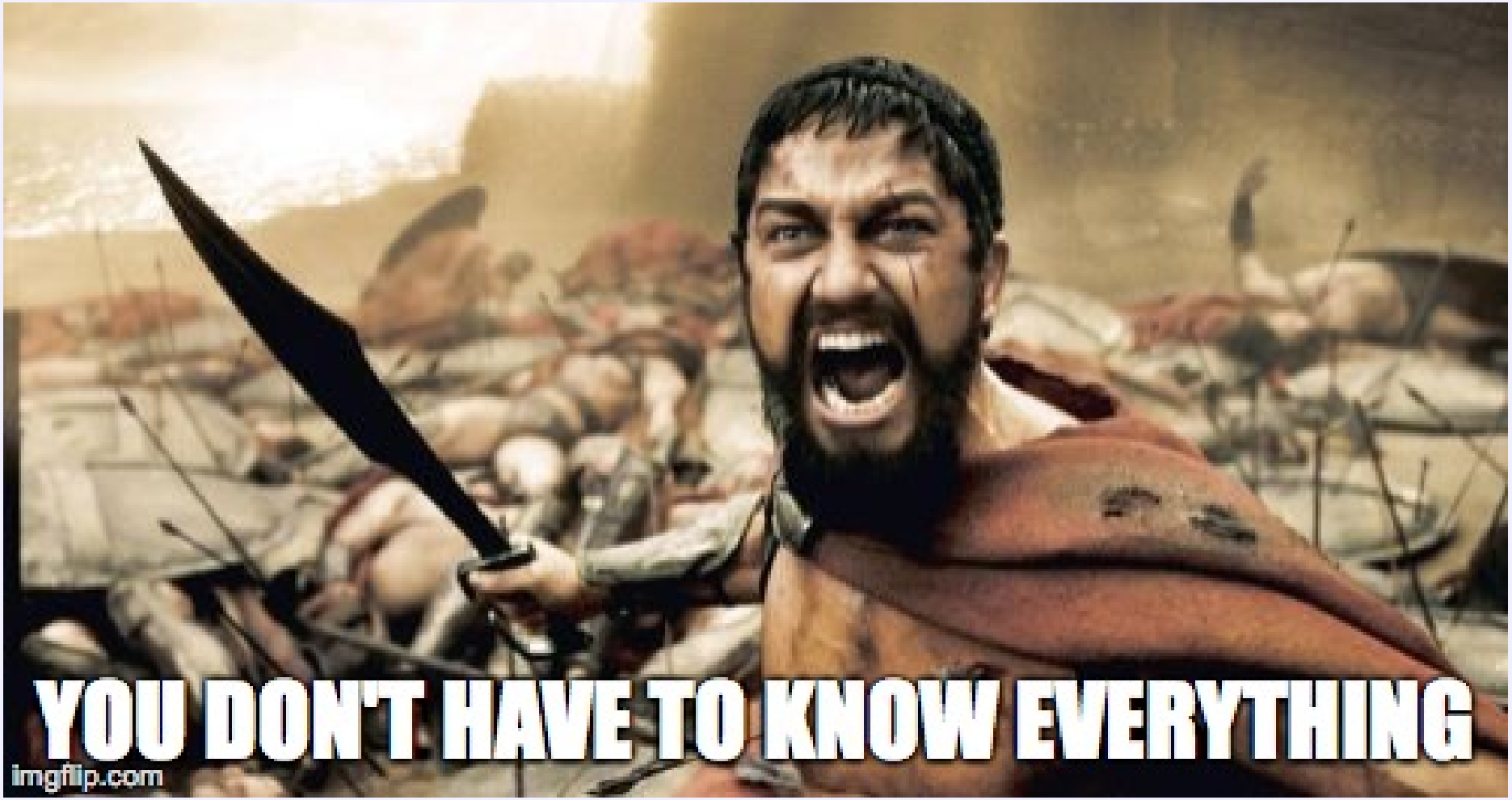
OWASP
The Open Web Application Security Project

OWASP
The Open Web Application Security Project

OWASP
The Open Web Application Security Project

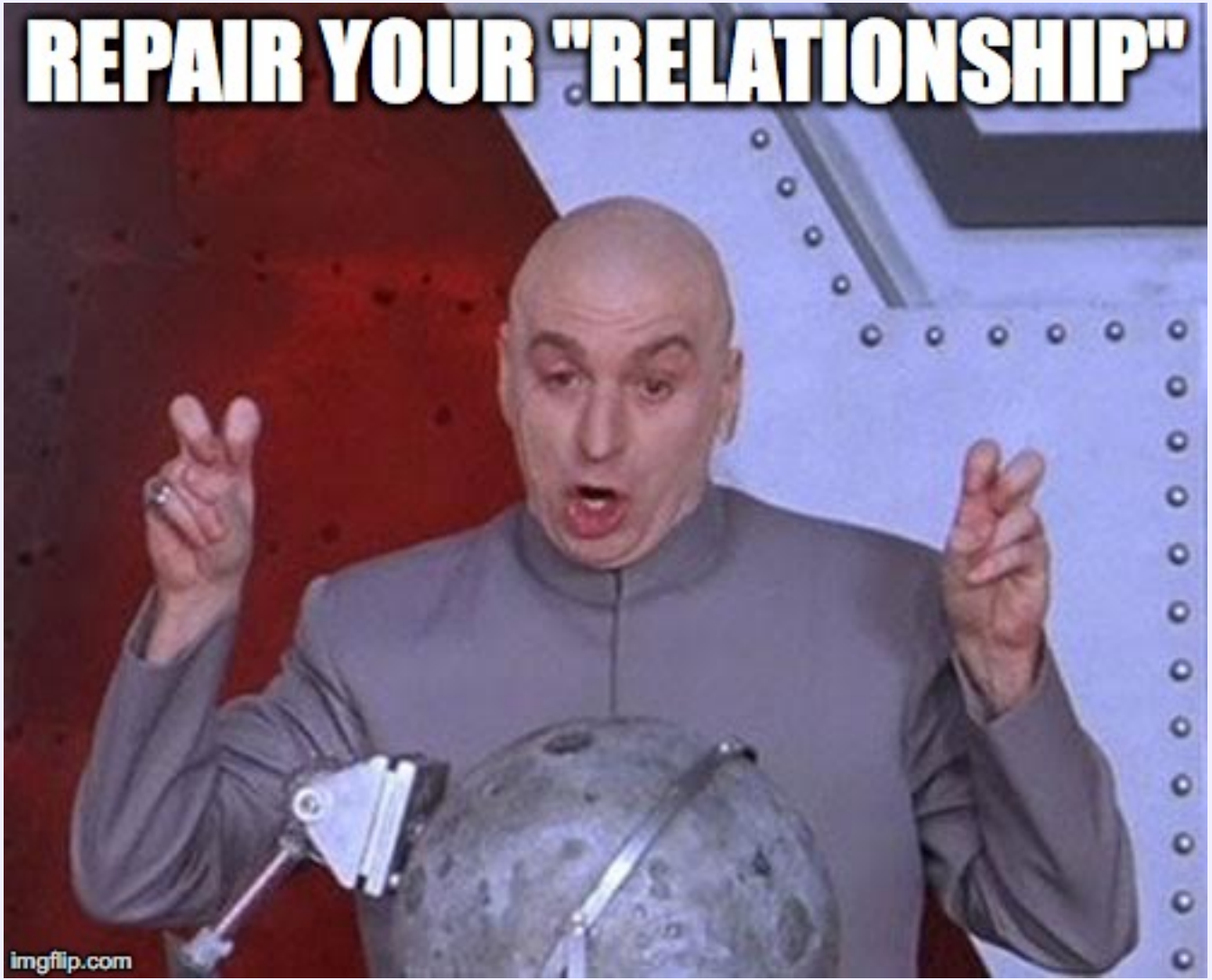

I TRIED TO READ THE SECURITY STANDARD
imgflip.com

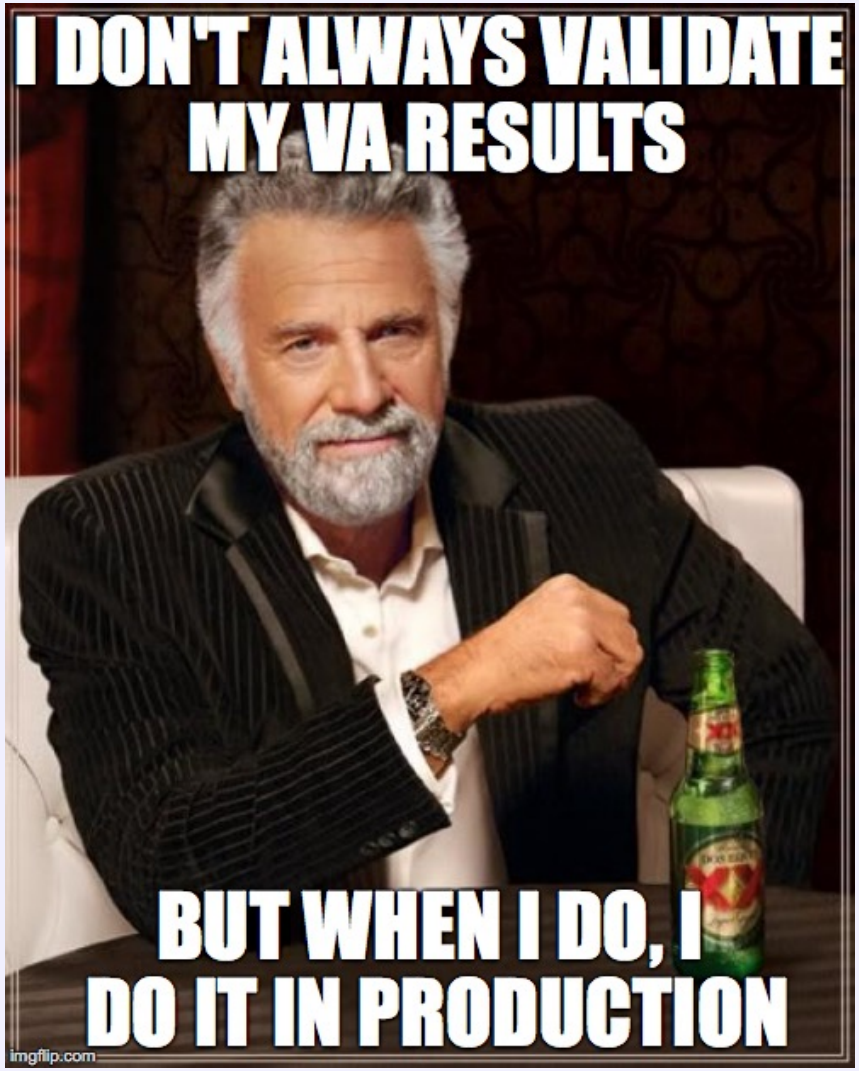

THEY'LL NEVER FIND THE
NEW SECURITY STANDARD
imgflip.com


YOU DON'T HAVE TO KNOW EVERYTHING
imgflip.com

OWASP
The Open Web Application Security Project

Break security testing into smaller pieces


REPAIR YOUR "RELATIONSHIP"
imgflip.com

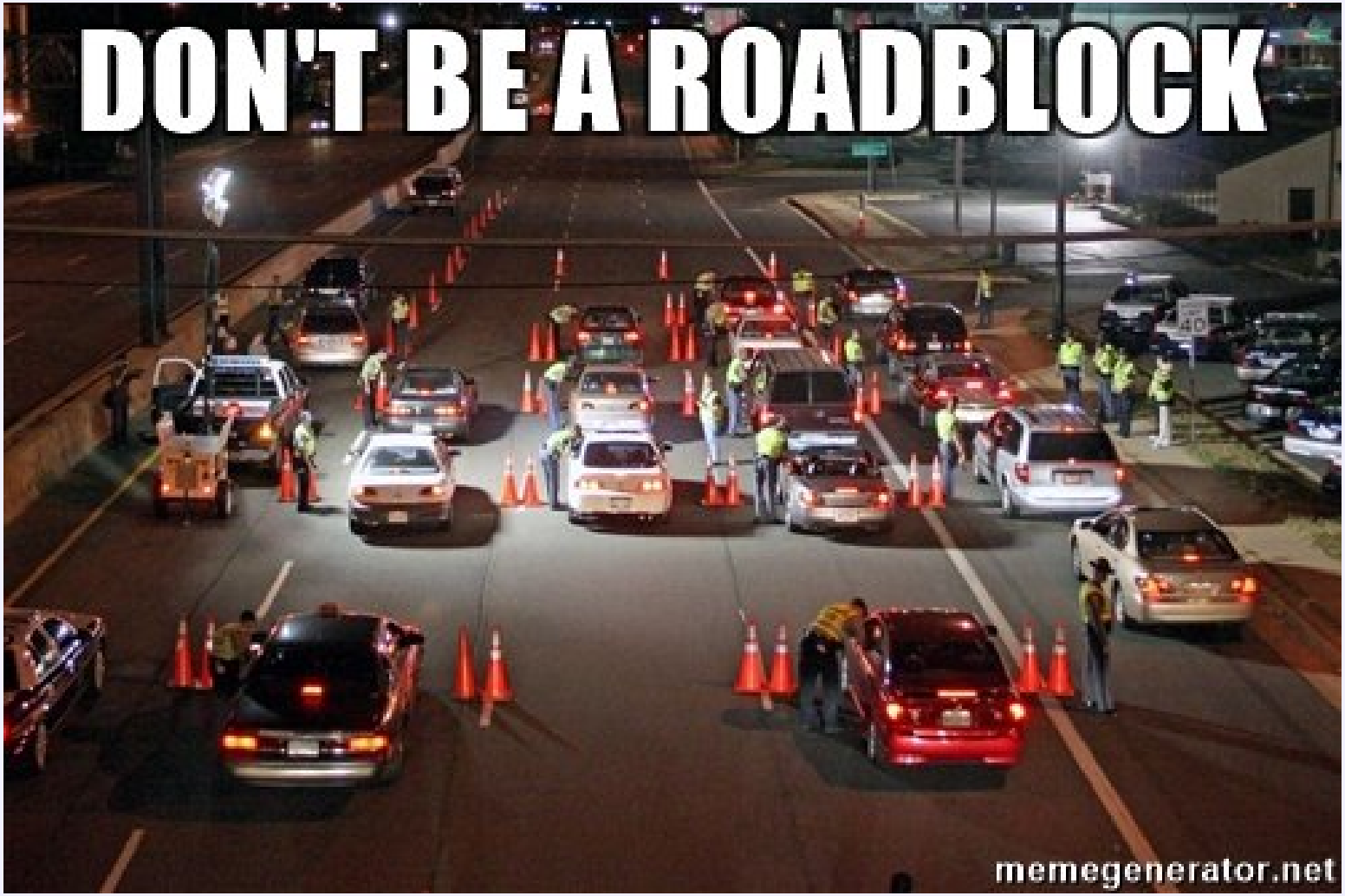

I DON'T ALWAYS VALIDATE
MY VA RESULTS
BUT WHEN I DO, I
DO IT IN PRODUCTION
imgflip.com

OWASP
The Open Web Application Security Project

Provide free training to developers


ROOMMATES
SEC AND DEV!
imgflip.com


UHHH, WHO'S PAYING?
imgflip.com


DON'T BE A ROADBLOCK
memegenerator.net

Job Shadowing

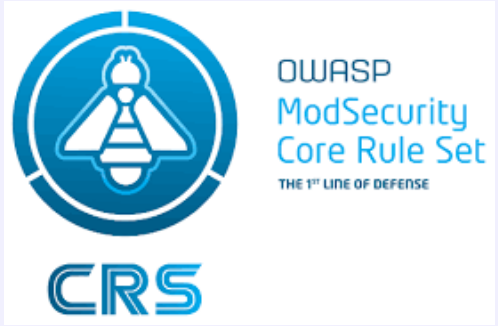
OWASP
The Open Web Application Security Project

Give Developers Security Tools!











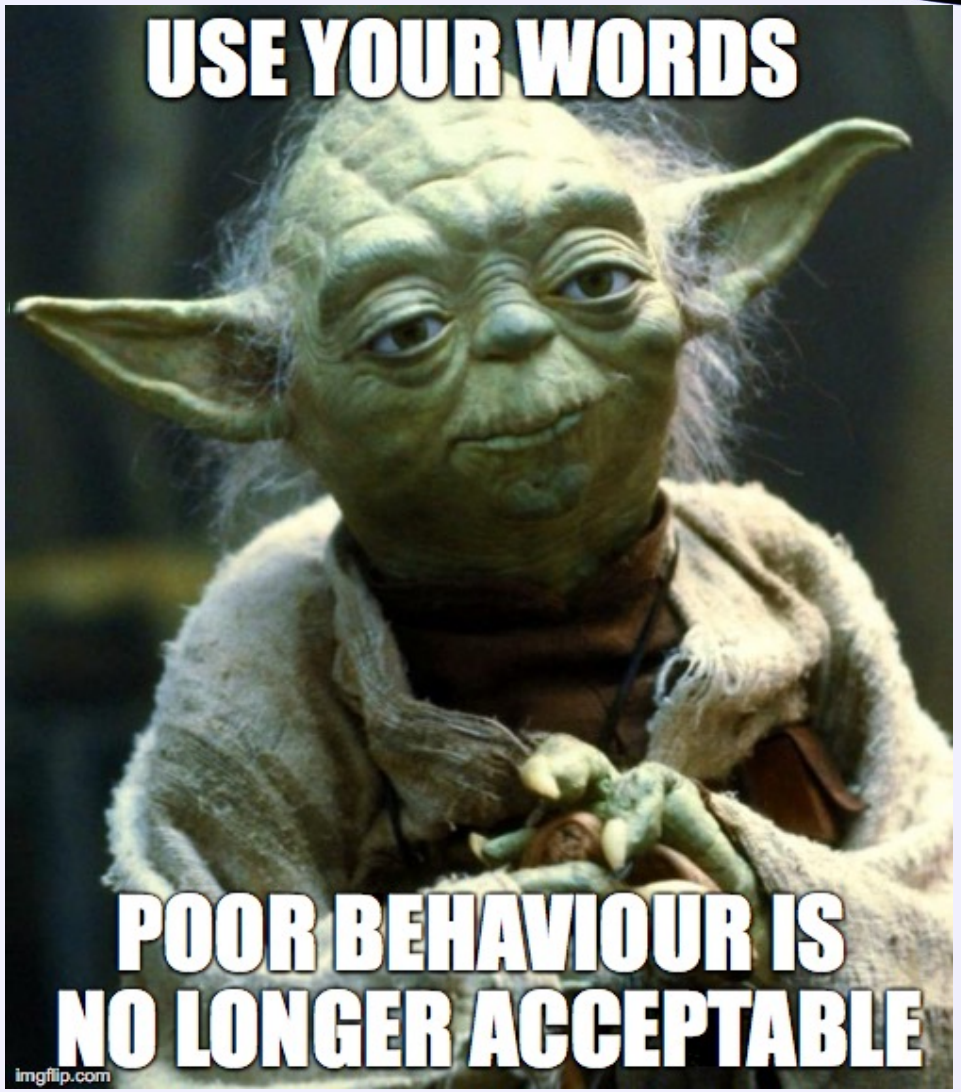


PROVIDE FREE SECURITY TRAINING
OFTEN
DON'T STOP!
imgflip.com

OWASP
The Open Web Application Security Project

OWASP: Your new BFF

The Open Web Application Security Project


USE YOUR WORDS
POOR BEHAVIOUR IS
NO LONGER ACCEPTABLE
imgflip.com


NO MORE BLAMING
BE RESPECTFUL
AT ALL TIMES
imgflip.com


WE ALL MAKE MISTAKES
NO NEED TO RUB MY NOSE IN IT
imgflip.com


OPEN DISRESPECT
CANNOT BE TOLERATED
imgflip.com

A message for conferences

OWASP
The Open Web Application Security Project

No more “we’re screwed” keynotes.

OWASP
The Open Web Application Security Project

Summary:

1. Support dev and sec team with processes, training, and resources so they can confidently get the job done.
2. Repair relationship.
3. Do not accept ‘bad’ behavior anymore.

OWASP
The Open Web Application Security Project

ANY QUESTIONS ?

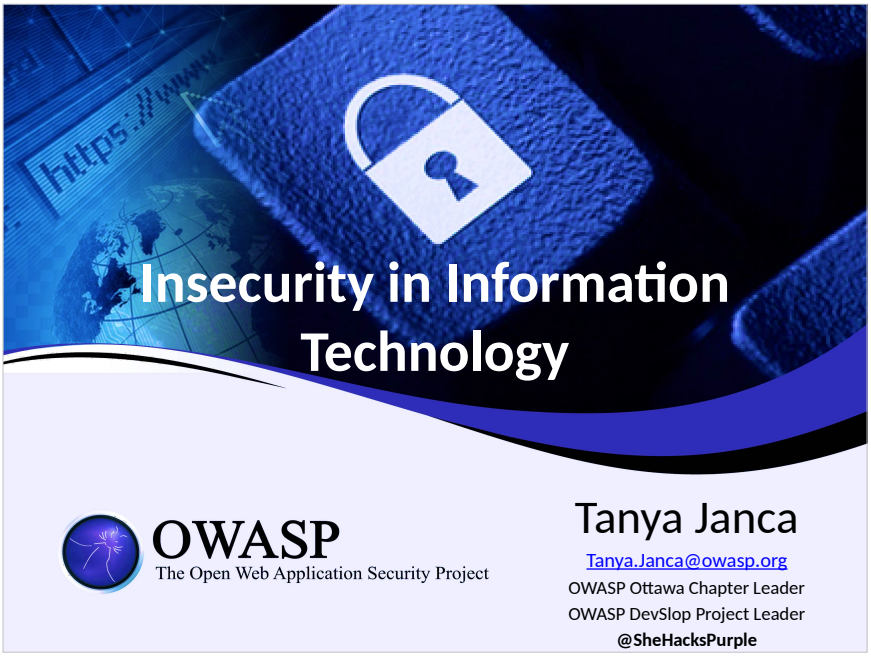
Tanya Janca

Tanya.Janca@owasp.org

OWASP Ottawa Chapter Leader

OWASP DevSlop Project Leader

@SheHacksPurple



New ITSec drinking game: drink every time someone says machine learning, Artificial Intelligence, or Block chain will fix everything.

The “I’m Qualified” Slide

Tanya Janca: Application security evangelist, web application penetration tester and vulnerability assessor, trainer, public speaker, ethical hacker, OWASP Ottawa chapter leader, OWASP DevSlop project leader, effective altruist, software developer since the late 90’s.

I’m extremely concerned about application security, and you should be too. Let me tell you more.

Thesis:

People make stupid decisions when they feel insecure.

Conflict between security and development makes some people (employees) feel insecure.

When this happens, insecure software is often the result.

Changes are required to solve this problem.

Warning:

This is not a talk about “feelings”.

This talk is about the bottom line, getting the job done, and making software secure.

However, it might get uncomfortable at times.

Try not to feel too defensive.

The Fine Print:

Although I relay all examples in the first person, as though I was there, these are not all my personal examples. They are from multiple members of the InfoSec community.

Talk Outline:

- Problem(s)
- Solution(s)

Problem:

The way many IT shops are run can create feelings of job insecurity in employees.

Security testing is performed so late in the game that developers

1) Don’t fix anything
2) Resent security for presenting last minute challenges

Developers

OWASP
The Open Web Application Security Project

- Do security testing without the security team, making them feel unrequired.
- Don’t cooperate with the security team to enable security testing, claiming they have no time/implying security is low priority.
- Ignore security advice from security reports and do not implement any or most of the mitigations.
- Do not take/ask for advice from the security team.

Security

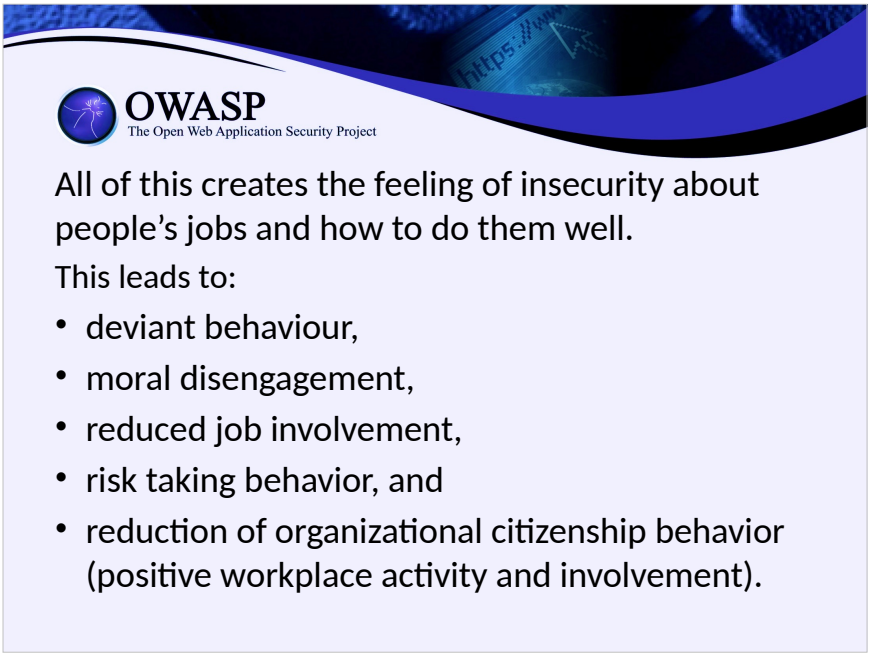
OWASP
The Open Web Application Security Project

The security team quotes policies at developers, or sends them an unvalidated 500 page report.

Security Team

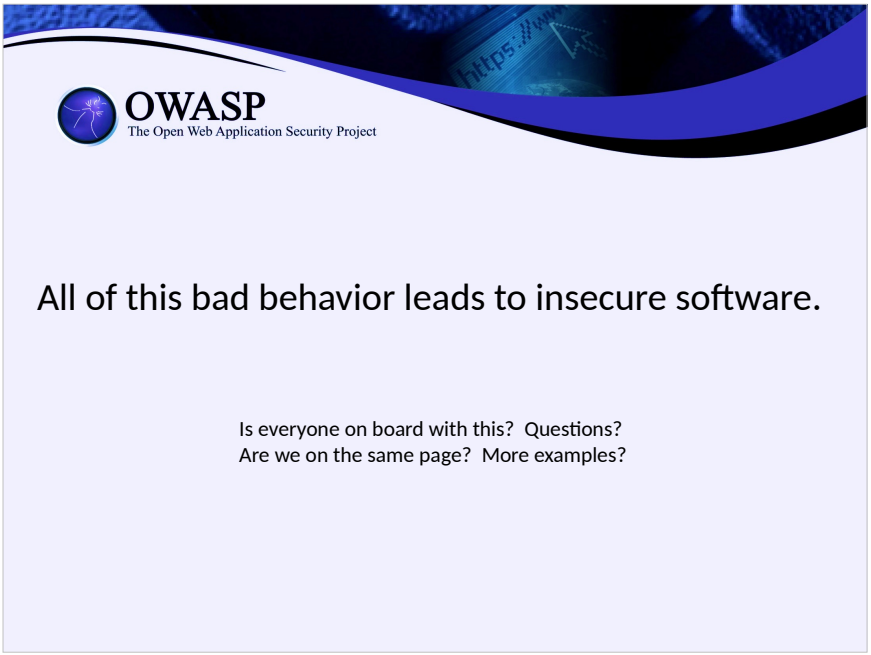
OWASP
The Open Web Application Security Project

- Does not give **usable** security guidance to the developers when asked.
- Acts or is seen as a gate, slowing down the SDLC.
- Adds project requirements without explanation, “because security”.
- When revealing issues, they make developers feel incompetent.



When people are acting like this, what kind of software are they making? Are they being diligent? Are they going the extra mile? I think not.

** There are many published studies to support these findings that job insecurities lead to these behaviors.



"Choose your own adventure" books.

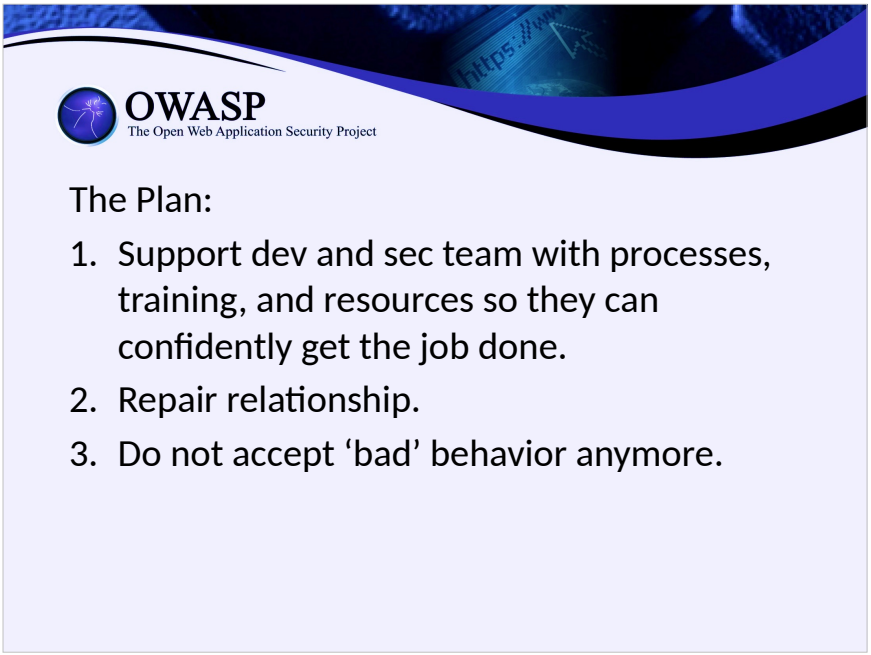
Make sure it is very clear that we are moving onto the next part of the talk.

- The security team has no idea how to give advice on software so they forward long, unreadable security documents that aren't at all helpful to try to hide the fact that they don't know. ITSG-33 or NIST example, depending on audience.
- Developers try to publish their apps without security seeing them at all. No example required, sadly we've all witnessed this.
- Security standard or policy published but not announced, developers not consulted or informed
- Story of VA teams sending unvalidated reports, wasting developers time and making the sec team appear & feel incompetent
- Story of "climb out the window and down the trellis", extraordinarily poor management example
- More examples from various sources, depending on length of time allowed for talk.

Story of "we use SAML tokens" (making developer look like a fool in front of client, developer responds in kind)

- Story of ridiculously inaccurate and vague web app standard (XSS and CSRF addressed with same mitigation and errors) published by sec team with no notification at all to the dev team

Make sure audience switches gears with you that this is the second part of the talk.



1)training, tools, standards, processes (appsec program & team)
2)Repair relationship (co-locate, team building, etc)
3)speak out and punish bad behavior, managers set good examples

If you do not have an application security team (the part of the security team that knows software and talks to developers), get one. Now. Run, don't walk. If you work in an enterprise sized business it is not acceptable to not have one, even if you have to lure security-minded developers to your team in order to staff it.

AppSec is the cause of approximately a quarter of security incidents, why aren't you spending a quarter of your security budget on it?

Setup a secure SDLC, and start security earlier. Perform security throughout the entire SDLC, so that fewer bugs are found near the end, meaning less last-minute-release-stress. Remember, if you aren't doing IT properly, you can't do security properly.

For every new major software project assign an AppSec representative. That person will stay on the project and go to major meetings to offer security advice. Anything security-related that the project team needs throughout their project this person will help with, either themselves or (again) hiring out. This will be their “go to person” for anything security related, making dealing with the security aspect of their project easier and hopefully no longer considered “painful”.

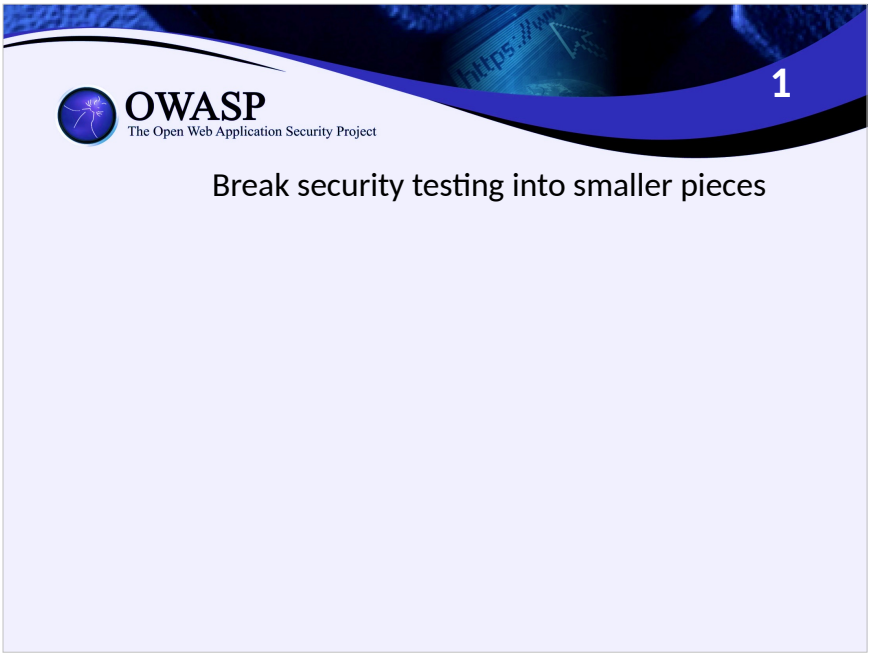
Create secure design and coding standards, with lessons/workshops/training to go with it. Consultations with developers are required to ensure 1) they agree with it 2) it's possible and 3) it makes sense. Then both teams need to come up with a plan of when they can comply (accept that legacy apps won't be compliant from the start), and how all teams can help get them there. Once the standard is agreed upon and published, promote it. This means workshops and/or training, not hand-slapping. Socialize it.

Create security lessons/workshops/training to go with those standards you just made. Teach what you expect them to do. Don’t make them guess.

The security team is NEVER allowed to respond to requests for help by sending links to extremely long documents (ITSG-33 or NIST, for example) that are essentially unreadable to non-security-people and leave them more confused than before. Give specific and detailed advice. If you don't know the answer conduct research or hire someone to do it for you. Not answering is NOT an option.

A company MAY NOT publish an unreadable (too technical/all security jargon) to an unfindable/borderline-hidden location onto the intranet and call it a day. This is NOT useful. This is not helpful. This is making a problem, not solving one.

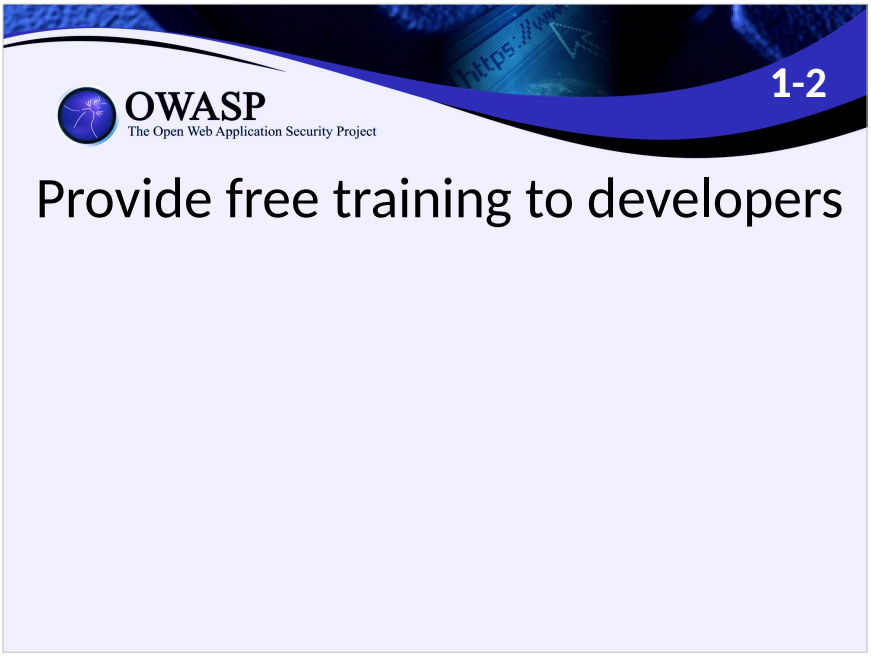
There is no shame in going on training so that you have a better handle on what you are expected to do for 40 hours+ per week. If you are weak in a specific area, ask for training. If there's no budget train yourself online. If you still don't know, call a pro, that's what consultants are made for, you can always hire out if you don't know. Never leave important things as unknown or ambiguous, this is where insecurity starts.



Security testing is cut into smaller pieces, so that it is more manageable and preferably does not slow down the SDLC any more than absolutely necessary. For instance, doing static code analysis that only looks for XSS in one round, then another that only looks for SQLi, rather than doing one huge sweep that would take 2-3 weeks to analyze. Send them an OWASP Cheat sheet with the results or a link to your secure coding guidelines (which exist, right?). So you're showing them a problem but also how to fix the problem in the same breath.

Repair your relationship: Step 2

Ensure that all security testing results have been fully validated, no more false positives. If you are unsure, hire out/get more training.

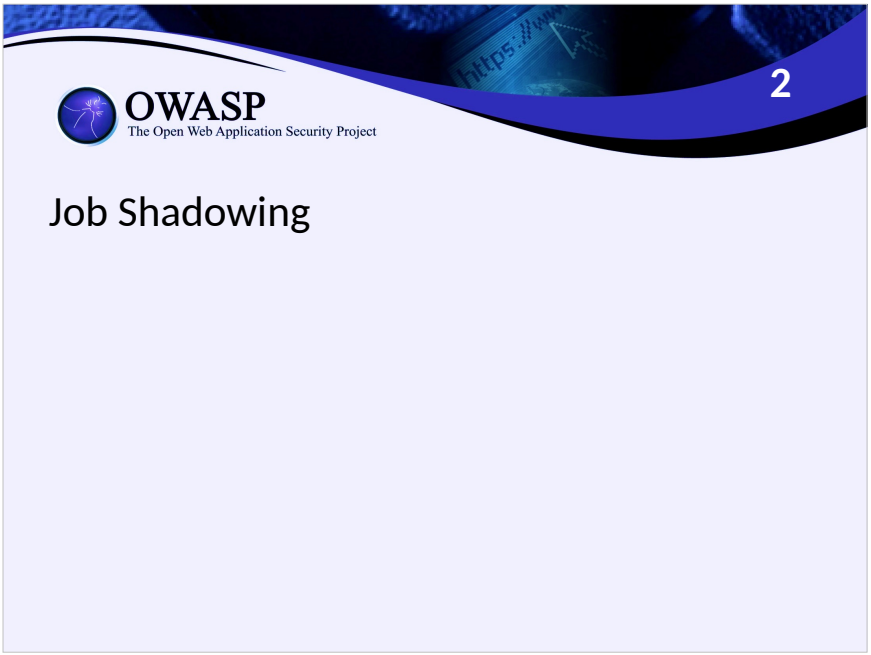


Provide free security training to developers, and the rest of IT while you're at it. They should not be expected to pay for this out of their own training budgets. If you are a developer and you "need to know everything" for your job, you are not going to spend your limited training dollars on security when there are ten other topics that need your attention this year. But if the training was free and you had approval from your manager to attend…. you'd be there in a second. Make this a reality for the dev team and watch your apps become more secure overnight.

Physically locate the application security team near the developers. It’s more difficult to be rude to someone’s face.

Try not to worry about who has to pick up the bill. I realize training and consultants costs money, but 1) it’s worth it if you need it 2) consider it a long term investment, 3) you are getting your mandate done (software that is more secure) so you might as well pay for some of it and 4) it all costs less than a breach.

CIA: don’t become the threat to Availability.
Enable developers, don’t be a roadblock. Never say “No.” Say “you can’t do that, but you CAN do x, y or z”. Give them options. If you don’t know the answer offer to work through the problem with them; brainstorm until you create a solution together. Understand their problem and what they are trying to accomplish. Explain why they can’t do it the way they want to do it, preferably with examples of the risk, so they understand where you are coming from too.
No more saying “Because security”, you must always explain the reasons.



Bring a developer on a security incident, on a pentest, on a code review. Sit in and help a developer fix the security bugs you just reported to them. Create opportunities for job shadowing or other ways for the teams to see things from each other's point of view. When you understand the issue, it's easier tor be sympathetic.

Give developers security tools; they might actually use them. Give them web app scanners, give them static analysis tools, buy them books, whatever they want. Help them use them, show them how. They are basically doing your jobs for you! This is a great deal!

CONTINUE to give regular security awareness training that is job-specific, so that everyone is aware. OFTEN. Even for people who do not have privileged or special access. When there is a culture of "security is everybody's job", everyone is already on board.
Be careful not to overwhelm them, if they are quite busy, then just give them a little, if that is all they have time for.

Invite your developers to participate in OWASP. Offer to host it at your company! If there isn't a chapter in your city, start one! This is a great way to get your developers interested in AppSec, because they (we?) are the care bears of security!

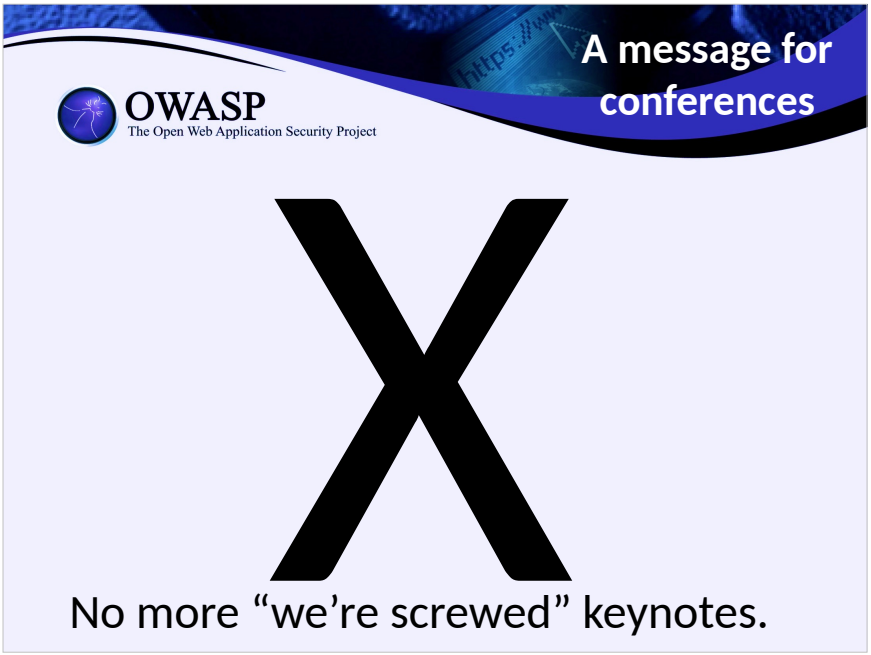
We are in the third and final part of the solution: culture changes. Bad behaviour is no longer acceptable, and management needs to make that clear by speaking out against it if and when they see it. Even if it’s uncomfortable. Even if it’s us who is saying it.

Put an end to blame and be sensitive of the words we use (both sides). We shouldn't care about who caused it, we only care about how we are going to fix it. Never say that a piece of software is "garbage", that is someone's creation that you are talking about. Never say that security rules are ridiculous or overbearing, the person who wrote it just wants to protect the company. We are all well-meaning when we come to work, so use language that reflects that intention by always be professional and respectful at all times. If you have open disrespect happening in your workplace you have a serious problem.

If someone makes a mistake try to ensure that they can save face, there’s no need to rub someone’s nose in it.

Disrespect cannot be tolerated as part of your culture if you want it thrive. Last slide on section 3.

Even if you are in a team meeting and no one can hear, if you are talking negatively about the other team it will be reflected in your body language and mannerisms the next time you see the other team. And you never know who can hear you. (examples)



If at all possible, stop booking the “we’re all fucked” keynote talks at security conferences. The ones that present huge problems but few or unclear solutions. They do not inspire confidence, so why give them audience? They do not help the situation, they only discourage security practitioners who already have an uphill battle. FUD Marketing: Fear, Uncertainty and Doubt.

Summary:

1. Support dev and sec team with processes, training, and resources so they can confidently get the job done.
2. Repair relationship.
3. Do not accept ‘bad’ behavior anymore.

ANY QUESTIONS ?

Tanya Janca
Tanya.Janca@owasp.org
OWASP Ottawa Chapter Leader
OWASP DevSlop Project Leader
@SheHacksPurple