Will it Blend?

Questions to ask your Application Security Tool Vendor

MERISTEM
INFOSEC

Introduction

- 15 years in Enterprise IT Services doing…not security
- 10 years as an Application Pentester
  - 5 years for a great 50 person company

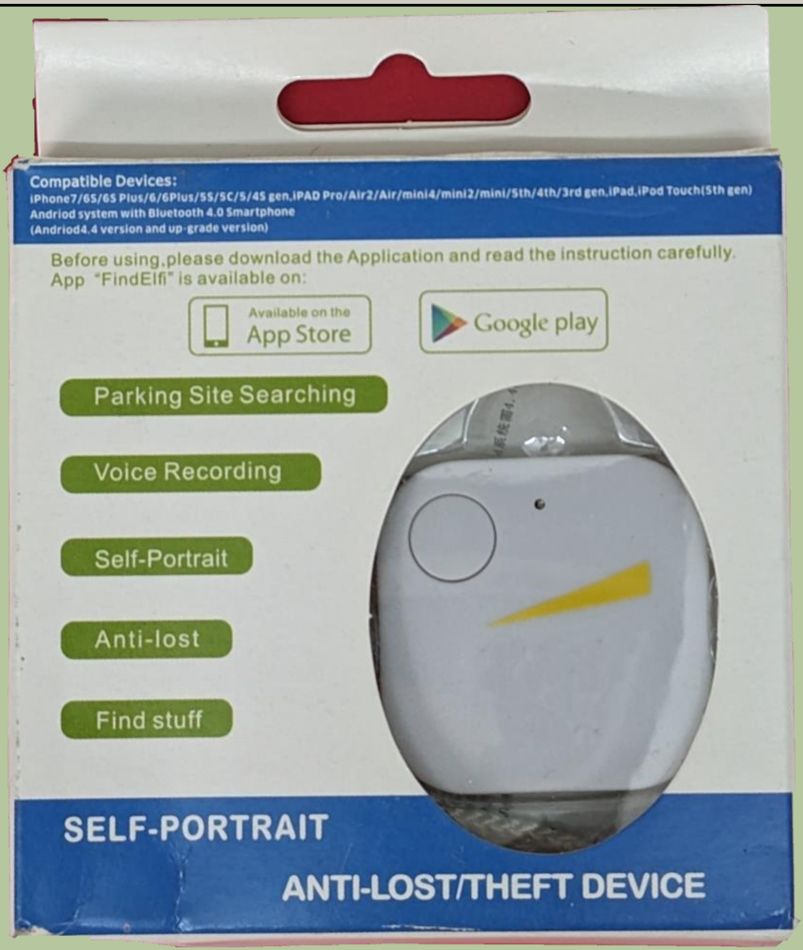
    BOUGHT OUT
  - 9 months at a "big 4" firm
  - 4 months for a great 50 person company

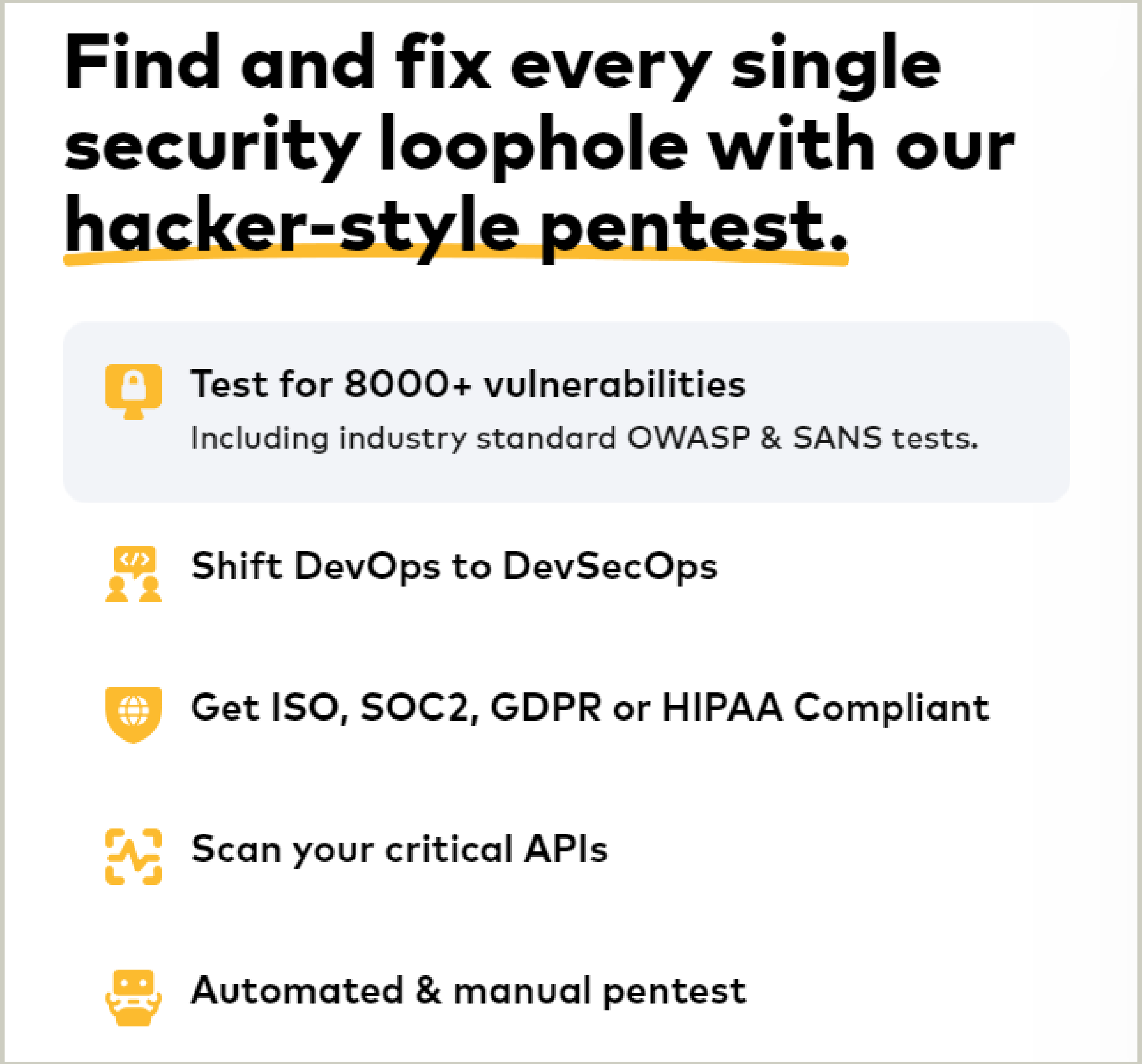
    BOUGHT OUT
  - 3 years at an aspiring "big" firm
  - Started my aspiring great 50 person company

Disclaimer

Background Image by kjpargeter on Freepik

Fool me once…


Compatible Devices:
iPhone7/6S/6S Plus/6/6Plus/5S/5C/5/4S gen,iPAD Pro/Air2/Air/mini4/mini2/mini/5th/4th/3rd gen,iPad,iPod Touch(5th gen)
Andriod system with Bluetooth 4.0 Smartphone
(Andriod4.4 version and up-grade version)
Before using,please download the Application and read the instruction carefully.
App "FindElfi" is available on:
Available on the App Store
Google play
Parking Site Searching
Voice Recording
Self-Portrait
Anti-lost
Find stuff
SELF-PORTRAIT
ANTI-LOST/THEFT DEVICE



Find and fix every single security loophole with our hacker-style pentest.

Test for 8000+ vulnerabilities
Including industry standard OWASP & SANS tests.

Shift DevOps to DevSecOps

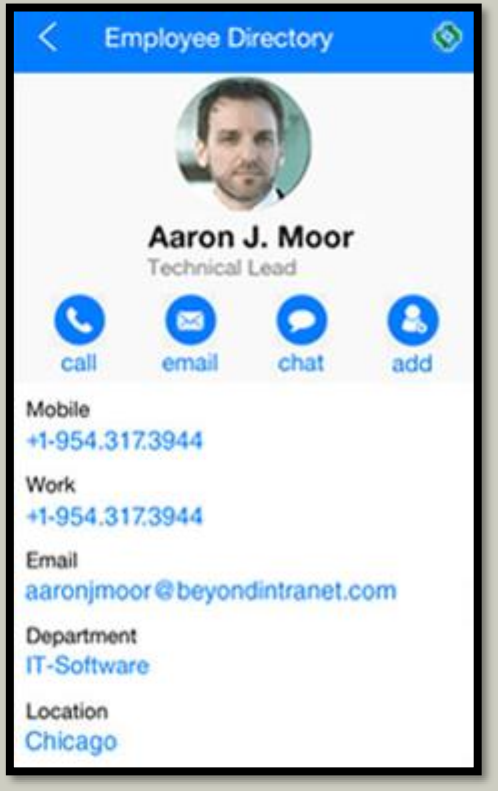
Get ISO, SOC2, GDPR or HIPAA Compliant

Scan your critical APIs

Automated & manual pentest

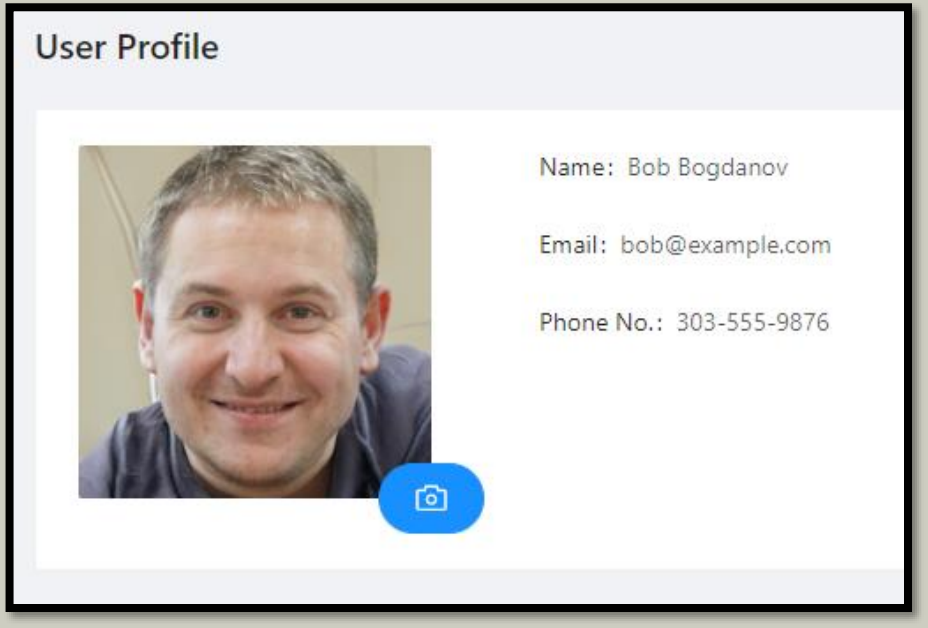
The Impossible Question – Access Control

Should one user be able to access another user's phone number?


Employee Directory

Aaron J. Moor
Technical Lead

Mobile
+1-954.317.3944
Work
+1-954.317.3944
Email
aaronjmoor@beyondintranet.com
Department
IT-Software
Location
Chicago


User Profile

Name: Bob Bogdanov
Email: bob@example.com
Phone No.: 303-555-9876

[redacted]

Vulnerability Scanners / Automated Pentests

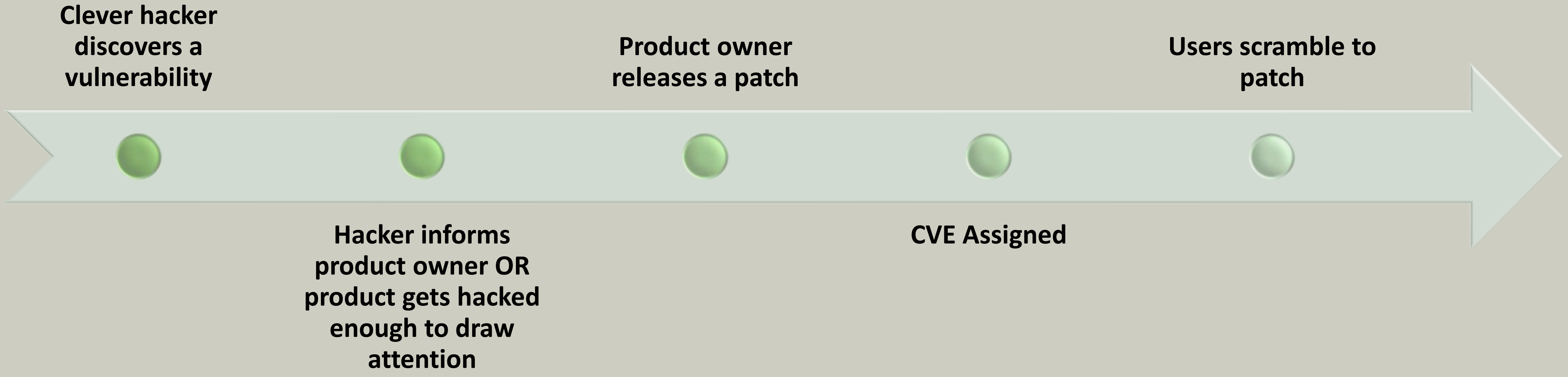
Tools that search a network for active services and report known vulnerabilities in the software version.

1. Start with a port scan
2. Attempt to interact with services on known ports to identify software and versions.
3. Attempt "Safe" exploitation of known vulnerabilities.
4. Report successful exploitation and potential vulnerabilities that are "unsafe" to probe.

<u>Tools in the Family</u>

- Nmap with NSE scripts
- Nessus
- Nikto
- Nexpose
- Greenbone / OpenVAS

Vulnerability Scanners – The Good


Clever hacker discovers a vulnerability
Hacker informs product owner OR product gets hacked enough to draw attention
Product owner releases a patch
CVE Assigned
Users scramble to patch

Defenders must constantly monitor announcements from every product in their environment and/or use an automated tool to monitor new vulnerabilities for them.

Vulnerability Scanners – The Bad

- Vulnerability scanners only detect the vulnerabilities other people have discovered.
  - There is a delay between CVE release and scanner update
    - Can be very short for the best tools
    - Is probably faster than individuals working alone
  - Not intended to defend custom web applications

<u>Tough Questions</u>

- Can the scanner find vulnerabilities in my custom applications?
- How long does it take for a vulnerability to be announced until the scanner's database is updated?
- Does the scanner support authentication with services? How is that configured?

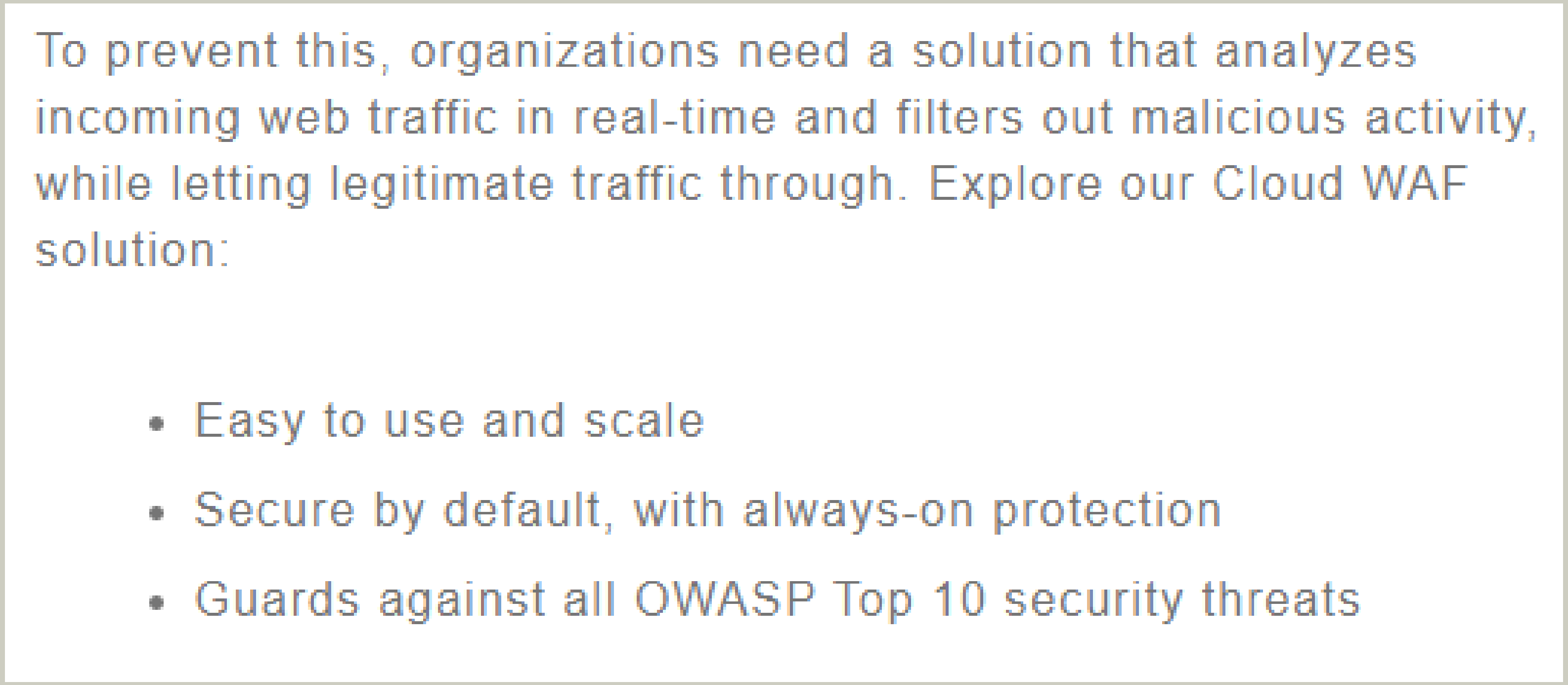
Vuln. Scanner Bonus – Dependency Scanning

Tools that analyze libraries referenced by your source code to see if any published CVEs affect them.

1. Enumerate referenced libraries.
2. Determine the library version.
3. Match the library name and version to known CVEs.
4. Report any CVEs that apply.

<u>Tools in the Family</u>

- Dependency-Check
- Snyk
- Sonatype Nexus
- Black Duck


To prevent this, organizations need a solution that analyzes incoming web traffic in real-time and filters out malicious activity, while letting legitimate traffic through. Explore our Cloud WAF solution:

- Easy to use and scale
- Secure by default, with always-on protection
- Guards against all OWASP Top 10 security threats

Web Application Firewalls

Web Application Firewalls

Tools that monitor network traffic for text patterns that match known attacks.

1. Install inline with application network traffic (post TLS termination)
2. Run in "training" mode and gradually tune rules until legitimate traffic is not blocked.
3. Switch to "blocking" mode to reject any traffic that matches a rule.

<u>Tools in the Family</u>

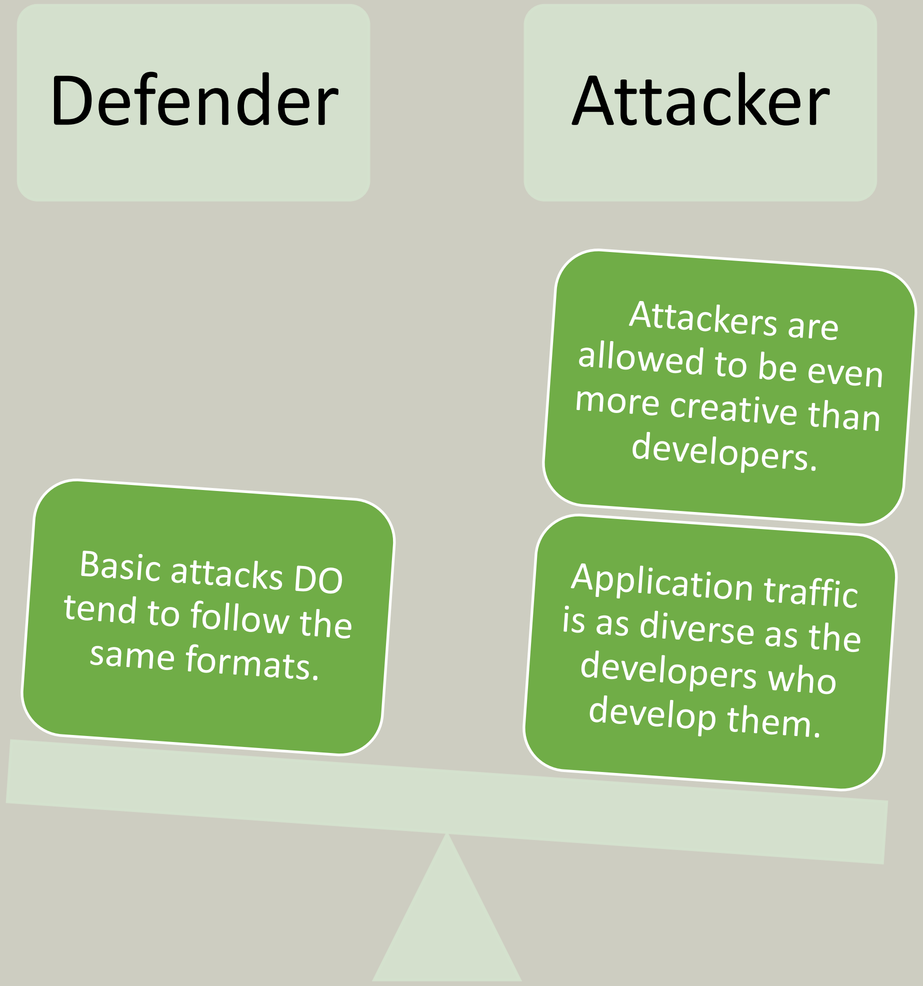
- ModSecurity
- Imperva
- F5 Cloud WAF
- AWS/Azure WAF

OWASP Top 10?

Comprehensive protection for the Open Web Application Security Project (OWASP) top 10 security risks

- **A01 - Broken Access Control**
- **A02 - Cryptographic Failures**
- **A03 - Injection**
- **A04 - Insecure Design**
- **A05 - Security Misconfiguration**
- **A06 - Vulnerable and Outdated Components**
- **A07 - Identification and Authentication Failures**
- **A08 - Software and Data Integrity Failures**
- **A09 - Security Logging and Monitoring Failures**
- **A10 - Server-Side Request Forgery**

WAFs – The Bad



Defender

Attacker

Attackers are allowed to be even more creative than developers.

Basic attacks DO tend to follow the same formats.

Application traffic is as diverse as the developers who develop them.

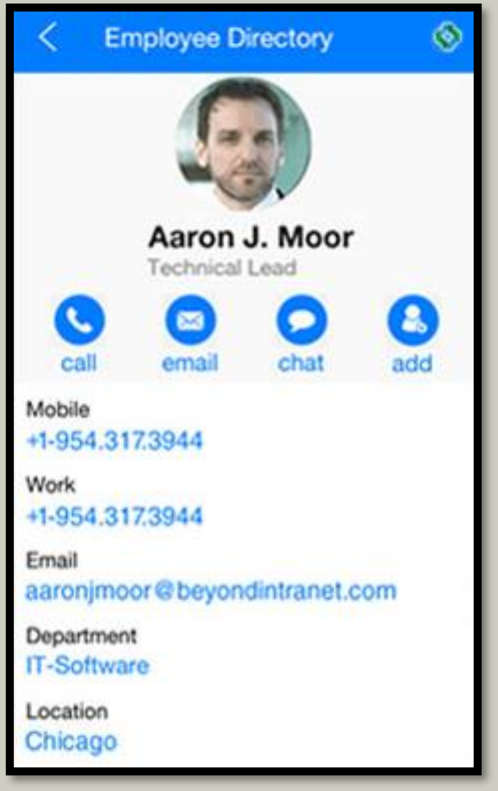
WAFs – The Dream Implementation

Allow List instead of Block List

- Instead of (or in addition to) trying to guess the format of what the attacker will do, define what the application does and reject everything else.
- The data types of most application parameters are known and relatively simple.
- Defining a pattern for every parameter takes time.
- Some parameters (free form text) will require a very permissive pattern and must be backed up by application validation.

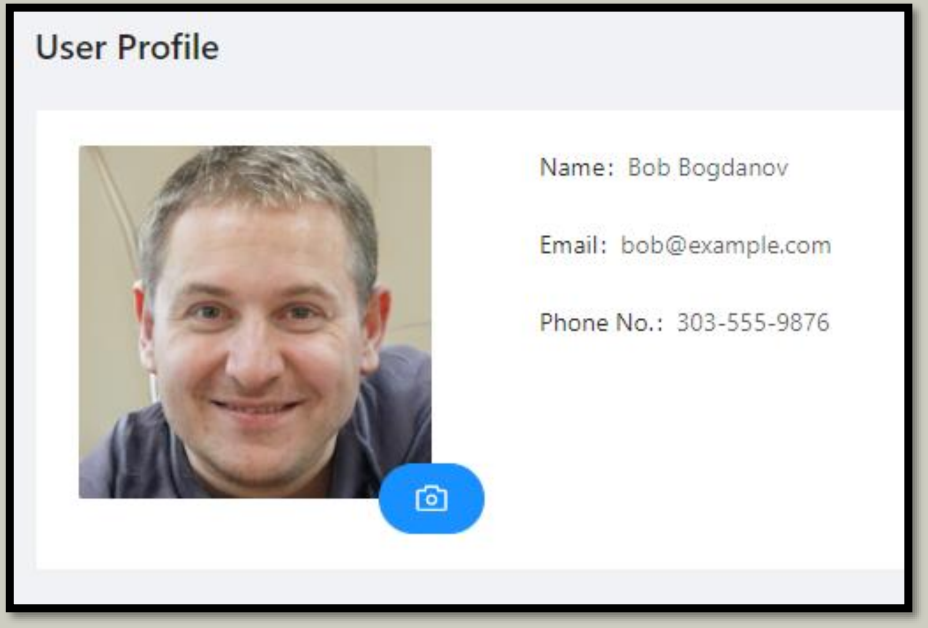
The Impossible Question – Access Control

Should one user be able to access another user's phone number?



Aaron J. Moor
Technical Lead

Mobile
+1-954.317.3944
Work
+1-954.317.3944
Email
aaronjmoor@beyondintranet.com
Department
IT-Software
Location
Chicago



User Profile

Name: Bob Bogdanov
Email: bob@example.com
Phone No.: 303-555-9876

WAFs – What to ask

Tough Questions

- How restrictive are your rule sets out of the box?
  - Won't that block my application traffic OR won't that allow attackers in?
- What tools are available to help me configure the filters?
- How does your product defend against…
  - A01 - Access Control abuse
  - A02 - Cryptography Failures
  - A04 - Insecure Design
  - A07 - Identification and Authentication Failures
  - A09 - Security Logging and Monitoring Failures
  - A10 - Server-Side Request Forgery

Static Application Scanning Tools

Tools that examine the source code of an application to identify security vulnerabilities.

1. The user configures the location of the code base.
2. Processes the code to make an internal model*
3. Run a set of rules against the model looking for vulnerable patterns
4. Report all locations where the pattern was found

Tools in the Family

- ~~grep~~
- SemGrep
- SonarQube
- Checkmarx
- Fortify
- Veracode
- Snyk Code
- Coverity

Speed Running SAST History - grep

```
db.query("SELECT * FROM users WHERE username='"+param[1]+"';")
```

```
grep –i 'query.+SELECT.+param' *
```

```
# Run a query to select the username based on the submitted parameter
```

FALSE POSITIVE

Speed Running SAST History - Syntactic Search

```
findUser = "SELECT * FROM users WHERE username='"+param[1]+"';"
db.query(findUser)
```

```
syntactic-search "method='query', parameter ~= concat(string, var=param)"
```

```
def findUser(un):
      db.query("SELECT * FROM users WHERE username='"+un+"';")

findUser(param[1])
```

FALSE NEGATVIE

Speed Running SAST History - Symbolic Exec.

```
def findUser(un):
      db.query("SELECT * FROM users WHERE username='"+un+"';")

findUser(param[1])
```

1. Parse the code to produce syntactic tokens.
2. Simulate execution with no actual interaction with users, network, file system, etc. (make a lot of assumptions)
3. Track code flow between function calls and across branches.
4. Among other things track where input would have come from (a source) and where it might be dangerous (a sink).
5. Evaluate a wide variety of rules.

SAST – The Good

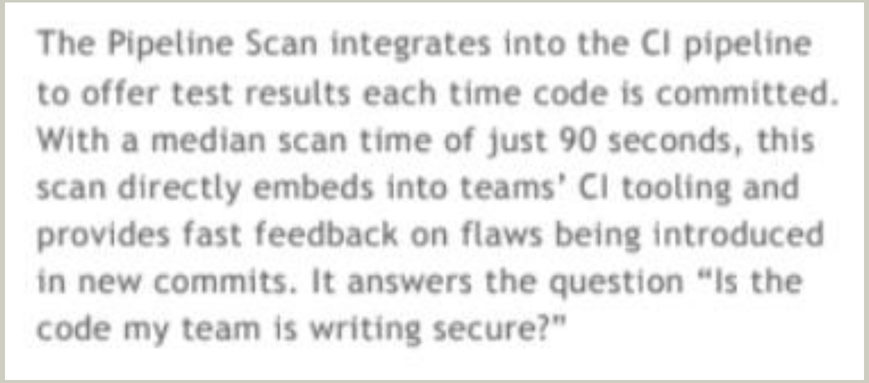
- A01 - Broken Access Control
- **A02 - Cryptographic Failures**
- **A03 - Injection**
- A04 - Insecure Design
- A05 - Security Misconfiguration
- **A06 - Vulnerable and Outdated Components**
- A07 - Identification and Authentication Failures
- **A08 - Software and Data Integrity Failures**
- **A09 - Security Logging and Monitoring Failures**
- **A10 - Server-Side Request Forgery**

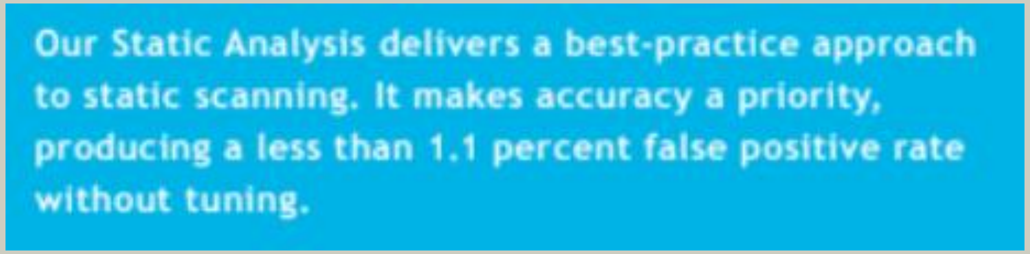
Good results can be achieved with proper tuning

- Identifying input validators
- Managing 3rd Party Libraries
- Gradually enabling rules to manage workload

SAST – The Bad

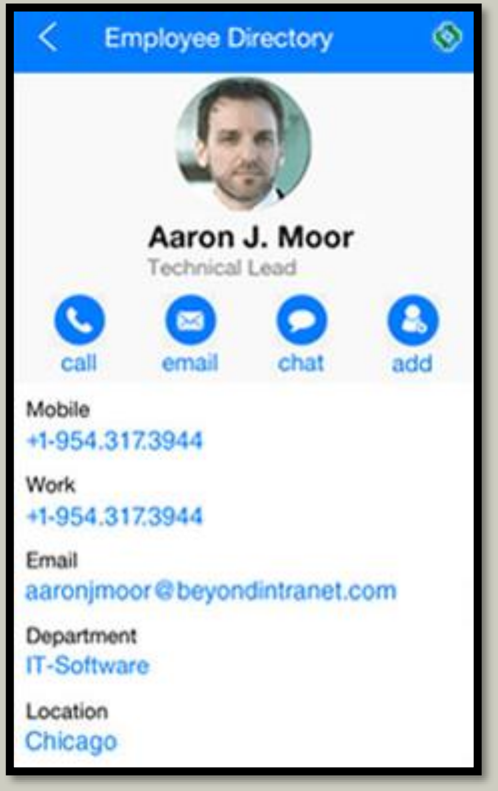
- Long Run Times
- High Numbers of False Positives


The Pipeline Scan integrates into the CI pipeline to offer test results each time code is committed. With a median scan time of just 90 seconds, this scan directly embeds into teams' CI tooling and provides fast feedback on flaws being introduced in new commits. It answers the question "Is the code my team is writing secure?"


Our Static Analysis delivers a best-practice approach to static scanning. It makes accuracy a priority, producing a less than 1.1 percent false positive rate without tuning.

The Impossible Question – Access Control

Should one user be able to access another user's phone number?



Employee Directory

Aaron J. Moor

Technical Lead

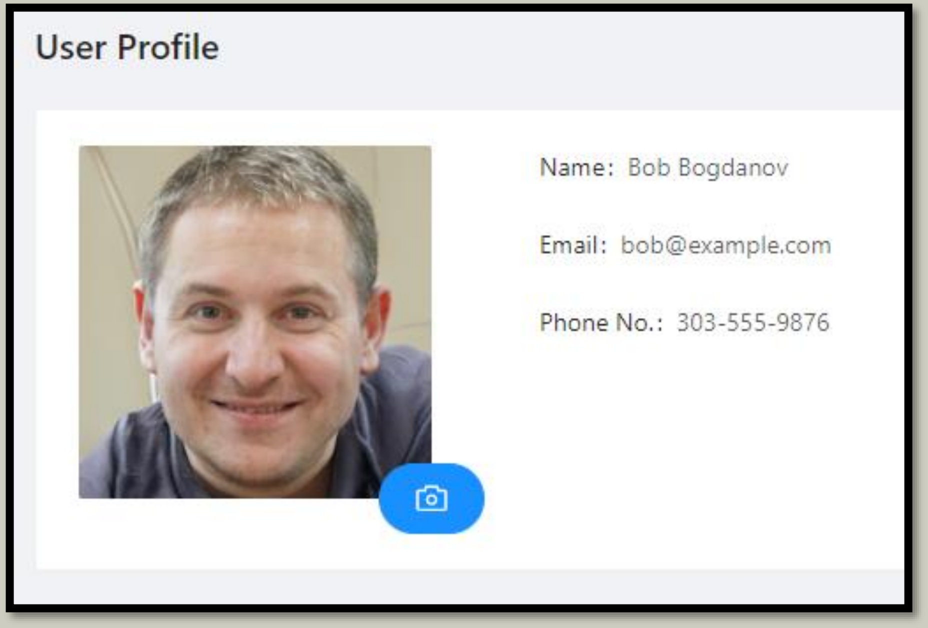
Mobile
+1-954.317.3944

Work
+1-954.317.3944

Email
aaronjmoor@beyondintranet.com

Department
IT-Software

Location
Chicago



User Profile

Name: Bob Bogdanov

Email: bob@example.com

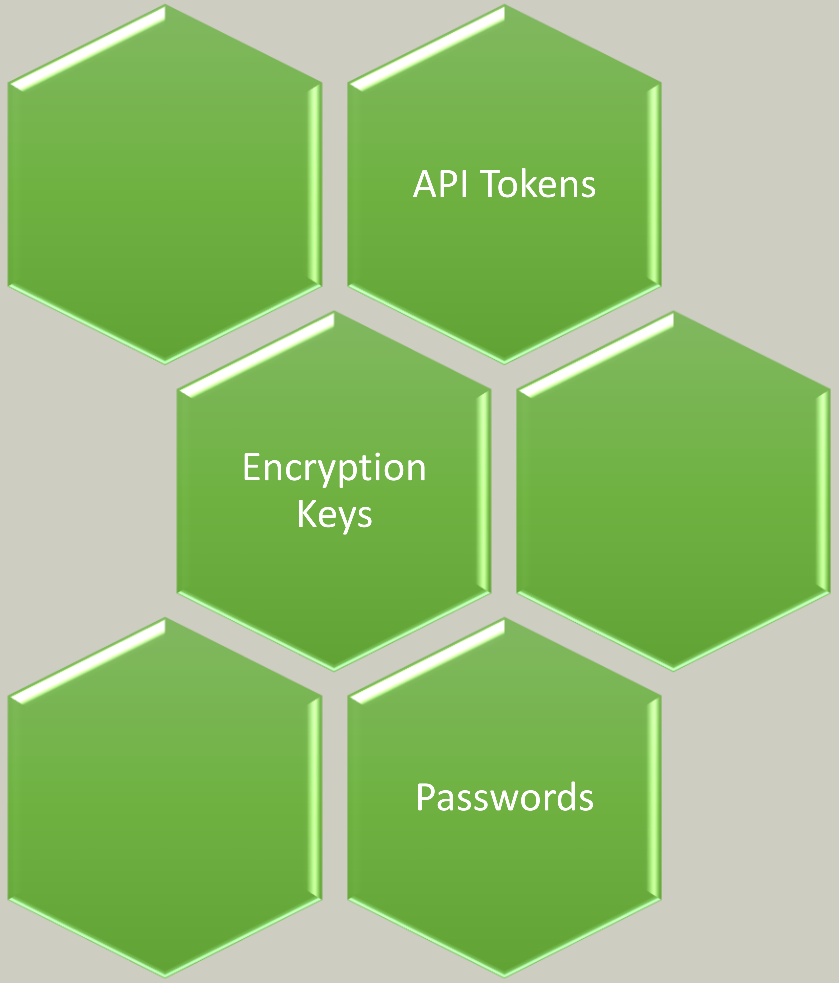
Phone No.: 303-555-9876

SAST – What to ask

Tough Questions

- How does your tool recognize/handle input validation methods?
- How does your tool keep analysis times down?
- How does your tool handle 3rd party libraries?

Static Bonus - Secret Scanning



Dynamic Application Security Tools (DAST)

Tools that simulate a user/browser and navigate a live site while submitting data that triggers or detects vulnerabilities.

1. Provide the scanner a starting URL and optionally credentials.
2. The scanner spiders the site looking for links and input fields.
3. Payloads are submitted where possible.
4. Responses are analyzed for signs of vulnerabilities.

<u>Tools in the Family</u>

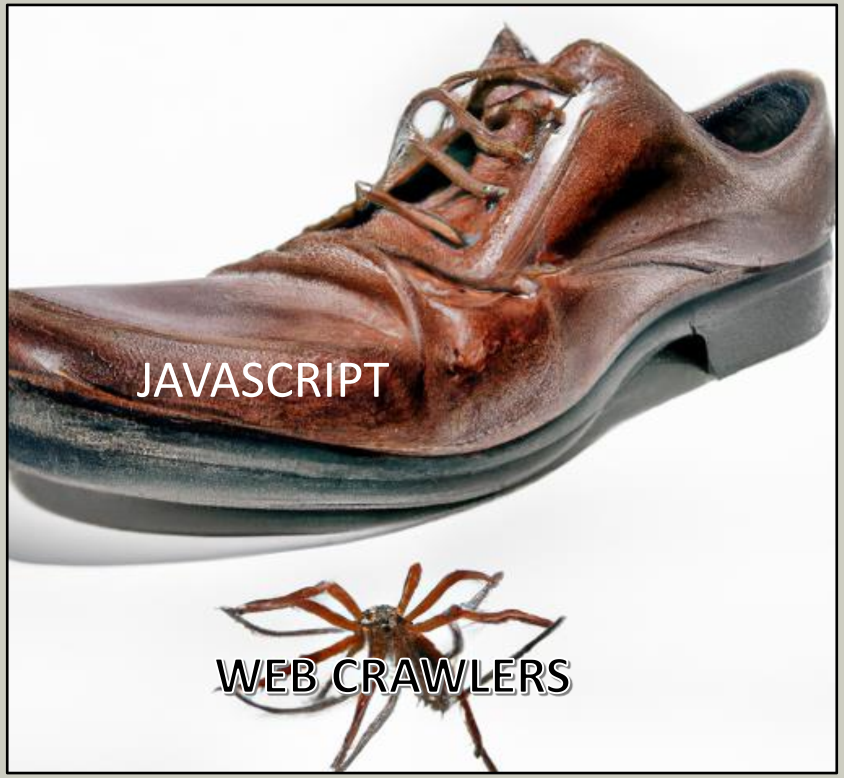
- Acunetix
- Invicti (Netsparker)
- BurpSuite Active Scan
- ZAP Active Scan

DAST – The Good

- A01 - Broken Access Control
- A02 - Cryptographic Failures
- **A03 - Injection**
- A04 - Insecure Design
- **A05 - Security Misconfiguration**
- A06 - Vulnerable and Outdated Components
- **A07 - Identification and Authentication Failures**
- **A08 - Software and Data Integrity Failures**
- A09 - Security Logging and Monitoring Failures
- **A10 - Server-Side Request Forgery**

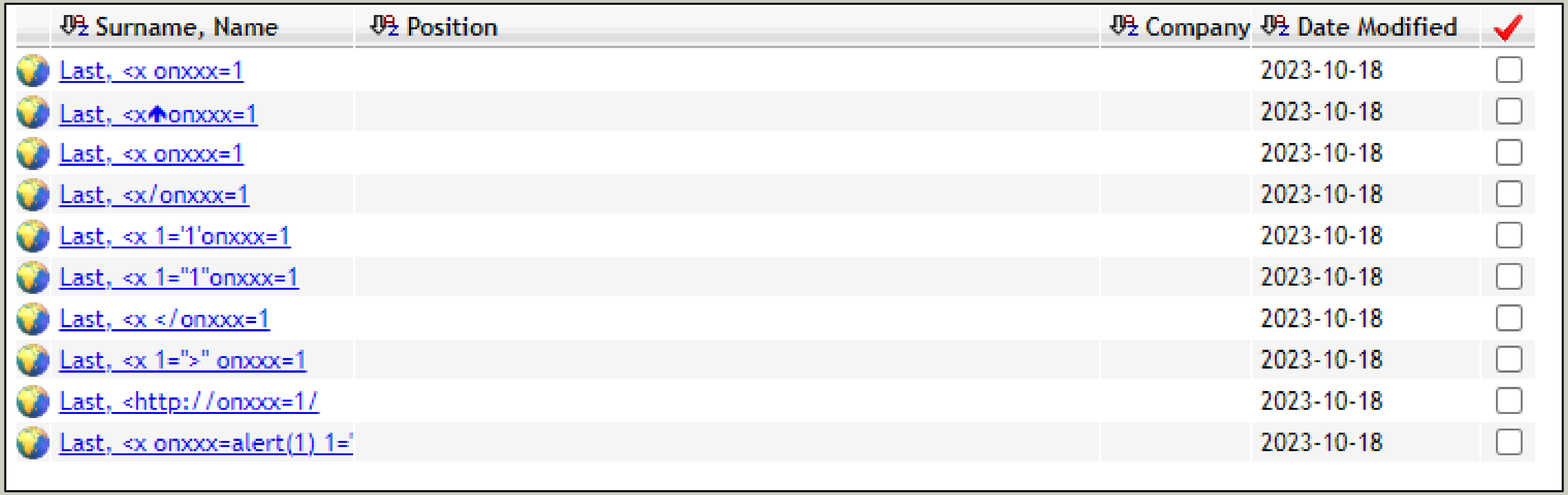
- Code agnostic
- Low rate of false positives
- Less setup/configuration

DAST – The Bad – Site Coverage



Site coverage is increasingly poor.

DAST – The Bad – Site Pollution



Surname, Name	Position	Company	Date Modified	✓
Last, <x onxxx=1			2023-10-18	☐
Last, <x🡅onxxx=1			2023-10-18	☐
Last, <x onxxx=1			2023-10-18	☐
Last, <x/onxxx=1			2023-10-18	☐
Last, <x 1='1'onxxx=1			2023-10-18	☐
Last, <x 1="1"onxxx=1			2023-10-18	☐
Last, <x </onxxx=1			2023-10-18	☐
Last, <x 1=">" onxxx=1			2023-10-18	☐
Last, <http://onxxx=1/			2023-10-18	☐
Last, <x onxxx=alert(1) 1='			2023-10-18	☐

Does your site look like this after a scan?

Why not?

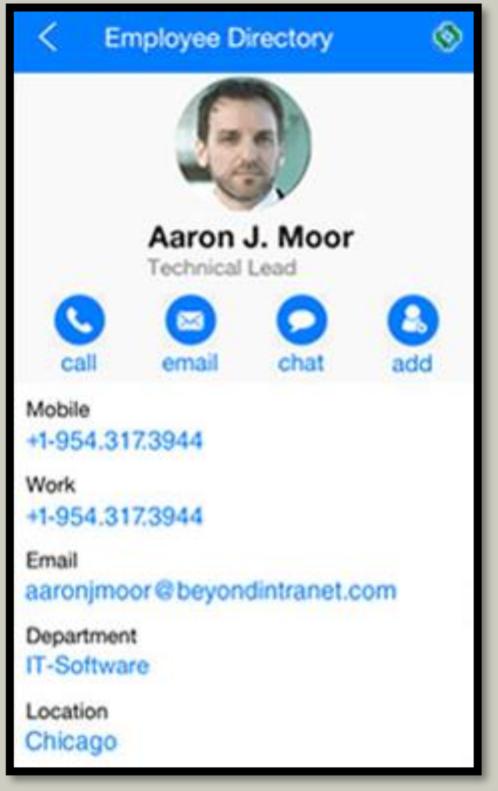
DAST – What to ask

Tough Questions

- How do your tool deal with complex JavaScript user interfaces?
- Does the tool leave behind objects with malicious payloads as attributes?
    - If not: How do you detect stored cross-site scripting?

The Impossible Question – Access Control

Should one user be able to access another user's phone number?



Aaron J. Moor
Technical Lead

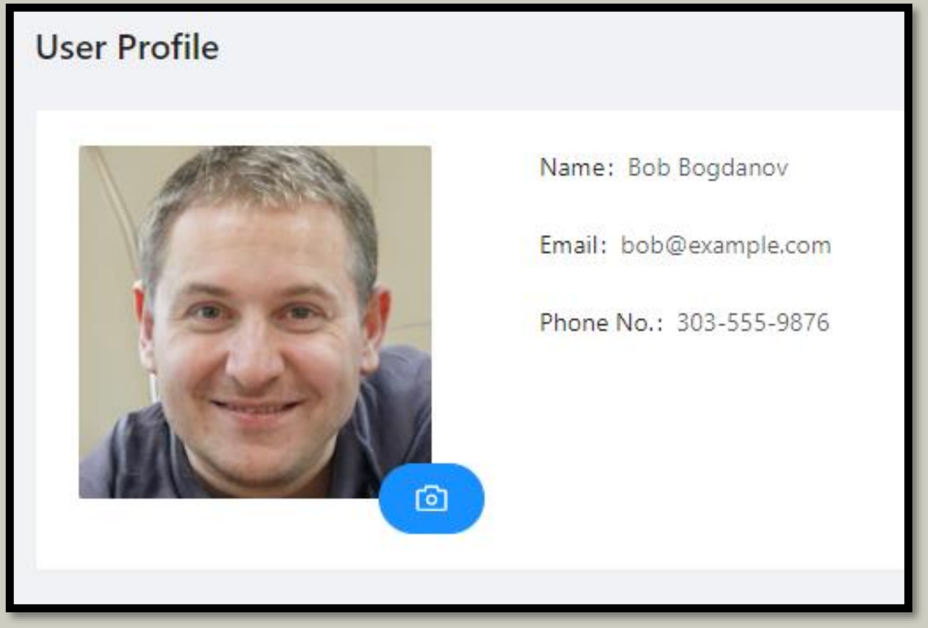
Mobile
+1-954.317.3944

Work
+1-954.317.3944

Email
aaronjmoor@beyondintranet.com

Department
IT-Software

Location
Chicago


User Profile

Name: Bob Bogdanov

Email: bob@example.com

Phone No.: 303-555-9876

DAST Bonus – Interactive AST

Tools that instrument (inject inside) the server process to observe code execution during active use.

1. Use debugging/reflection techniques to add monitoring inside the server process.
2. Exercise the site through normal use, QA scripts, automated browsers.
3. Observe sources, sinks, other vulnerable code handling of requests.

<u>Tools in the Family</u>

- Contrast Assess
- Synopsis Seeker
- Hdiv Detection

Conclusions

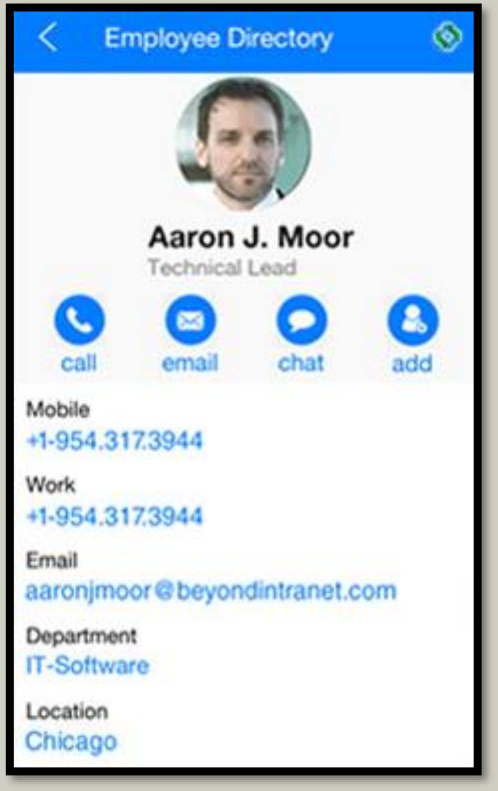
1. Put the tool into the correct family.
2. Think about what is hard for similar tools.
3. Ask the vendor HOW they overcome the difficulties.
4. Challenge flamboyant claims with specific vulnerability types

When in doubt?

How do you test for access control vulnerabilities?

The Impossible Question – Access Control

Should one user be able to access another user's phone number?


Employee Directory

Aaron J. Moor
Technical Lead

call email chat add

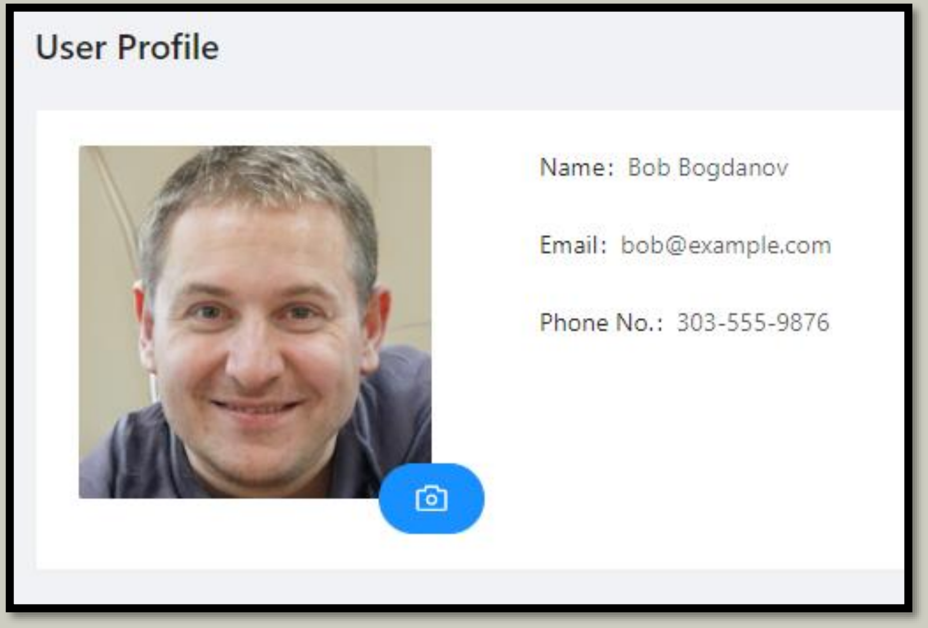
Mobile
+1-954.317.3944

Work
+1-954.317.3944

Email
aaronjmoor@beyondintranet.com

Department
IT-Software

Location
Chicago


User Profile

Name: Bob Bogdanov

Email: bob@example.com

Phone No.: 303-555-9876

Thank You!

Mark Hoopes

mark@meristeminfosec.com

https://www.linkedin.com/in/markhoopes/