Pole Vaulting over Agile Security Pits

Daniel Liber

~whoami

- Current: Security Leader @ CyberArk
  - – Product security
  - – Strategy and process driven
  - – A pain in the insecurity’s a$$

- Past @ multiple places
  - – Consulting, Research, PT

~whereami

- CyberArk
  - Privileged account security
  - Look us up (we're hiring ☺)

www.cyberark.com/

CYBERARK®

~quote

“Sometimes you just have to jump off the cliff without knowing where you will land”

~agenda

- Agile, a reminder
- SDLC and Agile
- Collaboration with R&D for security
- Crunching numbers – Why is this issue so important?

So… Agile?

Individuals and interactions over

processes and tools

Working software over

comprehensive documentation

Customer collaboration over

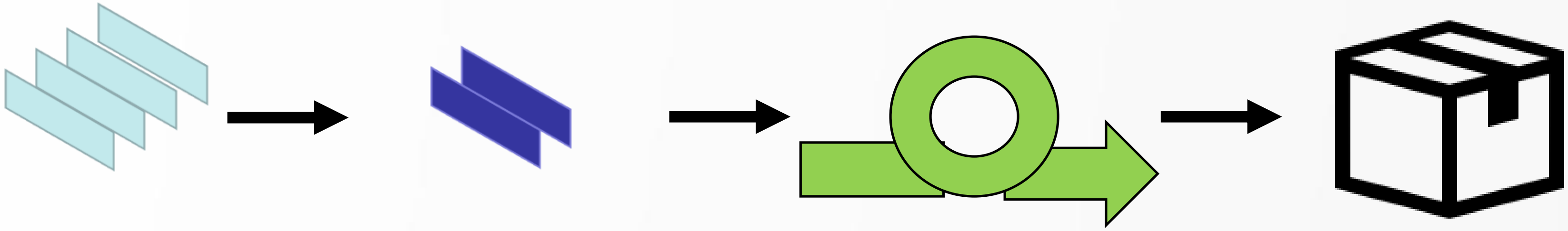
contract negotiation

Responding to change over

following a plan

So… Agile?

Scrum:



Product Backlog

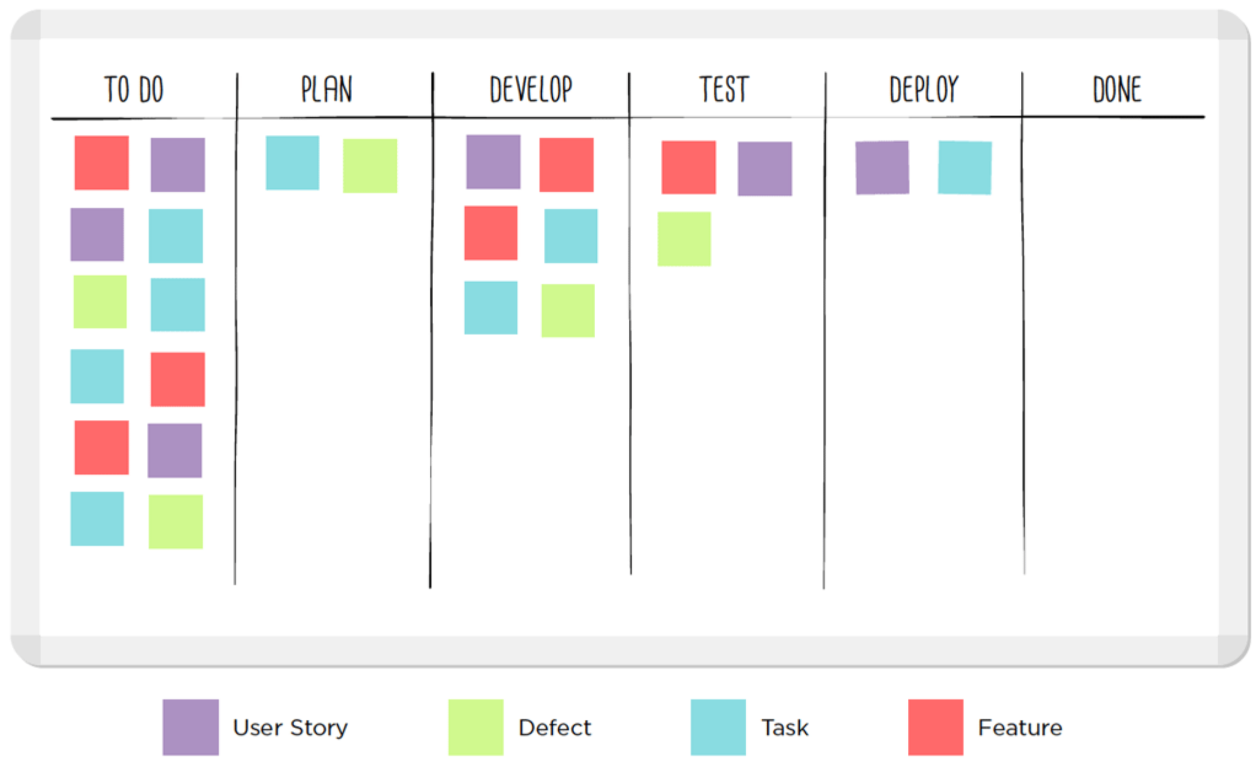
Sprint Backlog

Sprint

Deliverables

So… Agile?

Kanban:



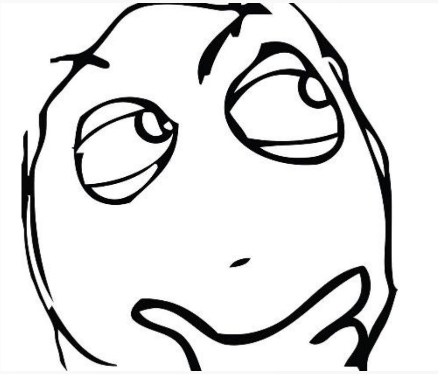
TO DO | PLAN | DEVELOP | TEST | DEPLOY | DONE

🟪 User Story 🟩 Defect 🟦 Task 🟥 Feature

Security Frameworks & Dev

Reflecting on Agile:

“Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.”



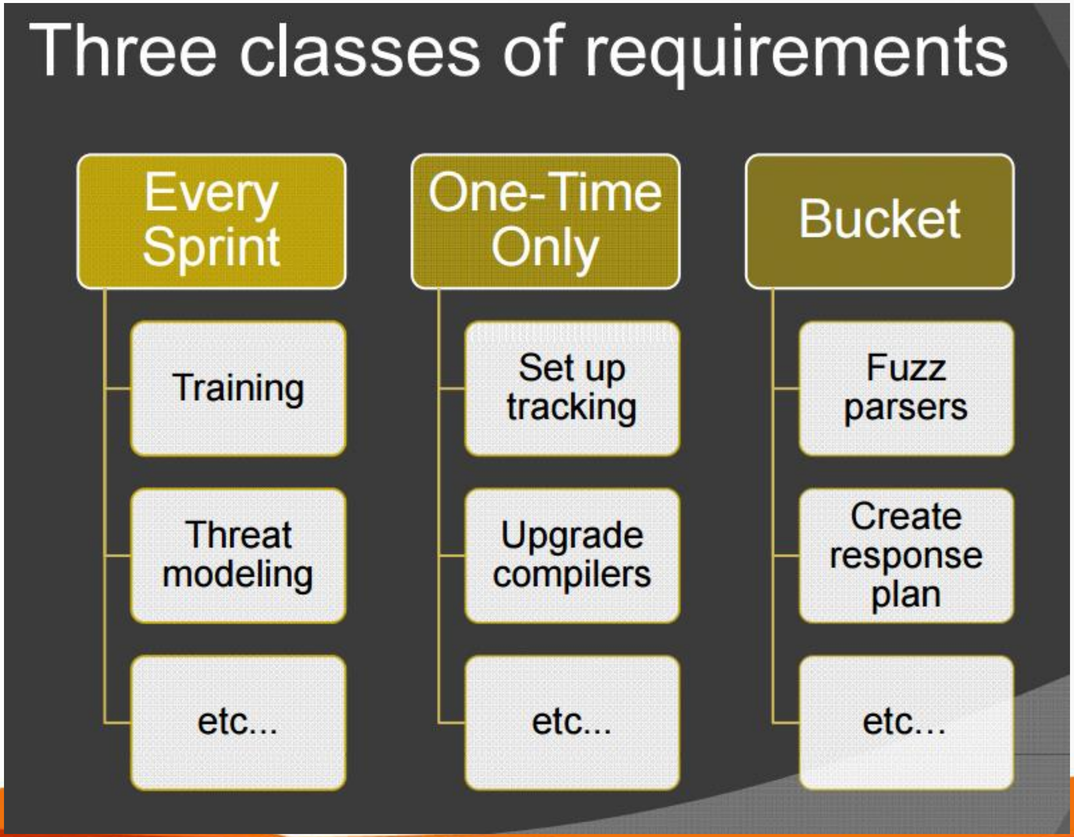
Security Frameworks & Dev

- Vendor SDLC programs
  - Microsoft
  - SAP
  - Cisco
  - Etc..
- Maturity Models
  - OWASP SAMM
  - BSIMM
- NIST

<Compatibility issues>

Security Frameworks & Dev

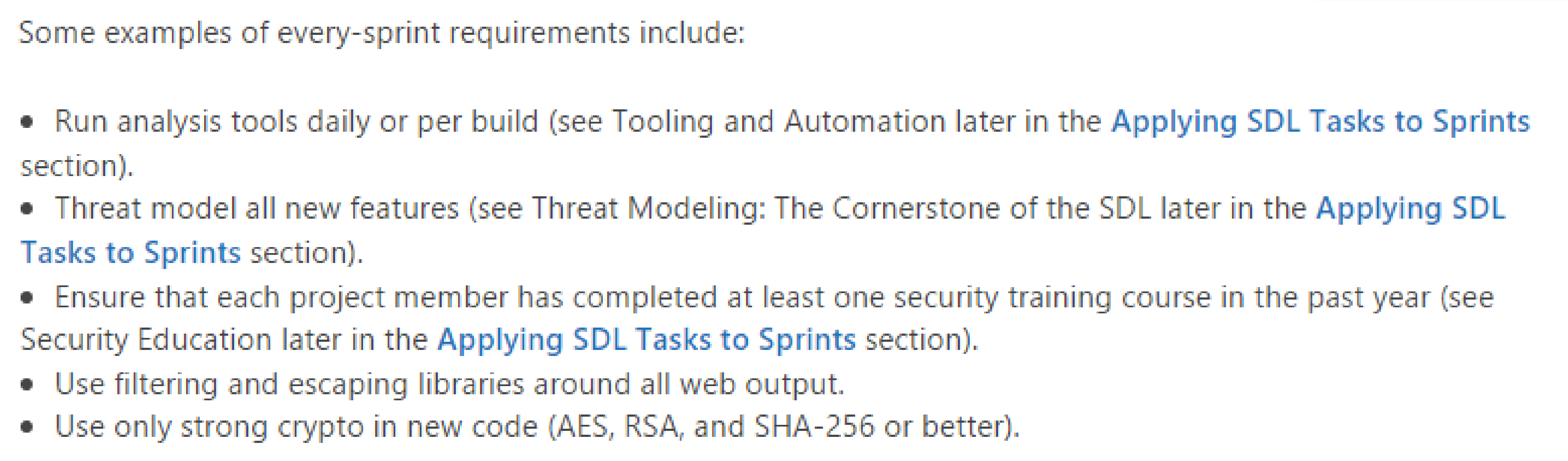
Bryan Sullivan (Microsoft) @ BlackHat 2010



Three classes of requirements

- Every Sprint
  - Training
  - Threat modeling
  - etc...
- One-Time Only
  - Set up tracking
  - Upgrade compilers
  - etc...
- Bucket
  - Fuzz parsers
  - Create response plan
  - etc...

Security Frameworks & Dev



Some examples of every-sprint requirements include:

- Run analysis tools daily or per build (see Tooling and Automation later in the Applying SDL Tasks to Sprints section).
- Threat model all new features (see Threat Modeling: The Cornerstone of the SDL later in the Applying SDL Tasks to Sprints section).
- Ensure that each project member has completed at least one security training course in the past year (see Security Education later in the Applying SDL Tasks to Sprints section).
- Use filtering and escaping libraries around all web output.
- Use only strong crypto in new code (AES, RSA, and SHA-256 or better).

(Microsoft SDL for Agile)

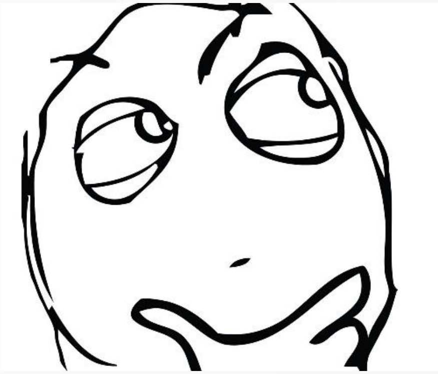
Security Frameworks & Dev

Security Frameworks & Dev

Reflecting on Agile:

“Welcome changing requirements, even late in development.”

→ Threat modeling not only for new features, but also for **CHANGED** features



Security Frameworks & Dev

<u>Threat Modeling</u>

- Approach:
  - Attack / software / asset centric
- Mapping
  - Assets / Actors / Entry points
- Flow
  - Data / Process / Logic

Not as lightweight as expected from a sprint task

Security Frameworks & Dev

<u>Coordinating with Product Owner</u>

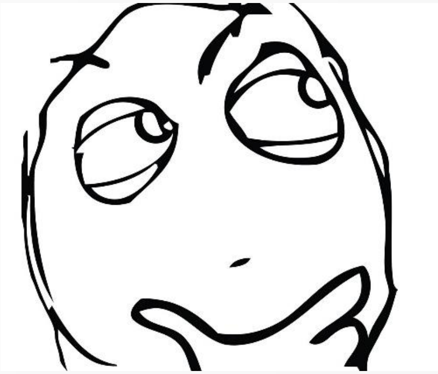
Emperor of the backlog

- Product’s roadmap
- ‘Sensitive’ features attention
- Setting security sprints (bucket security tasks)
- Cut-off for most important threats

Security Collaborations

Reflecting on Agile:

“The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.”



Security Collaboration

Security Collaborations

Pop Quiz

- Sprint of 2 weeks
- Overlooking 4 teams
- Participating in every daily (15 minutes long)

10 days X 4 teams X 15 min.

= 10 hours ~ 1 day = 10% of your time

Security Frameworks & Dev

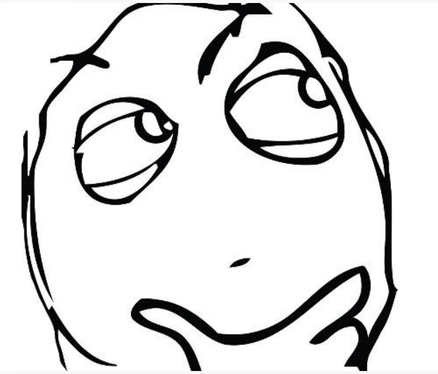
Security Champions

Team’s “security bouncer”

- Why?
  - – Probably knows the product better
  - – Reports back on security aspects
- Who?
  - – Curious, security friendly
- Growth potential – join the dark side

Security Collaborations

Reflecting on Agile:



“The best architectures, requirements, and designs emerge from self-organizing teams.”

→ Teams contain different positions, responsibilities, practices and quite versatile

Security Collaborations

The Team

Team Leader

Developer / Architect

QA

System Analyst

← The Security Guy

Security Collaborations

Customized Training

- Stop using ‘one session fits all’
- Create tracks per position
- Use examples from your products
- Track, certify, re-certify

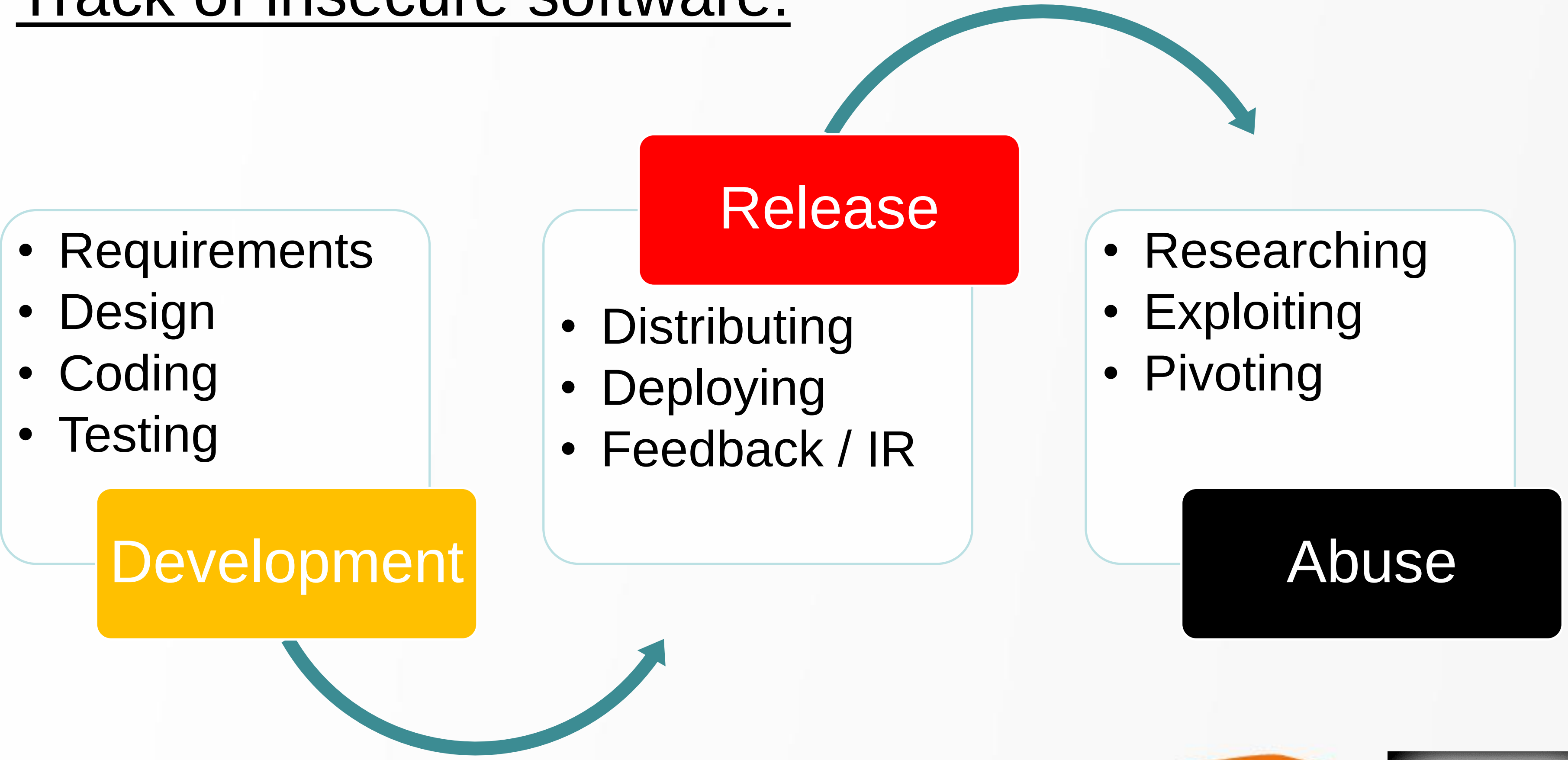
Flexibility in carrying out security tasks

Security Collaborations

Training Name	Developer	Architects	Functional Analyst	Security Team	QA	Team Leaders	PM
Basic Security Training	Yes	Yes	Yes	Yes	Yes	Yes (no test)	Optional
Security Analysis	Optional	Optional	Yes	Yes	Opt.	Opt.	Optional
Secure Design	Optional	Yes	Optional	Yes	Opt.	Opt.	Optional
Secure Development	Yes	Yes	Optional	Yes	Opt.	Yes (no test)	Optional
Security Testing	Optional	Optional	Optional	Yes	Yes	Opt.	Optional
Adv. Security Testing	Optional	Optional	Optional	Yes	Opt.	Opt.	Optional
Risk Management	Optional	Optional	Optional	Yes	Opt.	Yes (no test)	Yes (no test)

Crunching Numbers

Track of insecure software:



Development

- Requirements
- Design
- Coding
- Testing

Release

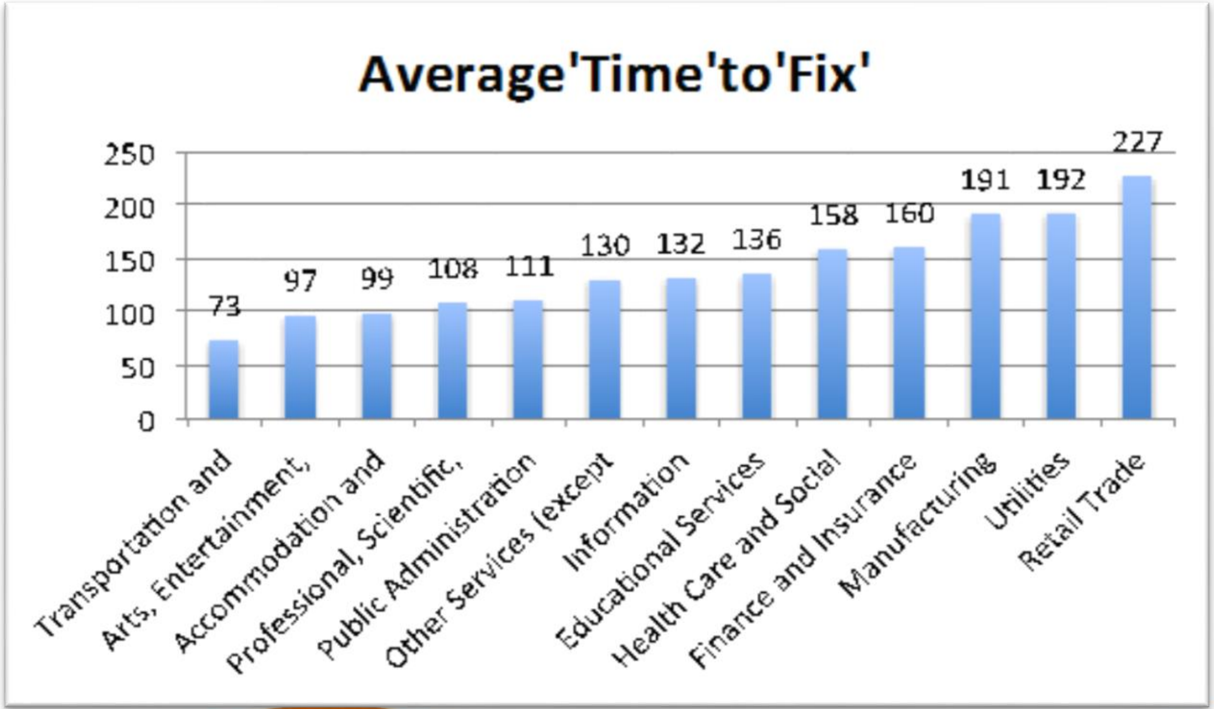
- Distributing
- Deploying
- Feedback / IR

Abuse

- Researching
- Exploiting
- Pivoting

Crunching Numbers

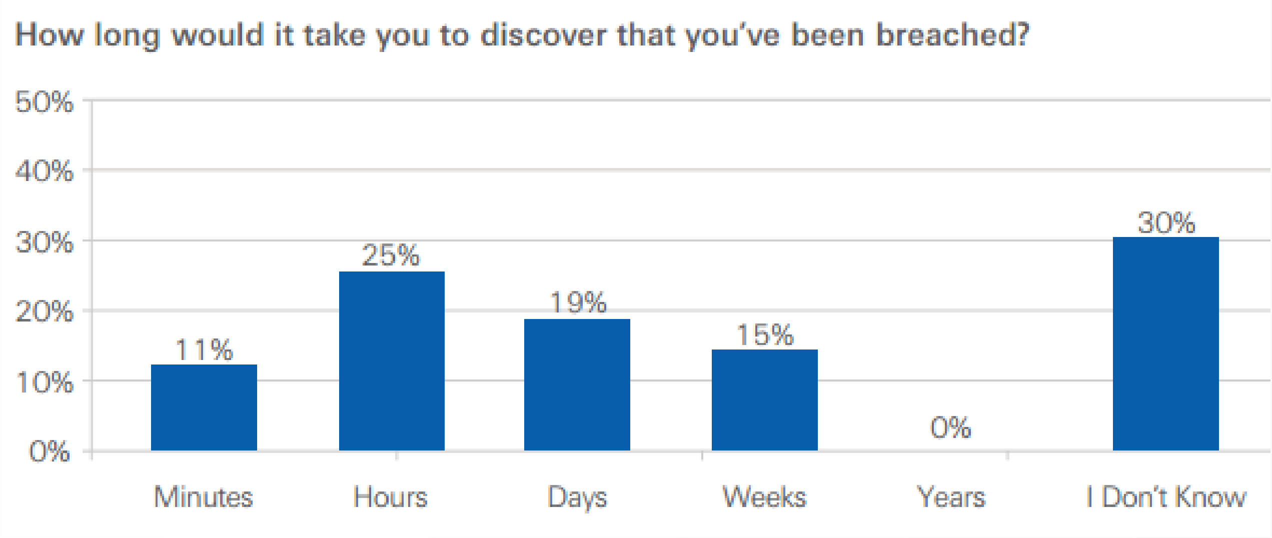
“We will fix it post release!”



Jeremiah Grossman
WhiteHat Security
AppSec Israel 2015

Crunching Numbers

“Ok. BUT – if our software causes a breach, the customer will surely detect it.”

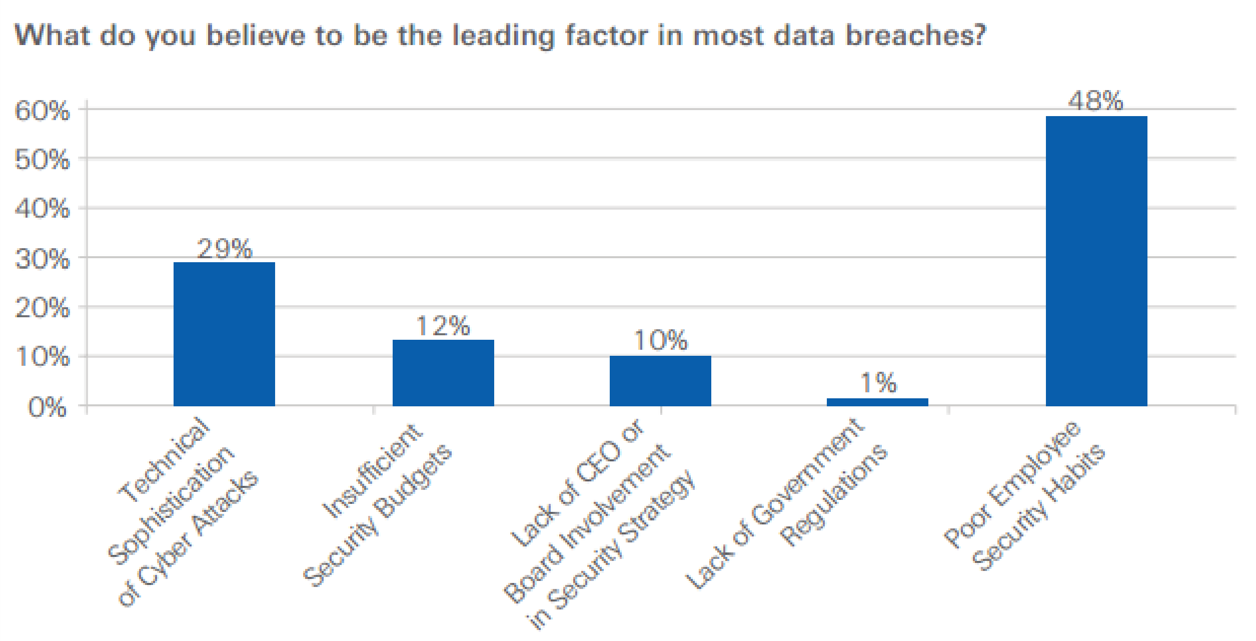


Global Advanced Threat Landscape Survey

CyberArk 2015

Crunching Numbers

“I’m sure that there are other factors for a breach than bad practices of development and deployment”

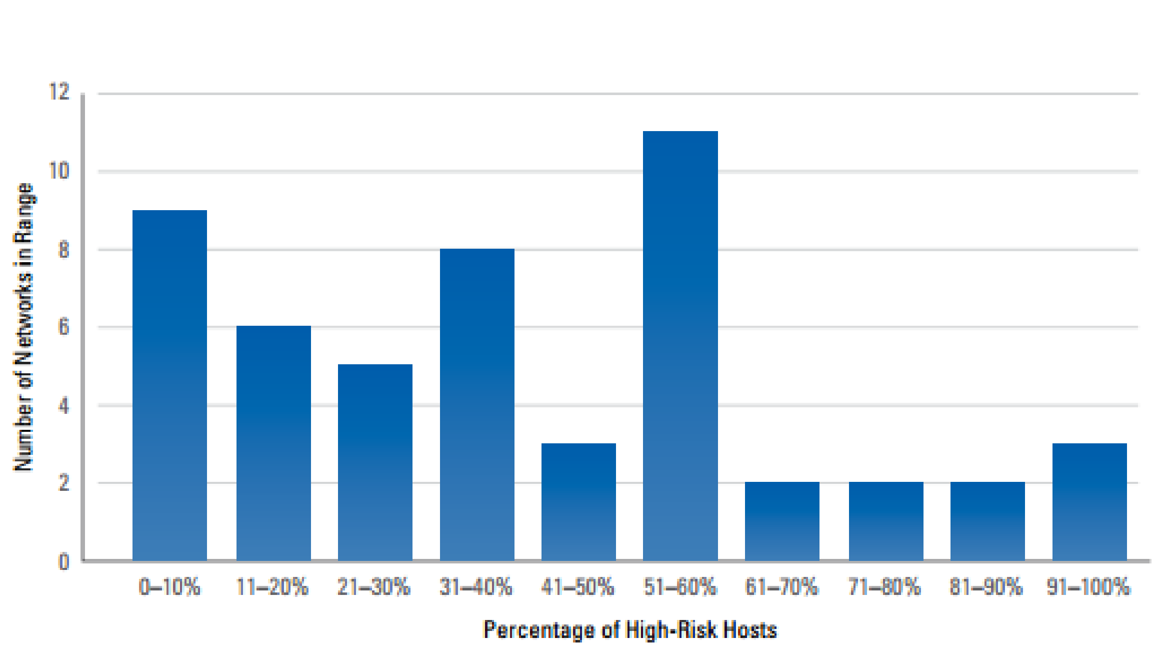

What do you believe to be the leading factor in most data breaches?

Global Advanced Threat Landscape Survey

CyberArk 2015

Crunching Numbers

“It doesn’t matter as a lot of companies secure their networks anyways against breaches”

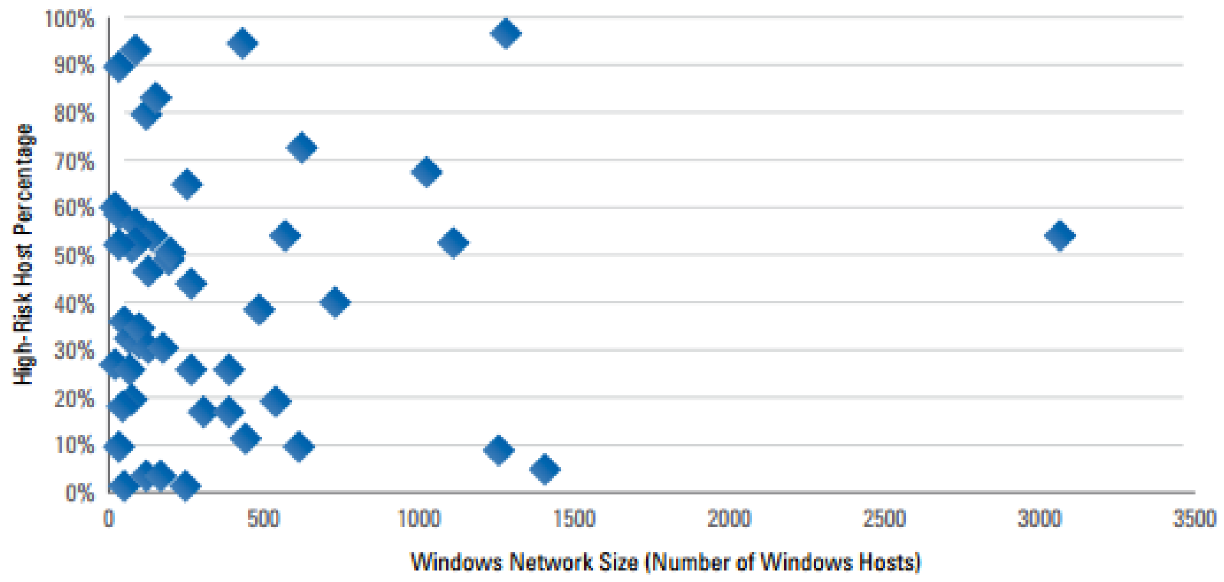


Analyzing Real-World Exposure to Windows Credential Theft Attacks

CyberArk Labs 2015

Crunching Numbers

(Size does not matter, in this case.)



Analyzing Real-World Exposure to Windows Credential Theft Attacks

CyberArk Labs 2015

Conclusions

- Agile is a modern methodology for software development which is commonly used
  - In theory – security could be integrated
  - In practice – there are some glitches
- Don’t be afraid to adjust (use the ★ in this ppt)
- There is a long chain of product security
  - SDLC is first in line
  - You really don’t want to experience security incident down the chain

Questions?

Thank you!

Daniel Liber

✉ Daniel.Liber@CyberArk.com

in https://il.linkedin.com/in/liberdaniel

CyberArk

CYBERARK http://www.cyberark.com/