Scanning Before Pressing Play

Phil Fenstermacher
pcfens@wm.edu

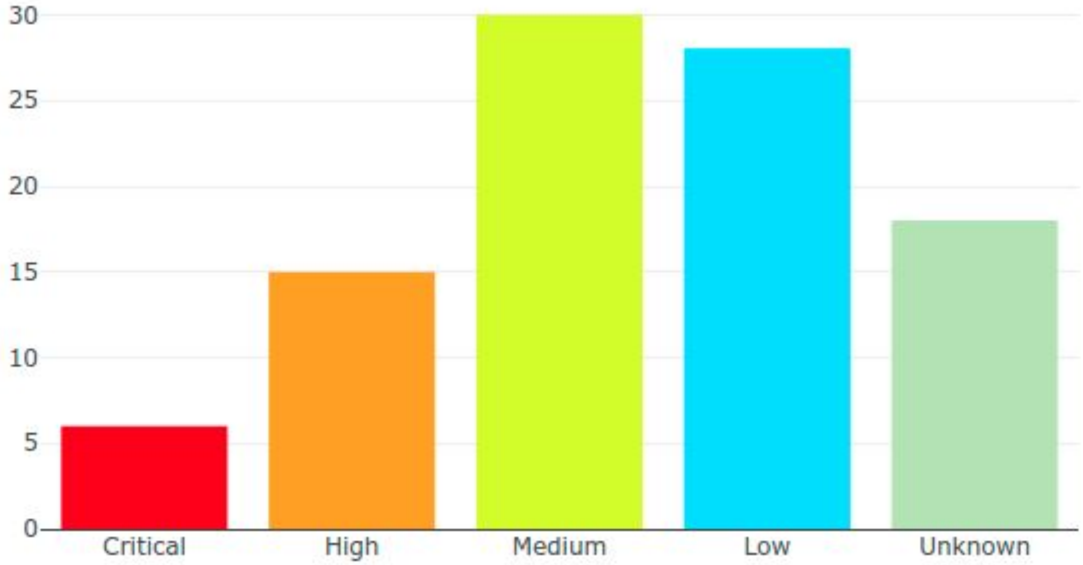
WILLIAM & MARY
CHARTERED 1693

The First Problem

Programmer throws application over the wall to sysadmins.

Operations team builds server and deploys the application, assuming all is well.

Security team runs a routine scan of the *servers*.


30
25
20
15
10
5
0
Critical
High
Medium
Low
Unknown

Somebody scrambles to mitigate

If we’re lucky.

Left to run a vulnerable system despite that expensive scanner with shiny graphs

Especially if it’s a commercial application

Security once it’s running is too late.

The Next Problem

That vulnerable form handler behind the secure CAS login goes undiscovered.

We (unintentionally) hide vulnerabilities.

Application Scanning

Static (SAST) let’s us scan the source code without running it.

Dynamic (DAST) requires the application to run and we throw things at it.

Static scanning is another type of scan, not a replacement.

SAST Analyzers

- Lots of cheap/free options
- Jumping between analyzers takes some planning

SAST Analyzers

August 2020 Release of GitLab Core (free) supports static analysis of ~18 languages

Demo

Securing the Server

- Scanning is hard
- Utilities unrelated to core application
- Libraries often tied to OS Version
  - Upgrading requires new servers
  - Migration, downtime, etc.
- Have to be online to scan

Securing the ~~Server~~ Container

- Smaller footprint
- Upgrade libraries independent of OS
- File and process isolation
- Can be scanned without running

Harbor

Place to store and scan artifacts popular in a container world (images, application bundles, helm charts, etc.)

Demo

Questions?

github.com/pcfens/vascan2020

pcfens@{wm.edu,Twitter,etc.}

WILLIAM & MARY

INFORMATION TECHNOLOGY