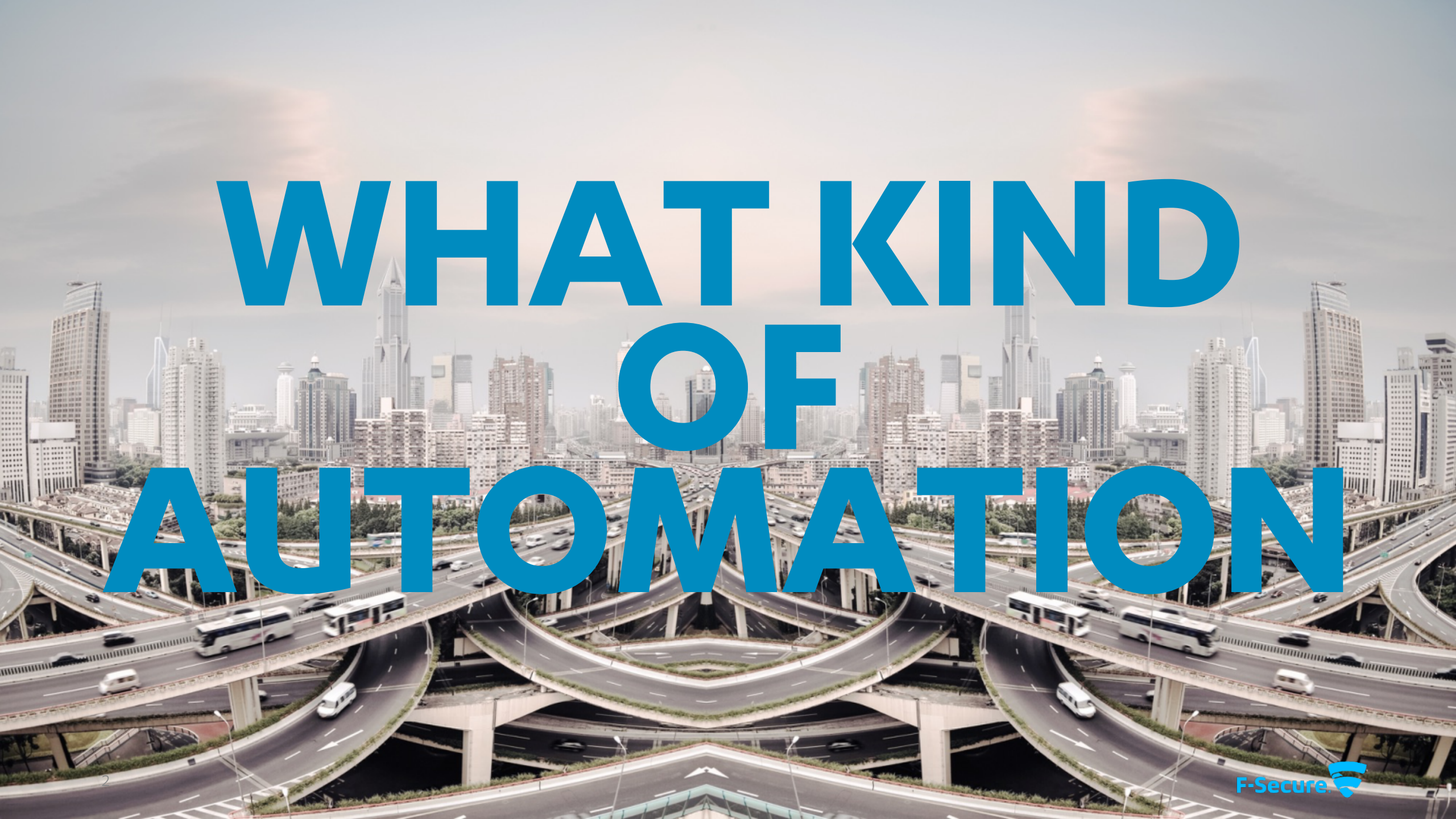
AUTOMATED APPLICATION SECURITY

Juha Kivekäs

WHAT KIND OF AUTOMATION



SECURITY ASSESSMENTS

- Explorative, tester is seldom familiar with application
- Time constrained
- Partly automated, but mainly manual work
- Point-in-time, often just before release
- **Finds issues late in the development lifecycle**

AUTOMATION, TWO VIEWS

Automated functional testing

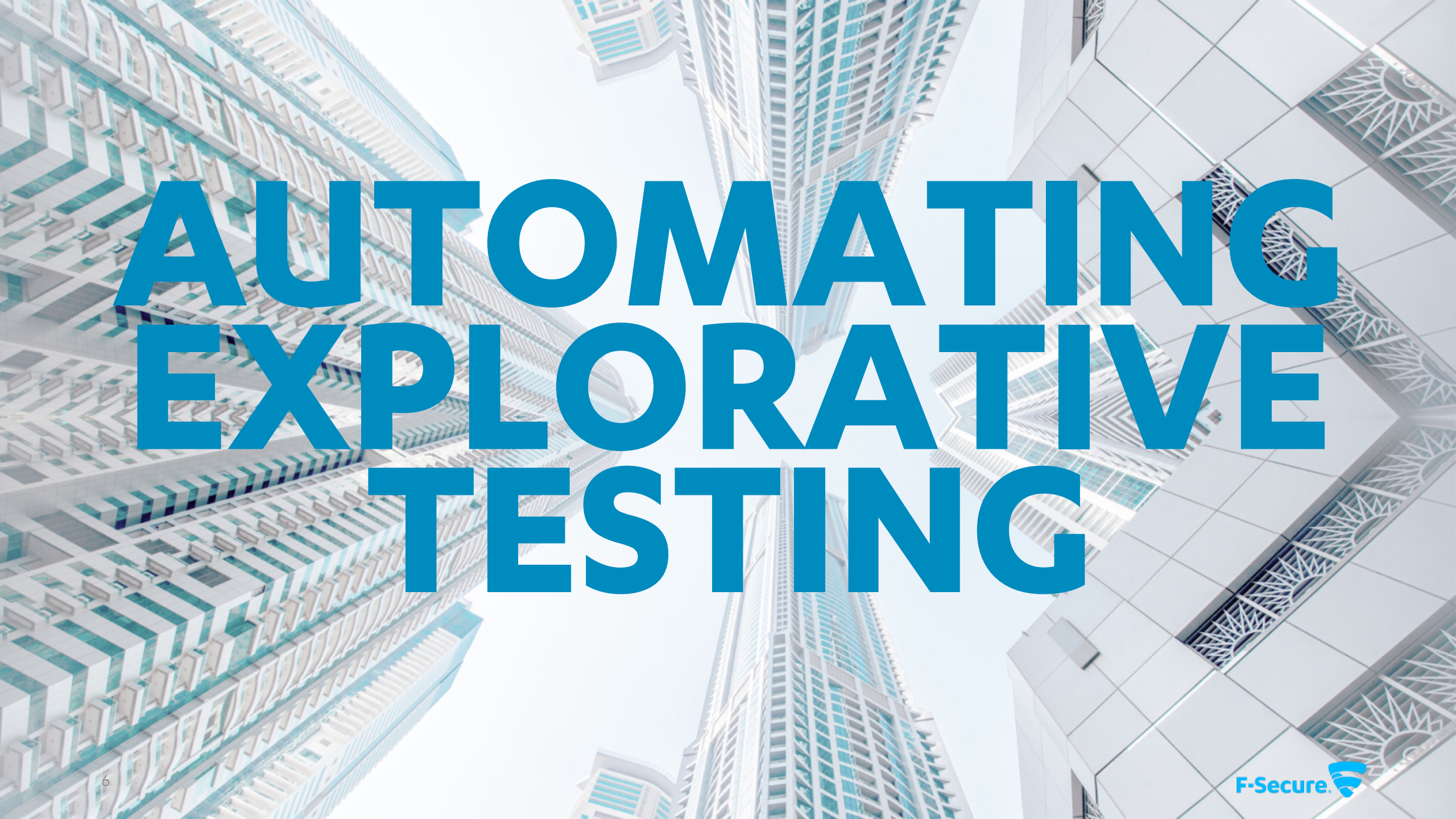
- Reoccurring (weekly, daily, commit)
- Unit tests, Integration tests, etc.
- Triggers

Automated security testing

- Non-manual
- Application scanning
- Network scanning

LEFT SHIFT SECURITY

- Find issues early
- Fix root causes before they propagate
  - Don’t build technical debt
- Bring security awareness to the developers



F-Secure.

USE EXISTING TOOLS

- BSIMM [ST2.1: 22] Integrate black box security tools into the QA process.
- Microsoft SDL Practice #12: Perform Fuzz Testing
- Microsoft SDL Practice #8: Use Approved tools

- Communicates what has been done
- Industry standard
- No detailed security knowledge needed

SECURITY TESTING TOOLS

- Network scanners
- Application scanners
- Proxies
- Code analysis
- Attack and exploitation tools

SCANNING

```
foobar
1'1
1 exec sp_ (or exec xp_)
1 and 1=1
1' and 1=(select count(*) from tablenames); --
1 or 1=1
1' or '1'='1
1or1=1
1'or'1'='1
fake@ema'or'il.nl'='il.nl
```

SPIDER & SCAN

Point and click!

1. Crawl the whole site
2. Test for known bad inputs on everything
    1. Look for typical bad files
    2. Enter known bad inputs
    3. Look for strings in responses (errors, versions, etc.)
3. Done

F-Secure.

SPIDER & SCAN

- Burpsuite Pro, with Carbonator plugin
  - `java -jar -Xmx2g -Djava.awt.headless=true burp.jar https www.example.com 80`
- ZAP quickscan
  - `./zap.sh -cmd -quickurl https://www.exmaple.com -quickprogress`
- Arachni
  - Highly customizable, if you run arch or gentoo, you're going to love it.
- Lots of tools of varying quality

SPIDER & SCAN

Pros

+Easy and fast

+Little setup needed

+Gets actual results

Cons

- Quiet malfunctions
  - Logout detection
  - Session invalidation
  - Fix: use a magic cookie in testing
- Limited coverage
  - Especially on responsive sites

ACTIVE SCAN

Scan whatever the user browses

1. User makes a request or sends a form
2. Form and URL gets scanned in a few hundred ways
3. Repeat

ACTIVE SCAN

Pros

+Good support for manual testing

+Visibility of malfunctions

+Works well on AJAX as well

Cons

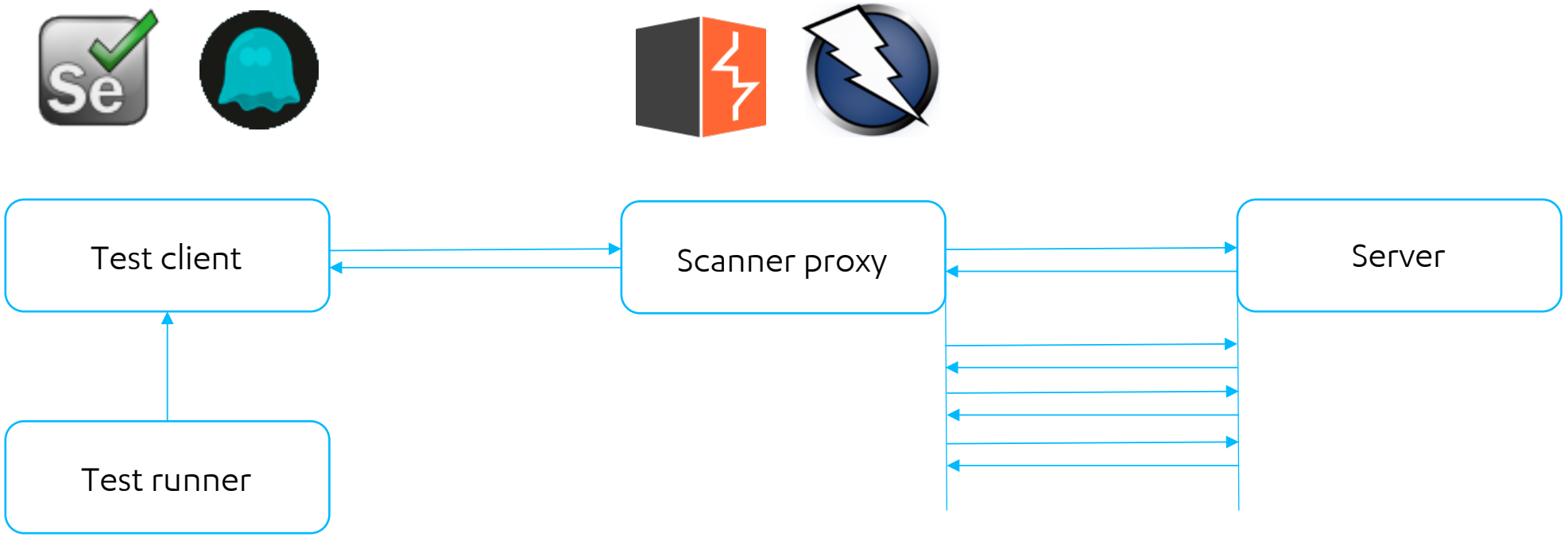
- Requires a human

ACTIVE SCAN OF TEST CASES

Active scanning, but replace human with functional tests

1. Run test case through proxy
2. Test request gets scanned
3. Repeat

ACTIVE SCAN OF TEST CASES



F-Secure.

MITTN

- Glue between:
  - Security testing tools
  - Functional tests
  - Findings database
- "Engineers interface"

ACTIVE SCAN OF TEST CASES

Pros

+ Same coverage as the functional tests used
+ No detailed security knowledge needed

Cons

- Harder to set up
- Reporting may not integrate
- Not a human


POTENTIAL
HICCUPS

FALSE POSITIVES

- There will always be false positives
  - ~~Ignore them~~
  - Verify, flag, and store
- Verification may need specialized security knowledge
  - Ask your security team member
  - Ask your local nerd

CONNECTING FINDINGS TO TEST CASES

- Which functional test triggered an issue
  - May be easily seen from the triggering request
  - May not even matter
- Could we fail a functional test if it causes a security issue?
  - Requires tight integration of test framework and security tool

ARE YOU CYBER-SAFE NOW?

- Unfortunately, no
- Automated tools can only do so much
- Some security culture will seep into the team
- Some easy-to-exploit issues will have been remediated

DO I STILL HAVE A JOB?

- Yes, tools are limited by complexity
- Security is much larger than scanning
  - Way
    - Way
      - Way
        - Larger


F-Secure®