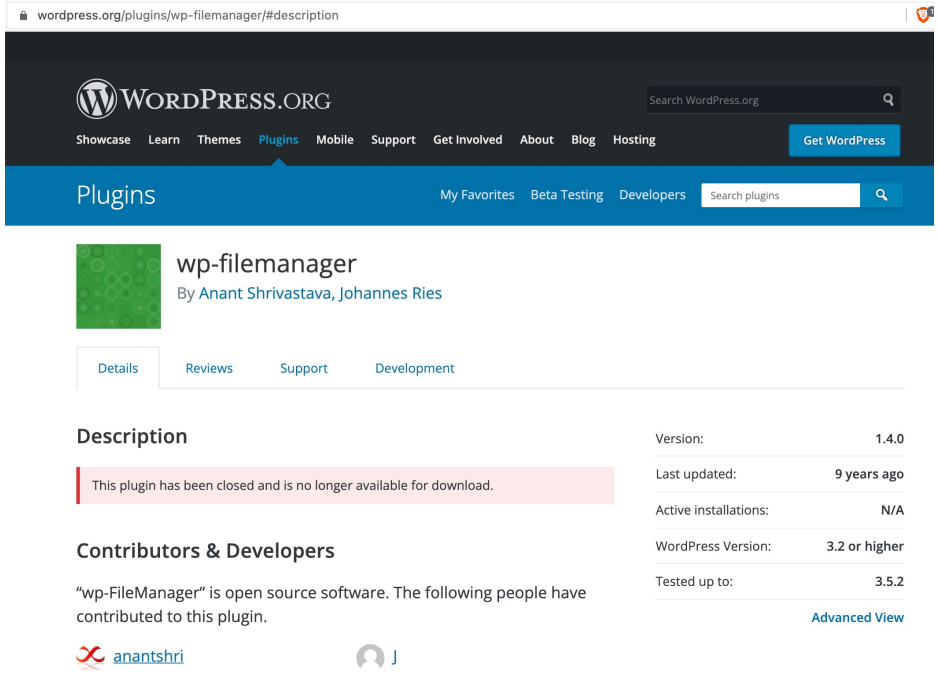
Developers and Security

My 2 paisa’s based on decade and half of my experience

Anant Shrivastava
Geek | Researcher

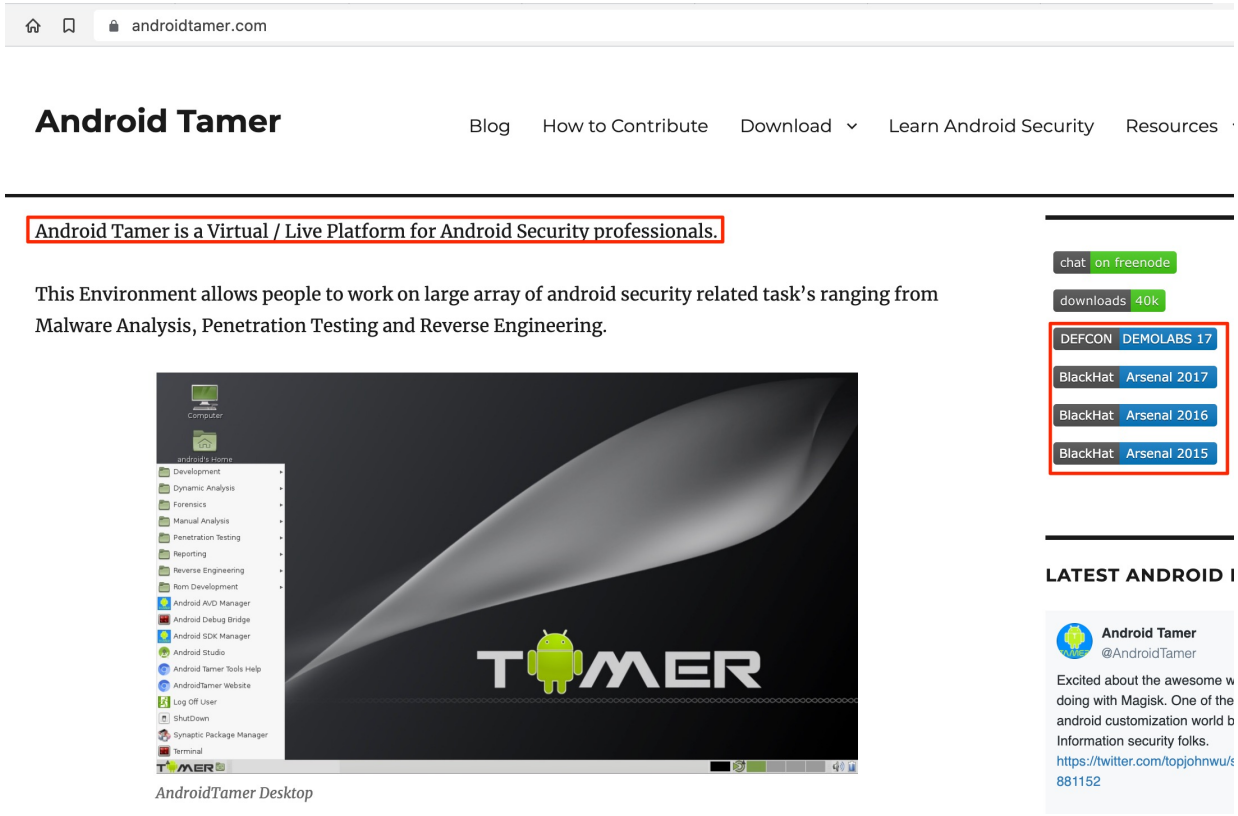
Why my thoughts matter

- Been a **Developer/Maintainer** of a moderately successful wordpress plugin.
- Closed plugin 9 years ago coz of other commitments and …
- Faced non responsible disclosure
- So fixed the bug and then called it quits



Why my thoughts matter

- **Maintained** a custom Debian based distribution single handedly since 2012-2018
- Next version to come out in 2 months timeframe.
- The entire infrastructure and related setup was handled as primary **dev + admin**



AndroidTamer Desktop

Why my thoughts matter

- Run a static code analysis project called codevigilant
- As of now focused on PHP mainly wordpress plugin and themes
- 200+ public disclosures,
- 150+ to be disclosed.
- Lots under validation process

Built and Maintained

Backend, automation, website

Disclosure process, co-ordination

codevigilant.com/category/

code vigilant

To err is human.. To fix is humanity

HOME DISCLOSURE BLOG RSS FEED ABOUT TEAM

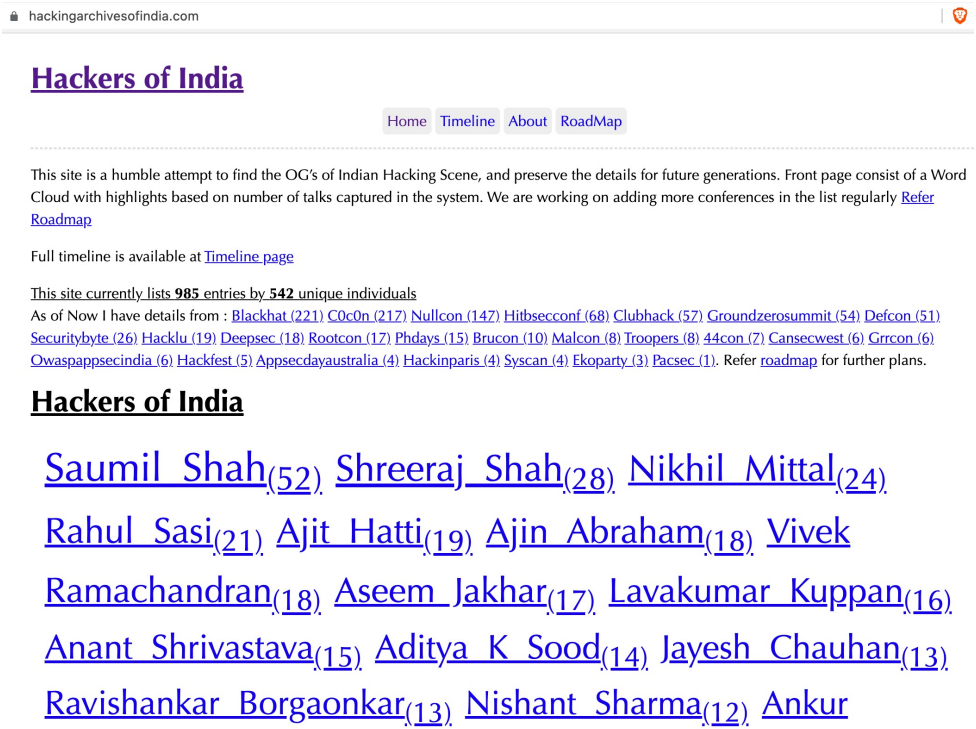
Category

Total Public Disclosure : 204

Category	Total Findings	Percentage	2014 Count	2021 Count
Injection	46	23%	9	37
Local File Inclusion	7	3%	6	1
Cross Site Scripting (X.S.S.)	119	58%	115	4
S.S.R.F.	4	2%	3	1
Unvalidated Redirects and Forwards	3	1%	3	0
Information Disclosure	1	0%	1	0
Using Components with Known Vulnerabilities	21	10%	21	0

Why my thoughts matter

- Building a fully static HTML CSS only website
- Website heavily data driven
- Specific aim to not use Javascript at all in website directly
- Coding my own hugo theme as well as writing custom wrappers



Why my thoughts matter

- Running my own collection of websites (~10+) on Wordpress self hosted since 2007
- Maintained entire offensive, defensive and operations network for an infosec company for 5+ years single handedly
- Build automations and supporting various opensource projects via time, effort, money, documentation etc

Why my thoughts matter

Worked at

Spoken / Trained at

Developers and Security

My 2 Paisa’s based on decade and half of my experience of Development / Administration / Infosec

Software eating the world

https://www.wsj.com/articles/SB10001424053111903480904576512250915629460

NES, A NEWS CORP COMPANY ▼

70.61 0.56% ▼ S&P 500 **4555.94** 0.46% ▼ Nasdaq **14425.19** 0.56% ▼ U.S. 10 Yr **4/32 Yield** 1.835% ▲

THE WALL STREET JOURNAL.

English Edition ▼ | Print Edition | Video | Podcasts | Latest Headlines

Home World U.S. Politics Economy Business Tech Markets Opinion Books & Arts Real Estate Life &

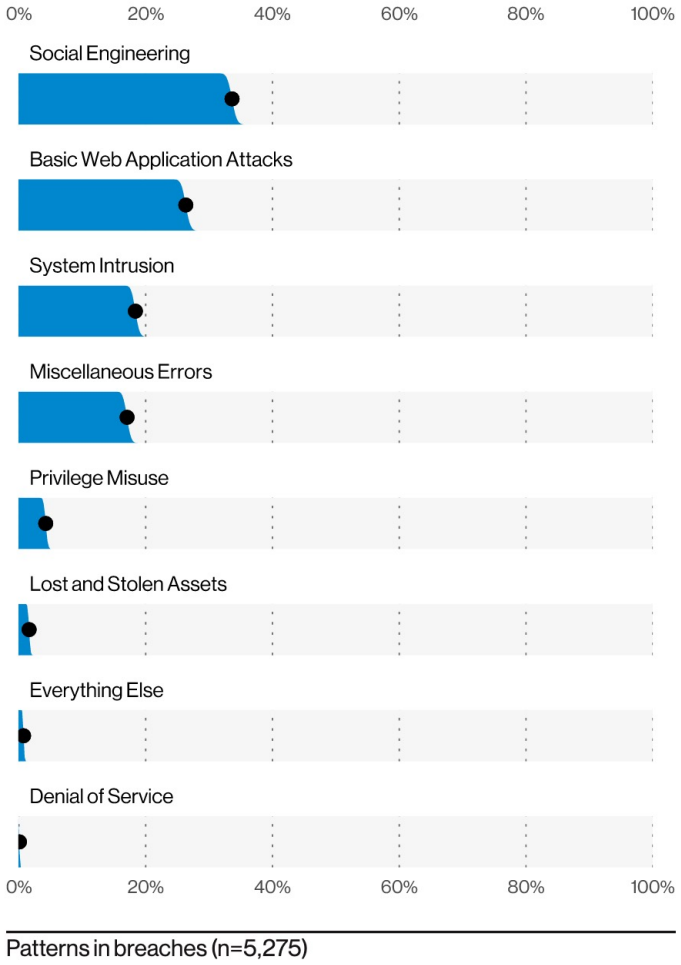
Why Software Is Eating The World

By Marc Andreessen

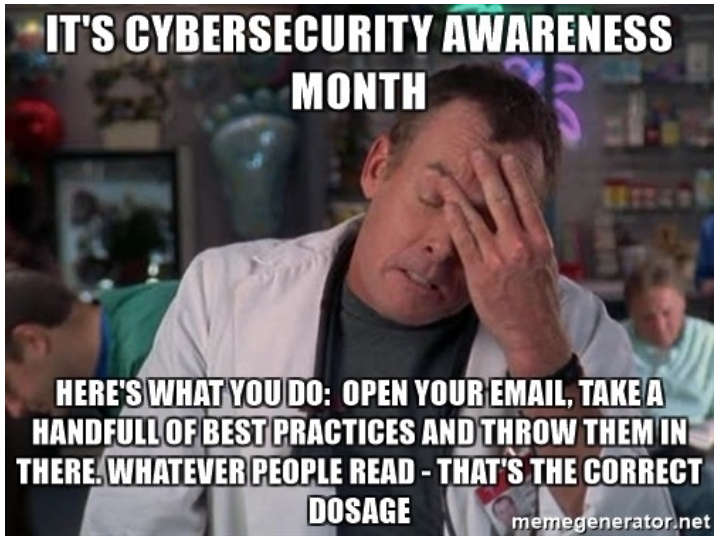
August 20, 2011

Data Breach Investigation Report 2021

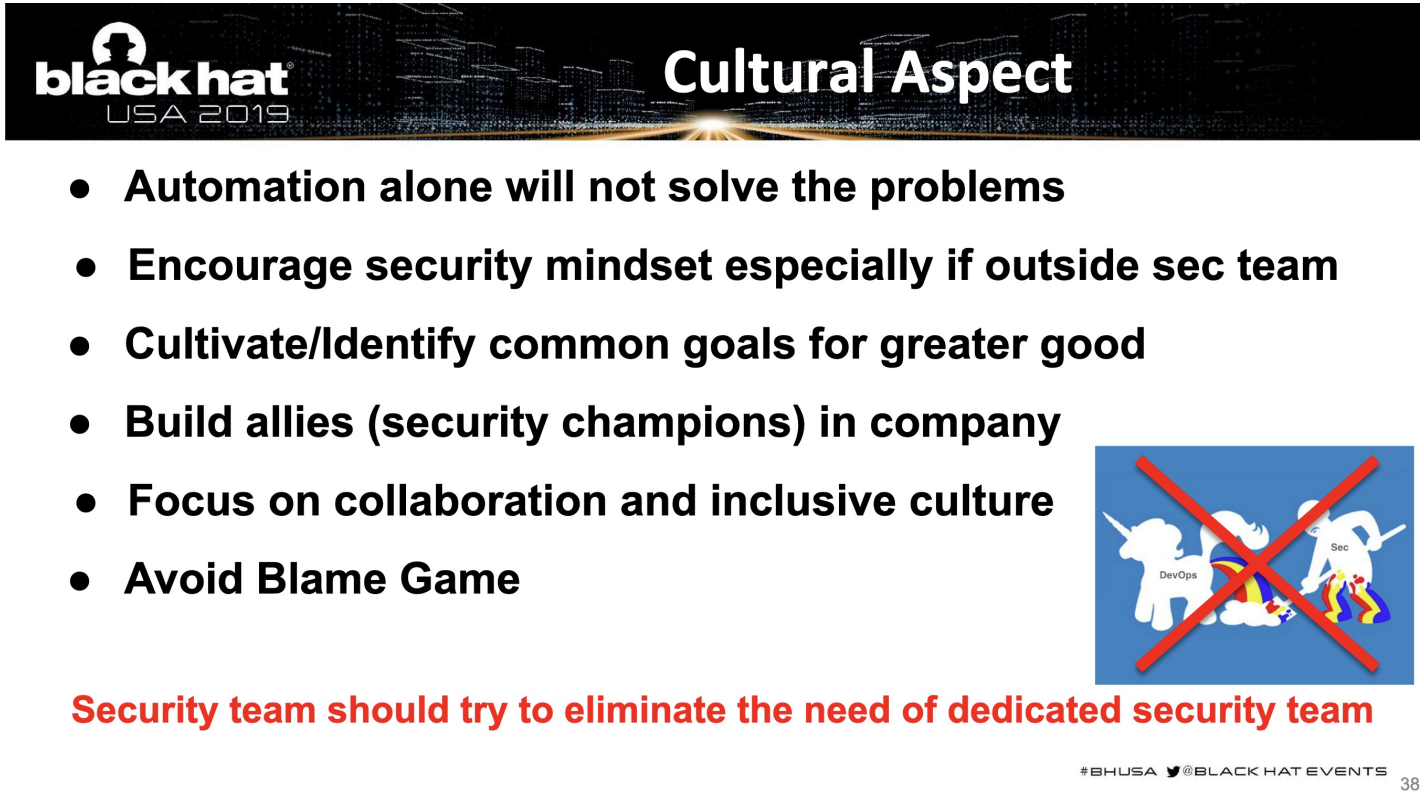
- Web applications are primary technical cause of breaches
- 2011 to 2021 : 10 years things have flipped



The mess of misunderstanding



How to move forward



https://www.slideshare.net/notsosecure/devsecops-what-why-and-how-blackhat-2019

Collaboration is the key

Defenders Assemble

- Defenders need to focus energy and work together
- Orgs sponsor defense tools and support public collaboration
- Appreciate the firefighters but also appreciate those who don't let the fire start
- Take care of yourself : It's a never-ending war don't focus on the battle only
- Focus on detection and containment
- Assume breach will happen: what happens after, that makes the difference

Refer: https://medium.com/@johnlatwc/the-githubification-of-infosec-afbdbfaad1d1R

OpenSSF
OPEN SOURCE SECURITY FOUNDATION

ATT&CK®

SIGMA

DEFEND™

https://www.slideshare.net/anantshri/diverseccon-keynote-my-2-paisas-on-infosec-world

Developers have a more ingrained role to play

- Security is considered an art and not a science
- Security needs to be commoditized and converted to science
- How do you do it
- Exactly how dev’s have done this with infrastructure
  - From manual and long drawn process we have reached to
  - All codified near instantaneous infrastructure deployments

DevOps needs to eat security

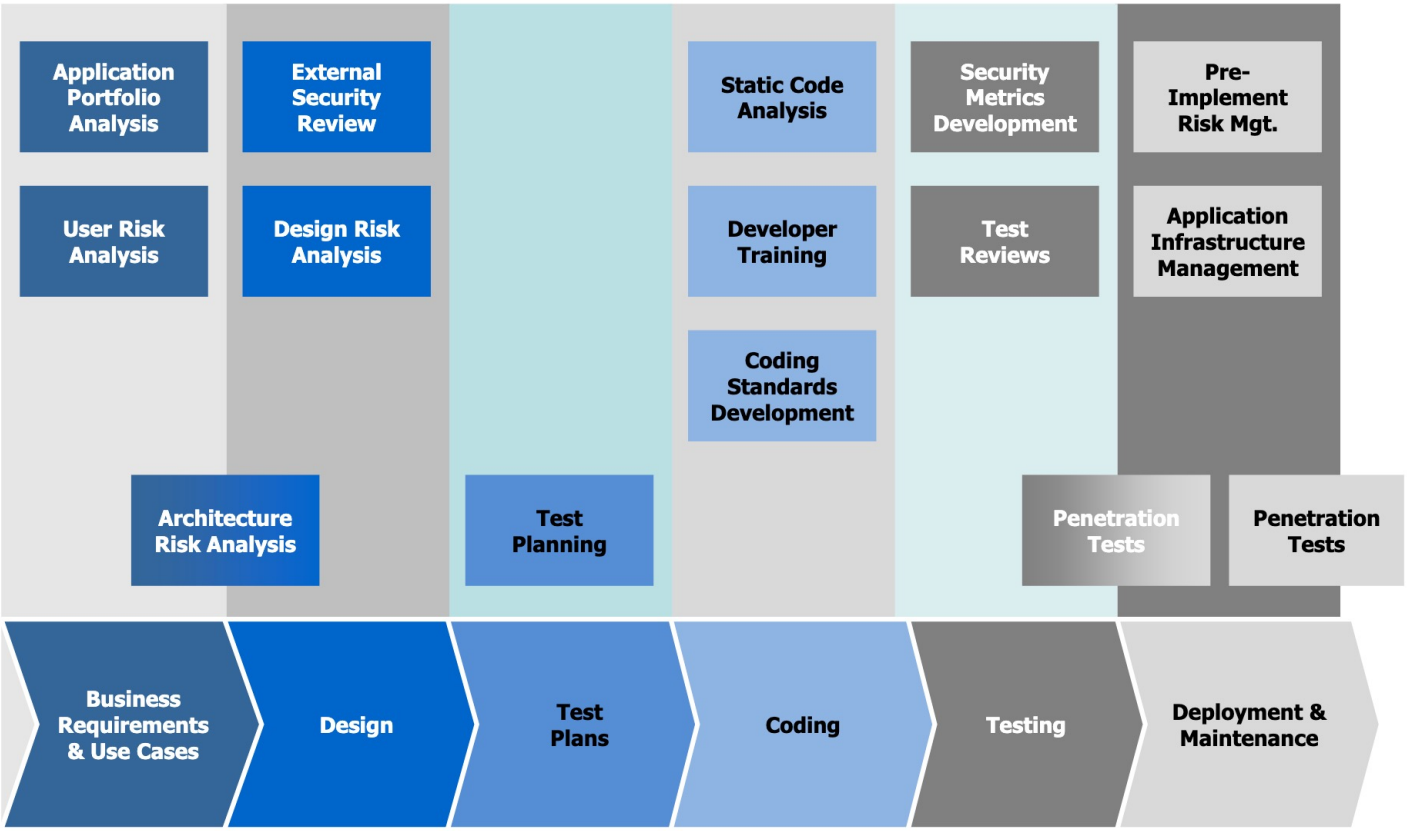
- DevSecOps as a term should not have existed but its here and people use it.
- Eat security art side and make it security science
  - Automatable
  - Documented
  - Testable
  - Repeatable

It may not be 100% possible but it is achievable in high 90’s

Developers to take full ownership

- No one and I repeat no one other then dev knows code better
- Leverage security team and support function:
  - Take inputs from them as early and as often as possible
  - Take final ownership of product in your hand
- If security team acts as bottleneck they are doing it wrong

World needs more stable and secure software



Eoin Keary & Jim Manico

https://owasp.org/www-pdf-archive/Jim_Manico_(Hamburg)_-_Securiing_the_SDLC.pdf

Collaboration in action

Anant Shrivastava @anantshri · Jan 17

I am going to talk with a group of web app developers in a few days. seeking inputs from various infosec professionals what is the one thing you would like me to tell them (it could be a learning, a request, suggestion) all thoughts welcome abuse/absurdity totally not welcome.

Show this thread

35 9 31

Impressions	New followers	Profile visits
10,819	0	53

https://twitter.com/anantshri/status/1483031251439464448

Mix bag of responses

- Some responses specifically asking devs to do this or that
- Lots of suggestions to follow secure early or put security in early stages.
- Unsurprisingly lots spooked with third party dependencies
- But a common theme emerged in all these tweets especially from veterans of this field.

Embrace your power

Simon Bennetts
@psiinon

Replying to @anantshri

"You are responsible for the security of your app. Security people can help you, but you need to build and maintain them to be secure". Oh, and @zaproxy can help a _lot_ 😄

4:55 PM · Jan 17, 2022 · Twitter Web App

1 Retweet **1** Quote Tweet **12** Likes

Jim Manico
@manicode

Replying to @abhaybhargav @anantshri and 7 others

In all seriousness, listen to developer needs carefully and address them from a security perspective. Developers are highly intelligent engineers and trust them to do good once you provide good information on how to fix and prevent AppSec issues.

12:04 AM · Jan 19, 2022 · Twitter for Mac

6 Likes

Glenn Pegden - ☎ 💻 💾 ⓗǎ(c)ǽĘ̄
@GlennPegden

Replying to @GlennPegden and @anantshri

The InfoSec community who encourage a myth that security is a complex discipline that needs dedicated security specialists to do anything security related. This is nonsense, most devs and testers can quickly match the skills of an junior pentester if given a little help

6:30 PM · Jan 17, 2022 · TweetDeck

1 Retweet **4** Likes

Glenn Pegden - ☎ 💻 💾 ⓗǎ(c)ǽĘ̄
@GlennPegden

Replying to @GlennPegden and @anantshri

Finally, don't let Security "own" your stuff, they may want to police it and be a blocker, but by taking ownership of your own security, you can encourage them out of the way of your delivery pipeline and assist you rather than slow you down

6:48 PM · Jan 17, 2022 · TweetDeck

2 Likes

Lots of other useful advices

Bharadwaj Machiraju
@tunnelshade_

Replying to @anantshri

Question and understand the why's of their infosec professionals' recommendations (when possible)

This improves the game of both teams eg. devs learn bug patterns which they can avoid/look for in PRs and security teams learn interesting ways to fix in nuanced environments

6:38 PM · Jan 17, 2022 · Twitter for iPhone

3 Likes

Sandesh Anand
@JubbaOnJeans

Replying to @anantshri @manicode and 7 others

Security is not a point in time activity. So, get into the mindset of thinking of security as long as you are building something

8:37 PM · Jan 17, 2022 · Twitter Web App

4 Likes

Rishi Narang
@oufsec

Replying to @anantshri

- accountable for what you develop
- preempt the use (+ abuse) cases.
- hash the secrets & encrypt data in transit
- client side security controls can/will be tampered.
- don't trust browsers. If it can be hacked today, it will be hacked tomorrow.
- peer review.

6:16 AM · Jan 18, 2022 · Twitter for iPhone

1 Like

Veronica
@verovaleros

Replying to @anantshri

1. Do not develop your own protocols
2. Encoding is not encryption
3. Use HTTPS

4:33 PM · Jan 17, 2022 · Twitter Web App

2 Retweets **17** Likes

Some very creative ideas

Robin
@digininja

Replying to @digininja and @anantshri

Slight off topic, but if they can get QA at least partially trained in security then they will get a huge win. QA love tickbox checks and automation, they may not understand why a single quote causing an error is a critical finding, but they'll be very good at testing for it

4:44 PM · Jan 19, 2022 · TweetDeck

1 Like

ScottG @scott44017 · Jan 17
Replying to @anantshri
ONE thing is to see/hear from the team what SECURITY CONTROLS and PROCESSES that they perform or are performed by others to their code, software components and platforms.

Think of this a review of their architecture and sdlc process to find things that matter to them

1 4

ScottG @scott44017 · Jan 17
From this I would lead to a couple owasp items:

1. Proactive Controls
2. ASVS
3. Owasp Integration standards

Much depends on app and team.

A Good read here owasp.org/www-project-in...

3 4

ScottG @scott44017 · Jan 17
ASVS level 1 is a good starting point from security pov for development requirements or testing/validating checks

owasp.org/www-project-ap...

Can be used at either stage in lifecycle effectively

1

Did we forgot dependency tracking

Glenn Pegden - ☎️📟💾 ⓜ̆ä̆ⓒ̆⁄̸E̦ī̆
@GlennPegden

Replying to @GlennPegden and @anantshri

Plan ahead. Dependancy management is a nightmare, but think about how one small just crucial security fix in a component use use could impact your stack. Is it safe to update and rebuild everything or have you relied on pinned versions and custom forks that break updating

6:34 PM · Jan 17, 2022 · TweetDeck

2 Likes

AMol NAik
@amolnaik4

Replying to @anantshri @manicode and 7 others

a request: plan for upgrading code dependencies regularly without breaking the app

4:36 PM · Jan 17, 2022 · Twitter Web App

6 Likes

Teri Radichel #cloudsecurity #cybersecurity
@TeriRadichel

Replying to @anantshri

So much more than one. But include with this: Every line of code is a potential bug. Use abstraction. I have more blogs like this coming based on what I see on pentests and 25 years of dev. Maybe another book coming.

```
            if isinstance(v,dict):printdict(v,tabs);return

if (len(str(v))) > 0 and str(v) != '[]' and v != None:
    print(tabs + k.strip() + ": " + str(v).strip());
#else:
    #print(k)
return

isinstance(v,str):
if not isinstance(k,list):
    if isinstance(v,dict): printdict(v,tabs);return
    if isinstance(v,list): printlist(v,tabs);return
    print(k + ":")
    if isinstance(k[v], dict):printdict(k[v],tabs); return
    if isinstance(k[v], list):printlist(k[v],tabs); return
```

medium.com
Every Line of Code is a Potential Bug
How to reduce the chances of a security flaw in your application with the principle of abstraction

Some basic ideas to kickstart the brain

- Use customizable tools like semgrep
- Learn how to test the vulnerabilities
- Try to find bug as close to writing code as you can

```
IDE Plugin > git commit hook > CI tool
```

Quick References

- https://owasp.org/www-project-application-security-verification-standard/
- https://owasp.org/www-project-proactive-controls/
- https://owasp.org/www-project-integration-standards/
- https://owasp.org/www-project-spotlight-series/

Important points recap

- Developers are the best judge of how the code gets changed
- Security teams can help but they can't take ownership
- Pick tools that work for you and automate sec stuff

Thanks, and open to questions