PRODUCT SECURITY BEST PRACTICES

Pawan Bhandari

Disclaimer

The opinions and thoughts expressed in this presentation and subsequent slides are solely of my own and do not necessary reflect the views or position of my employer.

The information provided in the slides are not comprehensive and created solely on the basis of experience I gather while working in projects.

About Me

Who I am

- Currently working with McAfee
- Security & ML Enthusiast
- Blue Team Member
- Love to cook and play with my daughters

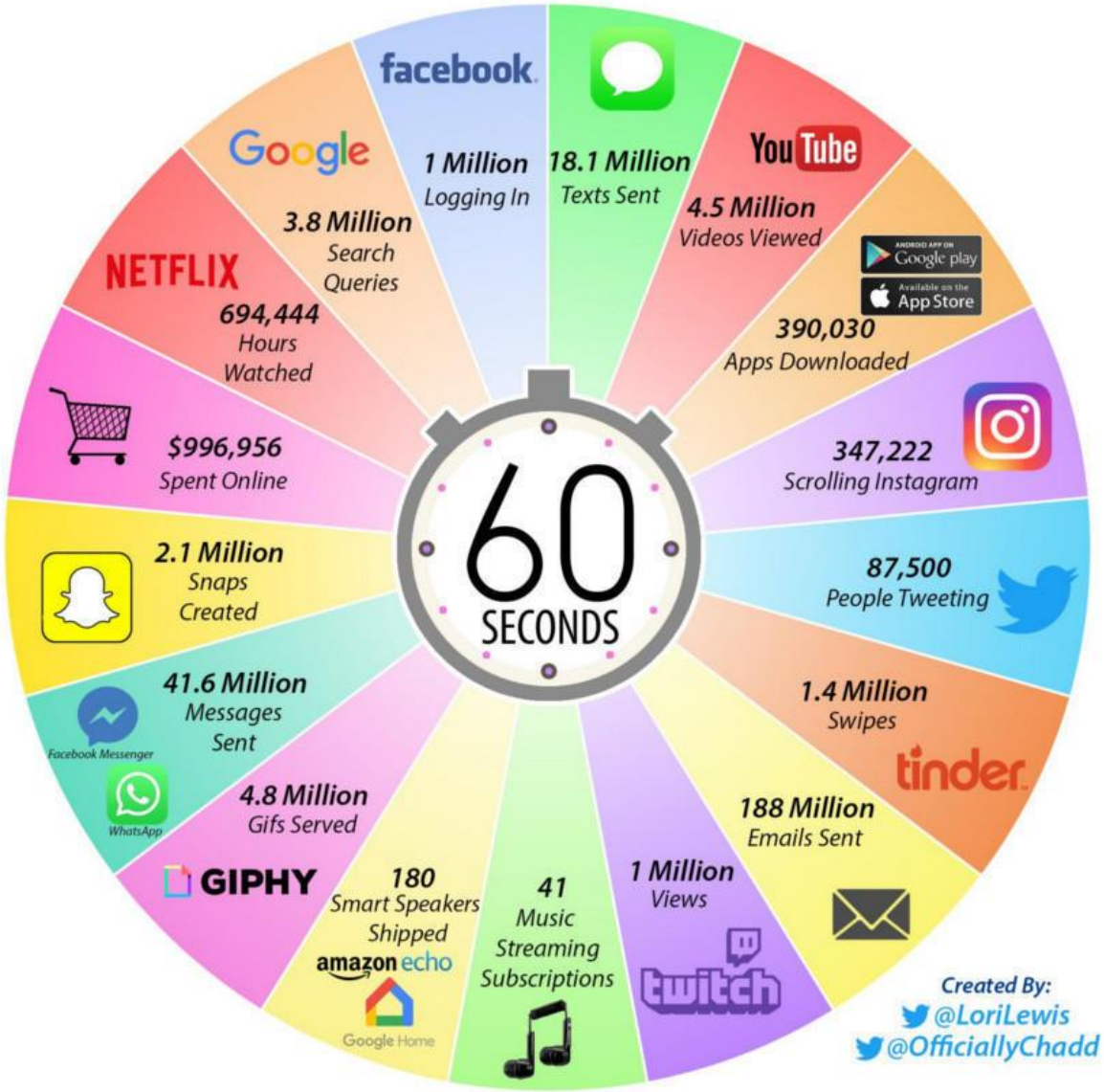
Agenda

- To learn about best practices while implementing
  - Web system
  - CLI
  - Application
  - DBs

Security is Important

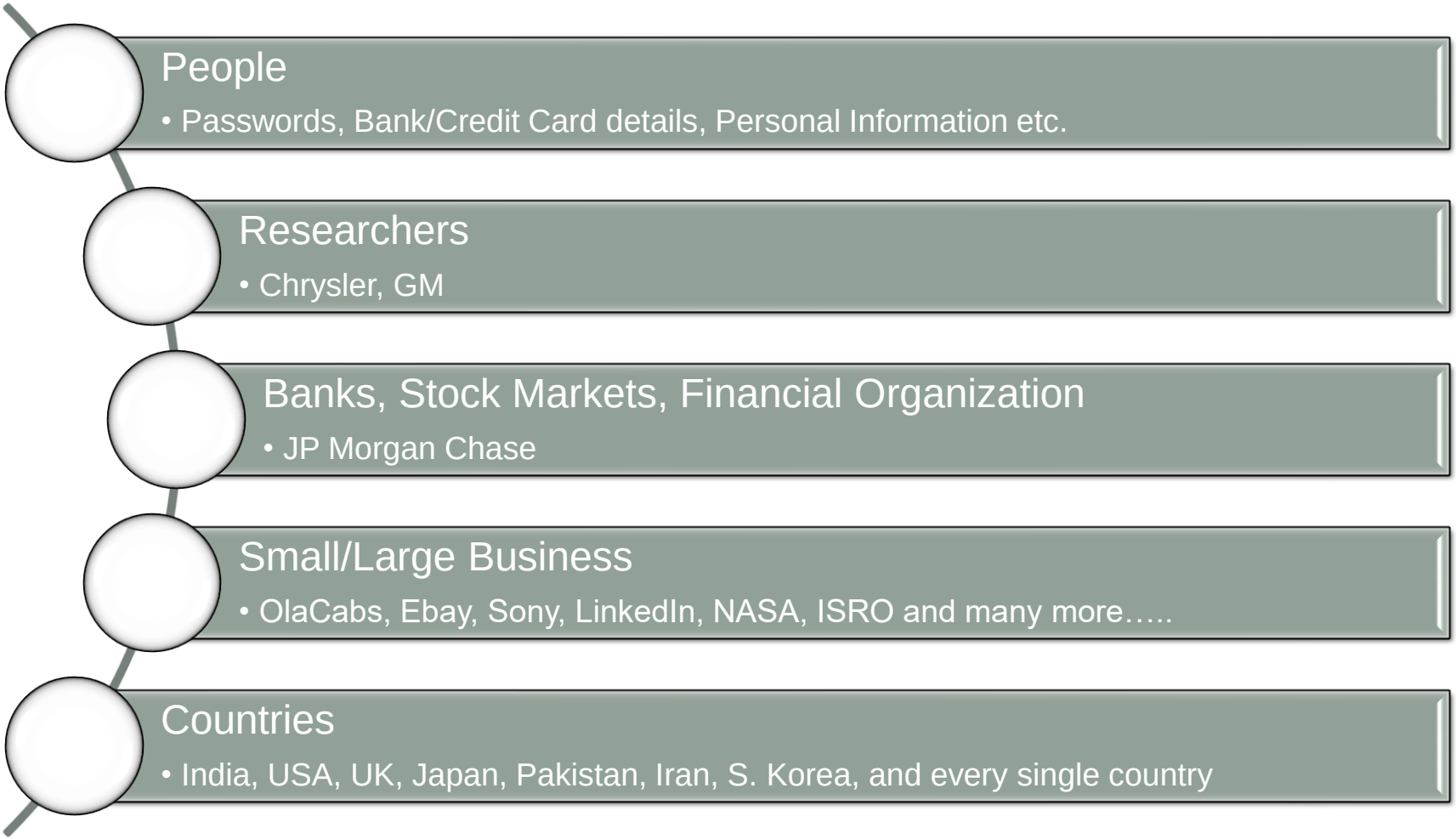
Anyone disagree??

2019 This Is What Happens In An Internet Minute



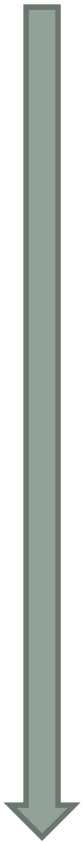
What is at Stake

Stakeholders



Attackers

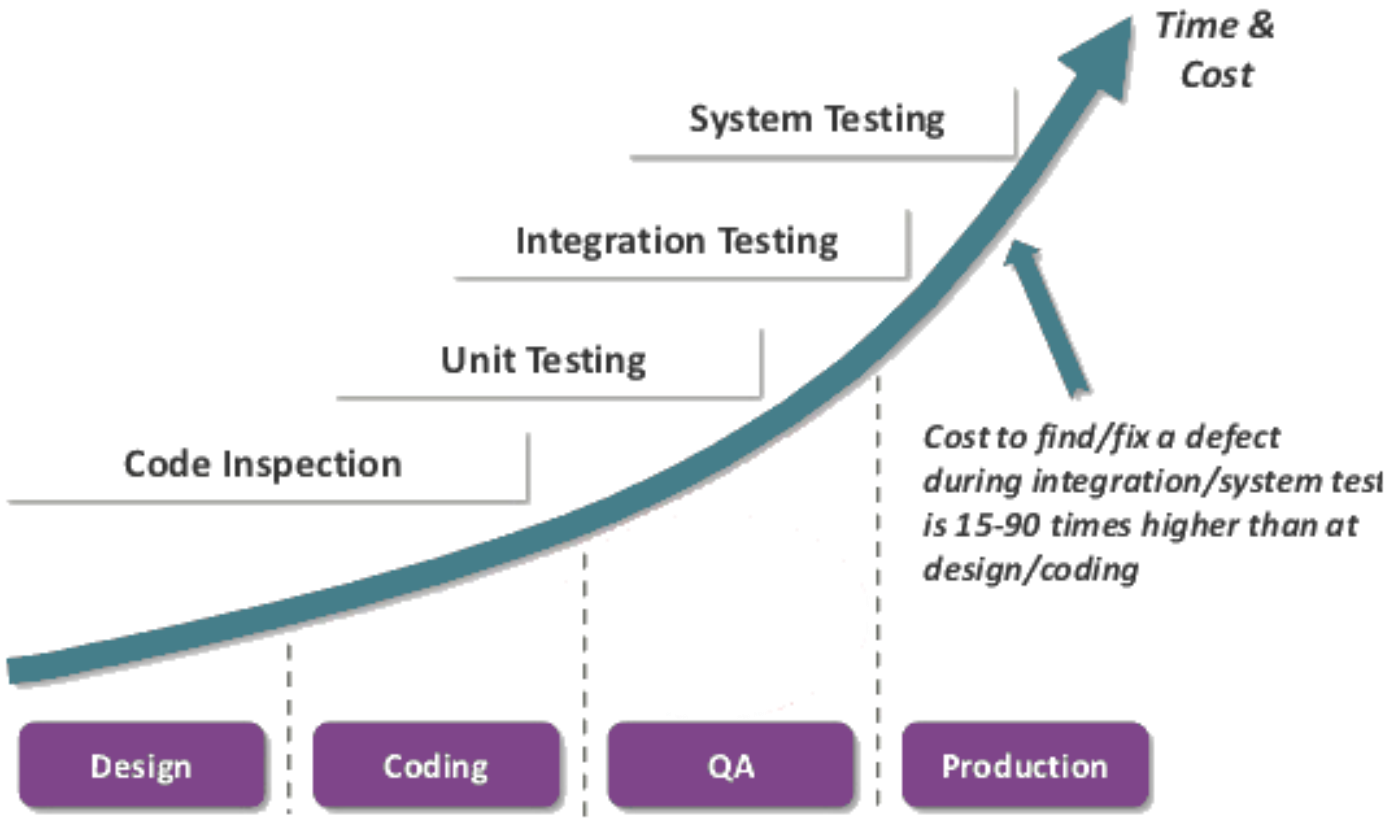
- Trusted Insiders
- Hackers/Hactivist
- Terrorist and Extremist Groups
- Industrial Spies and Organized Crime Groups
- Nation States



Less Structured
Less Skilled

Highly Structured
Highly Skilled

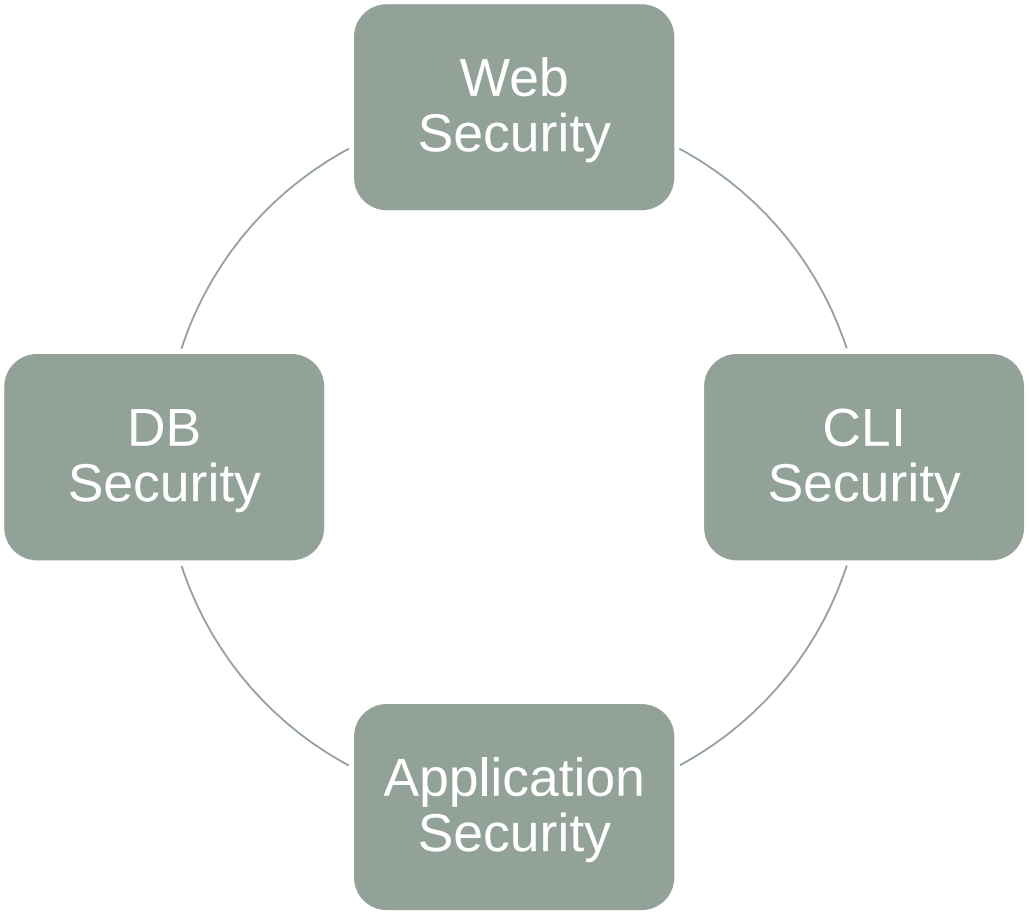
Prevention is Better Than Cure



Source: https://www.checkmarx.com

Our Goal

Make software secure and robust



Best Practices For Web Security

- Change password at the first time use
- Two-Factor authentication and proper authorization
- Store password in hash
- Implement strong password with lock and expiry
- Implement input validation
  - Server-side validation
- Use non-admin/root user (Least Privilege)
- Use safer protocols
- Use well known strong algorithms
- Implement file upload checks
- Collect limited information

NOTE: Applicable in other components also

Best Practices For CLI Security

Implement input validation

- Input length validation
- Input type validation
- Input range validation

Implement output validation

- Logs sanitization

Best Practices For Application Security

- Implement defense-in-depth
- Handle exceptions and errors
- Implement secure OS configurations
- Implement encryption of data
- Check/Maintain default values in libraries
- Use secure and latest trusted (3rd party) libraries
- Remove/Disable unnecessary libraries
- Implement fail-secure
- Do not use hard-code credentials

Best Practices For DB Security

Implement access control

- limited privileges
- Check source of connection request
- Limit concurrent connections

Keep check on concurrent connection

- Single source
- Multiple source

Top 25 Weaknesses (CWE)

CWE-89	Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')
CWE-78	Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')
CWE-120	Buffer Copy without Checking Size of Input ('Classic Buffer Overflow')
CWE-79	Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')
CWE-306	Missing Authentication for Critical Function
CWE-862	Missing Authorization
CWE-798	Use of Hard-coded Credentials
CWE-311	Missing Encryption of Sensitive Data
CWE-434	Unrestricted Upload of File with Dangerous Type
CWE-807	Reliance on Untrusted Inputs in a Security Decision
CWE-250	Execution with Unnecessary Privileges
CWE-352	Cross-Site Request Forgery (CSRF)
CWE-22	Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')
CWE-494	Download of Code Without Integrity Check
CWE-863	Incorrect Authorization
CWE-829	Inclusion of Functionality from Untrusted Control Sphere
CWE-732	Incorrect Permission Assignment for Critical Resource
CWE-676	Use of Potentially Dangerous Function
CWE-327	Use of a Broken or Risky Cryptographic Algorithm
CWE-131	Incorrect Calculation of Buffer Size
CWE-307	Improper Restriction of Excessive Authentication Attempts
CWE-601	URL Redirection to Untrusted Site ('Open Redirect')
CWE-134	Uncontrolled Format String
CWE-190	Integer Overflow or Wraparound
CWE-759	Use of a One-Way Hash without a Salt

http://cwe.mitre.org/top25/index.html

HOW DO WE ACHIEVE THIS?

More on Backup Slides

Secure Development Life Cycle

- Different than regular Software Development Life Cycle

- Security is required at every step of development:
  - Requirements (Architectural Risk Analysis, Security Requirement)
  - Design (Security Oriented Design)
  - Implementation (Secure Coding, Code Reviews)
  - Testing/ Quality Assurance (Security Tests, Penetration Testing)

Reality = ytilaeR

- Most of the people are not even aware of it
  - Those who are aware, don’t accept till it happens
  - Lack of resources

- Need time/ resources to create secure software
  - High competition
  - Short time to market
  - Everyone is looking for fast solution aka Agile DevOps

Thank You