Behavioural Security Group

UNIVERSITÄT BONN

Secure Software Engineering

Winterterm 2025/26

Risk Management & Risk Driven Test Planning

Dr. Christian Tiefenau

Outline today

- Vulnerabilities of the day
  - Log Overflow
  - Path Traversal
- Risk Assessment
  - Risk, When is a system secure (enough)?
  - Risk assessment in process
  - Protection Poker
  - Risk Management
- Risk-Driven Test Planning
  - Top-Down
  - Bottom-Up

Behavioural Security Group

UNIVERSITÄT BONN

Vulnerability of the Day #1

Log Overflow

Log Overflow

Behavioural Security Group

UNIVERSITÄT BONN

```
public class LoggedALot {
    private static Logger log = Logger.getLogger(LoggedALot.class);

    public static void main(String[] args) {
        PropertyConfigurator.configure(LoggedALot.class.getResource(args[0]));
        log.info("Some message.");
        log.info("Another message.");
        log.info("Yet another message.");
        System.out.println("Logging done!");
    }

}
```

Log Overflow

```
# Everything connects to the root logger
# INFO level, output is to a file named myfile
log4j.rootLogger=INFO, myfile

# Direct log messages to a log file
log4j.appender.myfile=org.apache.log4j.FileAppender
log4j.appender.myfile.file=loggedalot.log
log4j.appender.myfile.layout=org.apache.log4j.PatternLayout
log4j.appender.myfile.layout.ConversionPattern=%d{ABSOLUTE} %5p %c{1}:%L - %m%n
```

Question: Why is a log overflow problematic? How can an attacker exploit it?

-> Flood server with requests which leads to DoS (full hard drive or slow responses)

Log Overflow

Behavioural Security Group

UNIVERSITÄT BONN

CWE-400: Uncontrolled Resource Consumption

Weakness ID: 400
Vulnerability Mapping: DISCOURAGED
Abstraction: Class

View customized information: Conceptual | Operational | Mapping Friendly | **Complete** | Custom

Description

The product does not properly control the allocation and maintenance of a limited resource.

https://cwe.mitre.org/data/definitions/400.html

Log Overflow

```
# Everything connects to the root logger
# INFO level, output is to a file named myfile
log4j.rootLogger=INFO, myfile

# Direct log messages to a log file
log4j.appender.myfile=org.apache.log4j.RollingFileAppender
log4j.appender.myfile.MaxFileSize=1MB
log4j.appender.myfile.MaxBackupIndex=1
log4j.appender.myfile.file=loggedalot.log
log4j.appender.myfile.layout=org.apache.log4j.PatternLayout
log4j.appender.myfile.layout.ConversionPattern=%d{ABSOLUTE} %5p %c{1}:%L - %m%n
```

Much better: Log rotation with time/size limits

- Rolling Log
- E.g., Max 1 MB or daily logs

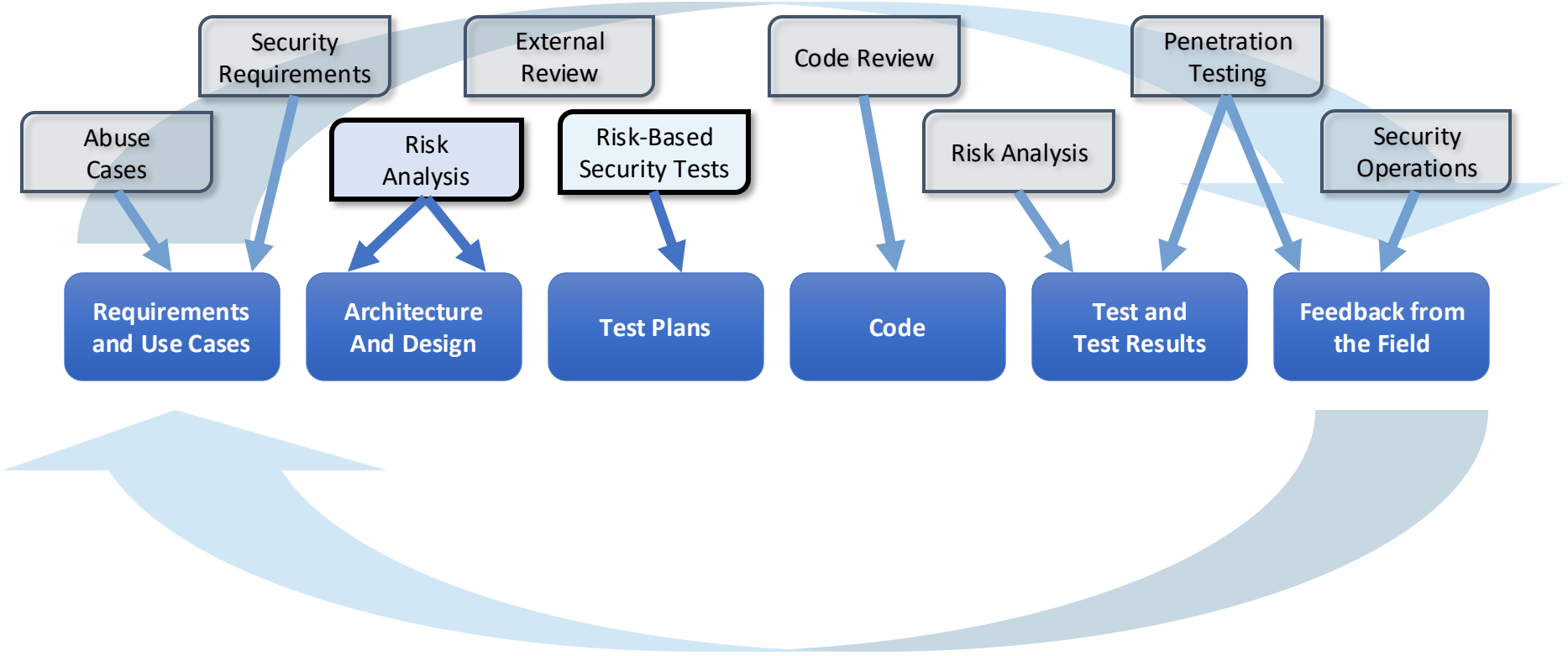
Real-world Log Overflow

CVE-ID	
CVE-2013-0231	Learn more at National Vulnerability Database (NVD) • Severity Rating • Fix Information • Vulnerable Software Versions • SCAP Mappings
Description	
The pciback_enable_msi function in the PCI backend driver (drivers/xen/pciback/conf_space_capability_msi.c) in Xen for the Linux kernel 2.6.18 and 3.8 allows guest OS users with PCI device access to cause a denial of service via a large number of kernel log messages. NOTE: some of these details are obtained from third party information.	

- Beware:
  - Attackers can exploit rolling logs as well!
  - Cover their tracks by deliberately causing log overwrite
- Mitigation to this problem:
  - Limits on number of allowed requests per user, etc.
  - Zip backups
- General point to take away:
  - Always restrict the resources a user can cause the system to use!

Security Risk Assessment & Risk-Driven Test Planning

Today‘s lecture



What is Risk?

Risk(incident) = p(occurrence) * impact

- The risk associated with an event is the probability that the event will happen times the impact magnitude of the event

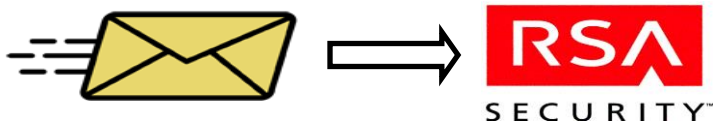
- Humans tend to over/underestimate risks
  - Difference between analytical risk assessment and human risk perception
  - Low p(occ), high impact
    … terrorist attacks, struck by lightning
  - High p(occ), low impact
    … credit card theft, keeping my old truck unlocked

How risky can it be to use Adobe Flash Player? 2011 Attack on RSA and U.S. Weapon Manufacturers

Behavioural Security Group

UNIVERSITÄT BONN

1: Social engineering & phishing



- March 3: Fake email to some RSA employees: [2011 Recruitment plan.xls] with embedded flash zero-day CVE-2011-0609 in Adobe Flash Player.
- Planted "Poison Ivy" trojan horse.

2: Digital Shoulder Surfing

- Poison Ivy connects back to control server, giving full control to attacker.
- Attacker gradually moves towards higher value accounts and data.

3: Collecting SecureID secret seed records, downloading them from staging server.

- RSA issues warning on March 17
- Unusually fast (e.g., attack on Nortel went unnoticed for more then 10 years)

4: Exploiting compromised SecureID to break into the target systems at defense industry.

- June 3: Lockheed discloses a blocked attack, which exploited the breach at RSA.
- RSA announced replacement program for tokens (>40M tokens worldwide, Lockheed > 45'000).
- August 2011: RSA acknowledge immediate 66M$ for recovery.
- March 27, 2012: NSA attributes attack to Chinese hackers

Similar Attack Pattern in 2016…

- Attackers planted trojans on machines of the German Bundestag
- Spread from one machine that was infected by one malicious email.

- Takeaway: Impact of opening a mail can be disastrous! But…
  … we still use mails.

  So when do we take the risk?

When is a system secure enough?

UNIVERSITÄT BONN

- **Your smart home-enabled fridge?**
  - When one can keep out script kiddies?
- **Your autonomously driving car?**
  - When you can successfully defend against over-the-air attacks?
- **Your industrial production line?**
  - When your direct competitors have less secure system?
- **Your weapons manufacturing plant?**
  - When you can defend against nation-state actors?
- **Hence: Risk depends on context, and so does sensible mitigation effort**

When is a system secure enough?

- **Your sma**
  - When
- **Your aut**
  - When
- **Your ind**
  - When
- **Your wea**
  - When
- **Hence: R** **ffort**

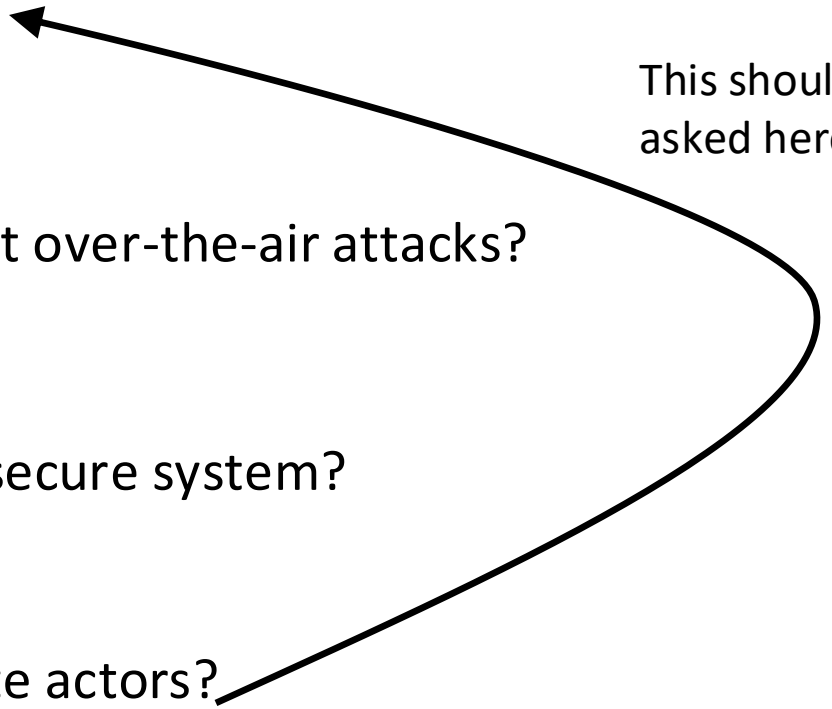
Chinese Electronics Firm to Recall its Smart Cameras recently used to Take Down Internet

Monday, October 24, 2016 Swati Khandelwal

You might be surprised to know that your security cameras, Internet-connected toasters and refrigerators may have inadvertently participated in the massive cyber attack that broke a large portion of the Internet

When is a system secure enough?

- **Your smart home-enabled fridge?**
  - ~~When one can keep out script kiddies?~~
- **Your autonomously driving car?**
  - When you can successfully defend against over-the-air attacks?
- **Your industrial production line?**
  - When your direct competitors have less secure system?
- **Your weapons manufacturing plant?**
  - When you can defend against nation-state actors?
- **Hence: Risk depends on context, and so does sensible mitigation effort**



This should have been asked here!

Example: 2017 incident with Miele Professional devices

Behavioural Security Group

UNIVERSITÄT BONN

Dishwasher has directory traversal bug

Thanks a Miele-on for making everything dangerous, Internet of Things firmware slackers

Richard Chirgwin Sun 26 Mar 2017 // 23:08 UTC

Don't say you weren't warned: Miele went full Internet-of-Things with a network-connected dishwasher, gave it a web server, and now finds itself on the wrong end of a security bug report – *and* it's accused of ignoring the warning.

Quelle: TheRegister

Example: 2017 incident with Miele Professional devices

Miele verspricht Sicherheits-Update für Desinfektionsautomaten

29.03.2017 12:47 Uhr – Torsten Kleinz

vorlesen

PG 8527

PG 8528

Quelle: Heise

Another Example: HeartBleed Bug

Behavioural Security Group

UNIVERSITÄT BONN

- You wake up on April 7, 2014 and read the news about the HeartBleed bug
- How do you know whether and how badly your project is affected?

- Your project does not even use OpenSSL
  - Go relax, cost: zero

- Your project links to OpenSSL but it turns out it's not actually being used
  - Remove dependency, go relax, cost: low

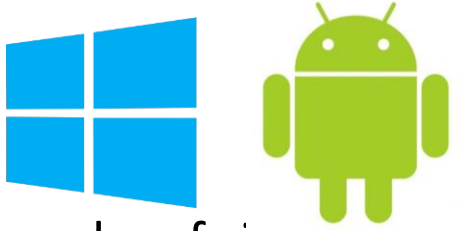
- Your project actively uses OpenSSL in a vulnerable configuration
  - Impact really depends on the type of connection and data
  - Might be urgent to fix: update library, update private keys! cost: possibly high
  - **Risk assessment might have already given you a list of steps to do this fast**

exploit

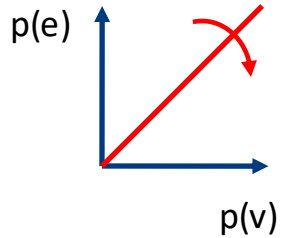
p(occurrence) vs. p(vulnerability)

- Not every vulnerability will be exploited
- p(occurrence) - existence of an exploit is…
  - Increased by *more* vulnerabilities
  - Increased by a *far-reaching* vulnerability
  - Increased by *discoverable vulnerabilities (*c.f. *security by obscurity)*
  - Increased by *scope of the project* (cannot always be controlled)

- Other factors that one cannot truly control
  - Market share → exposure
  - New malicious actors (e.g. activism spike)
  - Many, many other factors that we must ignore for the sake of simplicity



- Thus, we generally assume p(vulnerability) is proportional to p(exploit)
- Mitigations might change ratio between the two



Why do we study risk?

UNIVERSITÄT BONN

- Many outcomes are possible, not all are probable

- Enumeration
  - List of all potential threats, so (hopefully) none are overlooked early in design.
- Prioritization
  - E.g., a web app **could** be hacked through dozens of different vectors, but based on the architecture, SQL injection might be far more likely than hardware tampering.
- Discussion
  - Make informed decisions when it comes to allocating time/budget

Naïve Security Risk Assessment

Behavioural Security Group

UNIVERSITÄT BONN

- The naïve approach
  - Write down your worst fears for the system
  - Try to avoid those things

- Cons
  - Requires a big “bag of tricks”
    - Vulnerability of the day is only one of them
  - Easily overwhelming for security
  - Might miss whole areas

- But where could we start?

Assets

- Every software system has assets
  - Domain-specific e.g. patient records
  - Domain-independent e.g. passwords
  - Intangible properties e.g. availability

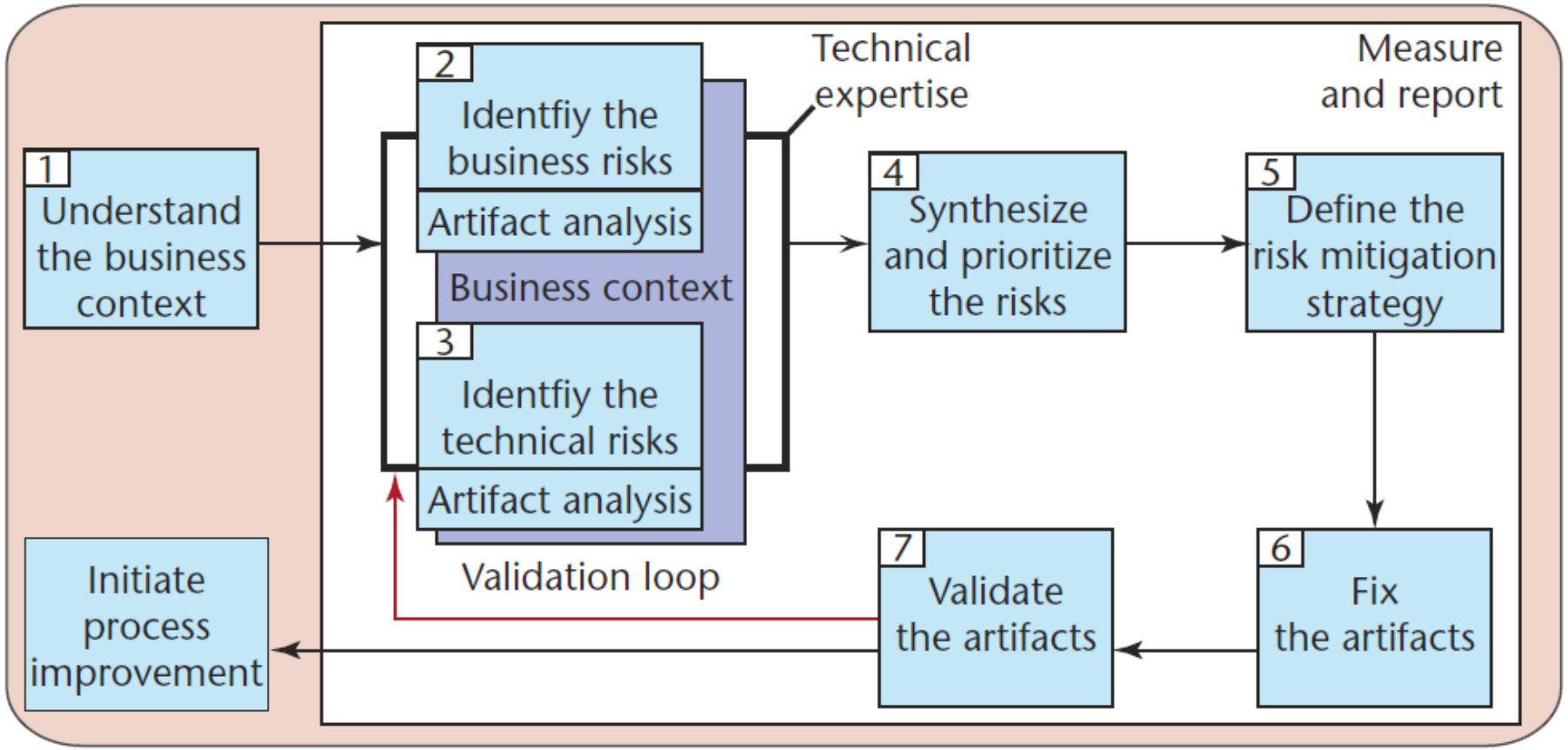
- These can be identified at the requirements and *design* stages

- Assets exist in the deployed system, so source code is not (necessarily) an asset

Places where assets live

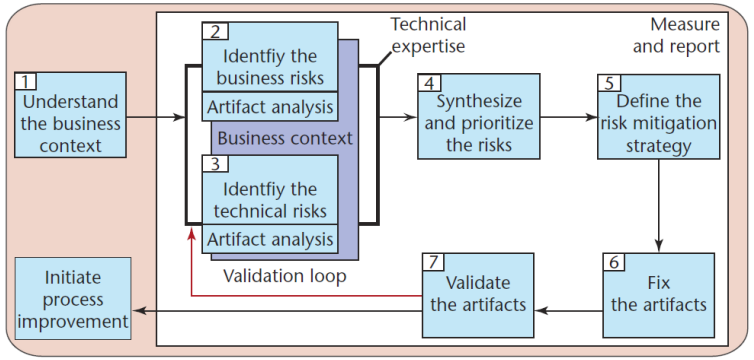
- Database tables
- User-supplied data
- Configuration files
- Configuration consoles
- File systems
- Security feature inputs
- Logs
- Sandboxing features
- Built-in examples
- Network traffic
- Cookies
- User interfaces

Risk Assessment in Process



- Source: https://www.synopsys.com/blogs/software-security/software-risk-analysis/

Risk Assessment in Process



Source: https://www.synopsys.com/blogs/software-security/software-risk-analysis/

1. Learn about the context
   - Specifications, Architecture docs, identify threats,…
2. Identify business risks
   - Interruption of cash flow,…
3. Identify technical risks
   - Data leaks,…
4. Determine probability of compromise + impact analysis
   - Let's us prioritize and rank risks
5. Develop mitigation strategy
6. + 7. Report findings, adapt changes to assets/artifacts

Exercise

- Take 3 minutes to discuss with your neighbor:

  What is the risk involved in somebody hacking your email account?

The Planning >= The Plan

- One of the most important elements of risk analysis is the process itself
  - Discussions that are brought up
  - Fighting over the mitigation strategies give insights into priorities of stakeholders

- Communication is very important at this stage

- Assessing the *change* in risk is more sound than absolute numbers
  - New assets?
  - Increased exposure?
  - Increased p(exploit)?

Abuse Cases vs. Risk Assessment

UNIVERSITÄT BONN

- **Abuse & Misuse Cases**
  - Involves planning
  - Potentially infinite
  - Emphasize domain
  - Scenario-driven
  - Originates from abusing/misusing functionality
  - What if?
  - What are the dimensions?

- **Risk Assessment**
  - Involves planning
  - Potentially infinite
  - Emphasize all risks
  - Quantitative
  - Originates from CIA, assets, p(exploit)
  - What might?
  - What is the risk value in each dimension?

Protection Poker

- A combination of product & process risk
  - Trace stories to assets
  - Quantify the risk for prioritization
    - Ease of attack
    - Value of the asset
  - Discuss the elements of the risk

- Originally designed for agile processes
  - Assumes we are in a sprint
  - Not comprehensive, but just-in-time

Laurie Williams, Andrew Meneely, and Grant Shipley. 2010. **Protection Poker: The New Software Security "Game"** *IEEE Security and Privacy* 8, 3 (May 2010), 14-20. DOI=http://dx.doi.org/10.1109/MSP.2010.58

Story Points Estimation

- In PP, we use story points
  - Dimensionless (unit-less)
  - Should *not* translate to hours, effort, etc.
  - You create the values (there is no “fixed” scale)
- Limited to a few choices
  - Why argue over 51 vs. 50?
  - Exponential in scale (~Fibonacci)
- Security risk = Ease of attack * value of the asset
  - Value between 1 and possibly infinite
  - More about the ”intervals” and resulting groups

Protection Poker - Example

- Step 1: Assign values to assets (Using fibonacci numbers)
  - 1, 2, 3, 5, 8, 13, 21, 34,…

Asset Value	Customer Data
	Customer login ID
	Customer password
	Email
	Customer name (first)
	Customer name (last)
	Credit card ID
	Credit card PIN
	Driver's license or passport
	Customer #

Protection Poker - Example

- 3 new features:
  1. Add „known allergies“
  2. Add „emergency responder“ role
  3. Add „customer group“ role

Asset Value	Customer Data
2	Customer login ID
5	Customer password
8	Email
3	Customer name (first)
8	Customer name (last)
21	Credit card ID
34	Credit card PIN
21	Driver's license or passport
1	Customer #

Protection Poker - Example

1. Add „known allergies“
2. Add „emergency responder“ role
3. Add „customer group“ role

- Create assets when necessary (see blue rows)
- Define which assets are needed/affected

Asset Value	Customer Data	**Used in Feature #**
2	Customer login ID	
5	Customer password	
8	Email	2,3
3	Customer name (first)	2,3
8	Customer name (last)	2,3
20	Credit card ID	
40	Credit card PIN	
20	Driver's license or passport	
1	Customer #	1,2,3
2	Known allergies	1,2
8	Customer group	3
8	Customer group #	3

Protection Poker - Example

- Sum up the points for each feature

Feature #	Total Value Points
Known Allergies	3
Emergency Responder	22
Groups	36

Protection Poker - Example

- Sum up the points for each feature
- Add Ease points

Feature #	Total Value Points	Ease Points
Known Allergies	3	1
Emergency Responder	22	5
Groups	36	13

Ease points: 1 – hard to attack … 100 – easy to attack

Protection Poker - Example

UNIVERSITÄT BONN

- Sum up the points for each feature
- Add Ease points
- Calculate Security Risk (Total Value Points * Ease Points)

Feature #	Total Value Points	Ease Points	Security Risk
Known Allergies	3	1	3
Emergency Responder	22	5	110
Groups	36	13	468

Risk Management

- Beyond assessment
  - Assess: Enumerate, Prioritize, Discuss
  - Manage: Act on those discussions
- Mitigate risk
  - Every risk has a mitigation
  - Plan, plan, plan
  - Know the limitations of your solution
- Track risk
  - Effective mitigations?
  - Increased p(exploit)?
  - Increased asset value?

Behavioural Security Group

UNIVERSITÄT BONN

Vulnerability of the Day #2

Path Traversal

Path Traversal

- User control of resource path
- Access of files outside the intended folder
  - ../../../.ssh/id_ecdsa_priv.key

```
public class ShowSandboxedFile {

    public static void main(String[] args) throws IOException {

        System.out.print("Filename you want to show in ./app/sandbox: ");
        Scanner scanner = new Scanner(System.in);

        System.out.println("\nThe contents of the file is...\n");
        File file = makeFile(scanner.nextLine());
        Scanner fileScanner = new Scanner(file);

        while (fileScanner.hasNextLine()) {
            System.out.println(fileScanner.nextLine());
        }

        fileScanner.close();
        scanner.close();
    }

    private static File makeFile(String filename) throws IOException {
        return new File("./app/sandbox/" + filename);
    }

}
```

Path Traversal

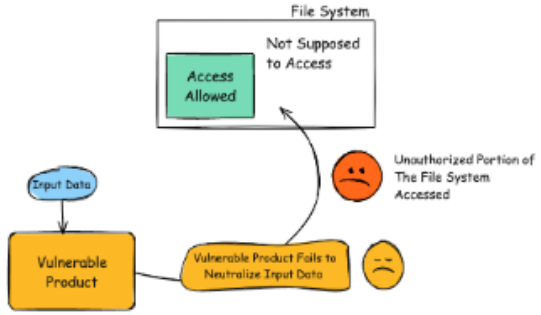
CWE-22: Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')

Weakness ID: 22
Vulnerability Mapping: ALLOWED
Abstraction: Base

View customized information: Conceptual | Operational | Mapping Friendly | **Complete** | Custom

Description

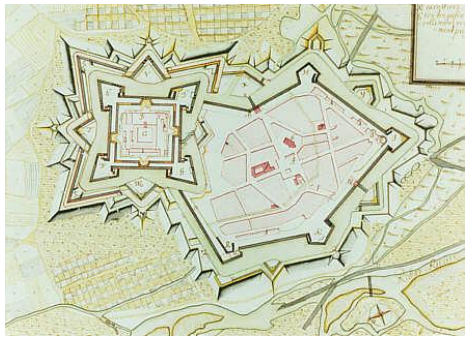
The product uses external input to construct a pathname that is intended to identify a file or directory that is located underneath a restricted parent directory, but the product does not properly neutralize special elements within the pathname that can cause the pathname to resolve to a location that is outside of the restricted directory.



https://cwe.mitre.org/data/definitions/22.html

Mitigations

- Permission management
  - Restrict file access to allowed folders only
- In-code checks
  - Allowlisting
  - Verify canonicalization
- Use containerization/VM



Path Traversal – Possible Mitigation

Behavioural Security Group

UNIVERSITÄT BONN

```
private static File makeFile(String filename) throws IOException {
    File sandboxDir = new File("./app/sandbox/");
    File file = new File(sandboxDir, filename);
    if (!file.getCanonicalPath().startsWith(sandboxDir.getCanonicalPath()))
        throw new IllegalArgumentException("Stay in the sandbox please!");
    return file;
}
```

Real-world Path Traversal

CVE-ID	
CVE-2009-2902	Learn more at National Vulnerability Database (NVD) • Severity Rating • Fix Information • Vulnerable Software Versions • SCAP Mappings
Description	
Directory traversal vulnerability in Apache Tomcat 5.5.0 through 5.5.28 and 6.0.0 through 6.0.20 allows remote attackers to delete work-directory files via directory traversal sequences in a WAR filename, as demonstrated by the ...war filename.	

- Also very common in web apps
- Somewhat similar to SQL injection in the sense that string concatenation is where things go wrong

Security Risk Assessment & Risk-Driven Test Planning

Idea of Risk-Driven Test Planning

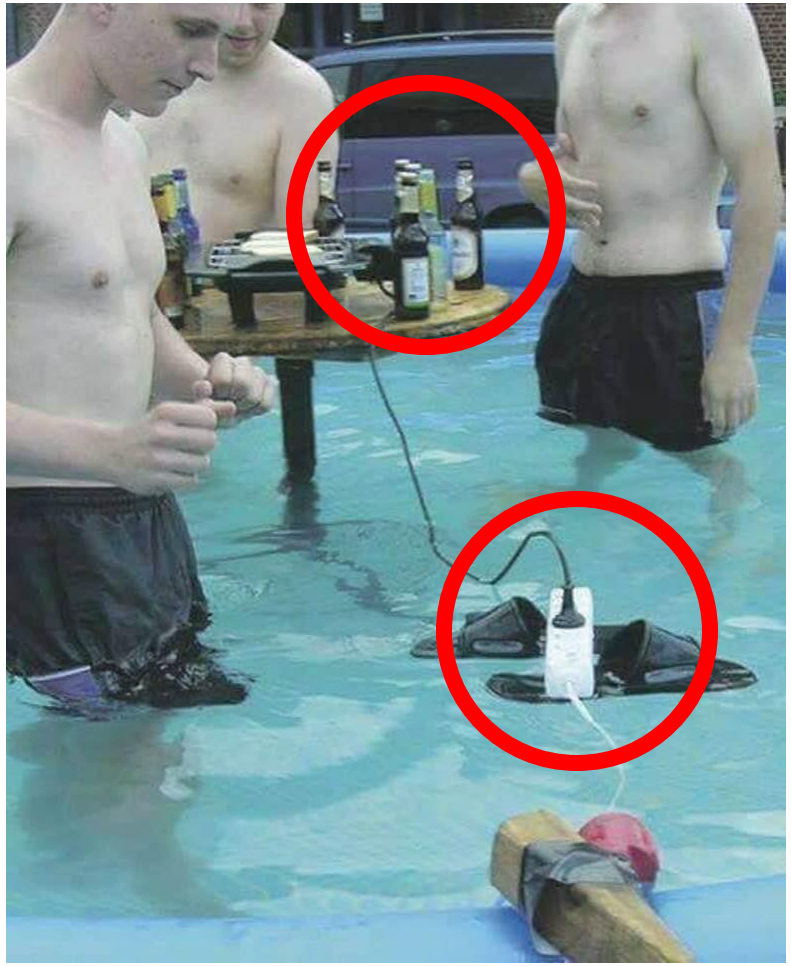
- In case of a service disruption, we do not want this:

- But rather this:

Goal of Risk-Driven Test Planning

- Mitigate negative impact on the customer
- Create the mitigation strategies early
- Allow a “disruption-free” usage of the product

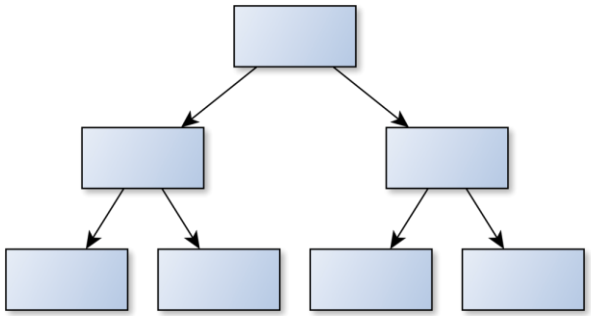
How to do Risk-Driven Test Planning



1) Do a risk analysis as described in the previous slides.
2) Focus on the area of the product where there is a high impact on business due to failure or high likelihood of failure in the production.
3) Fail fast.

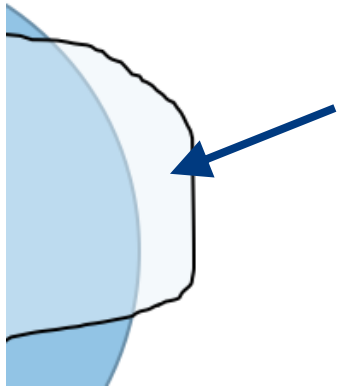
Top-Down Test Planning

- Start with the broad analysis of the domain
  - Goals
  - Assets
  - Top-down analysis (“forest-level”)



- Goals → Risks → Indicators → Tests

- Vulnerability-focused
  - Too much functionality
  - Move on when the vulnerability is found
  - Valued assets are given a priority



Goals -> Risks

- Goals
  - Overall objectives of the system
    - Business-focused objectives -> revenue streams
    - User-focused objectives -> branding
  - Constraints on the development, e.g. release dates
  - Availability concerns
  - A product has a ***finite*** number of goals

- High-Level Risks
  - Directly map to 1+ goals
  - Influenced by both p(vuln) & assets
  - A product has a ***finite*** number of high-level risks

Risks -> Indicators

- How will we know that a high-level risk manifests itself?
  - A measurable outcome of the system
  - What is the poor behavior of the system?
  - What are the potential underlying causes?

- E.g. downtime, asset exposure

- Indicators are potentially infinite

  …but three will get you very far

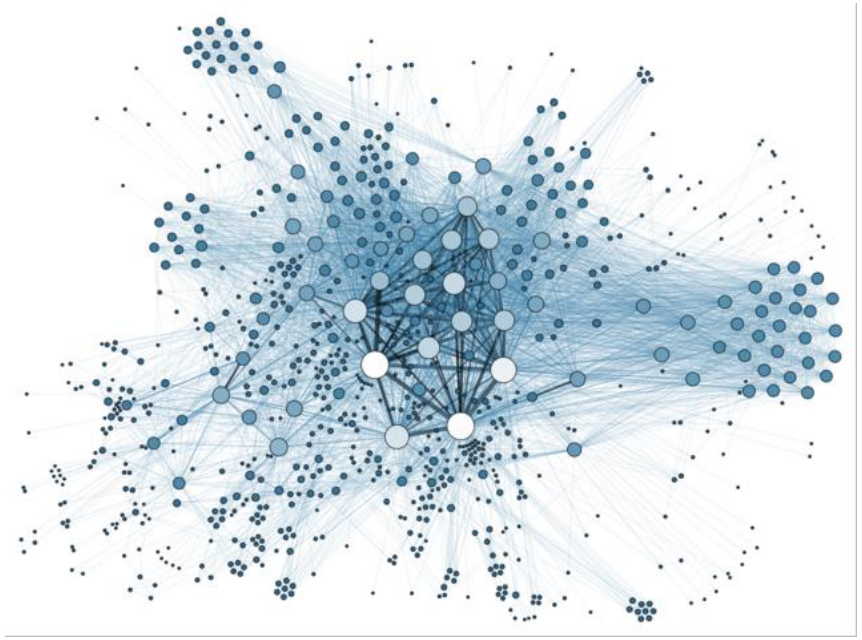
Indicators -> Tests

- Given an indicator, how do we ensure that the indicator is avoided or satisfied?
  - Test for it!
  - Key: *specific expectations*
- Tests are even more infinite
- Might require more design & architecture work to execute this step

e.g. BlogReader Goals

- Goals:
  - (user) Provide pretty-looking formatting of user's blogs
  - (business) Make money via advertisements
  - Constraints: web-based configuration, mobile app, continuous release / DevOps
  - Availability: 99.9% uptime (8.76 hours downtime/year)

- Assets
  - User subscription information (e.g. blog feeds)
  - Personal data (e.g. emails)
  - Social graph



e.g. BlogReader Risks -> Indicators -> Tests

UNIVERSITÄT BONN

- High-Level Risk: social graph disclosure
  - Indicator: APIs allow unauthorized access to social graph
  - Test: direct access to user friends should be denied
  - Test: votes logged are anonymized or digested

- High-Level Risk: availability is compromised
  - User-focused: users are unable to reach their feeds
  - Business-focused: customers move to a different tool
  - Indicators: high processor loads, full hard drives, downtime
  - Tests: stress tests for networking, disk activity, and crashes

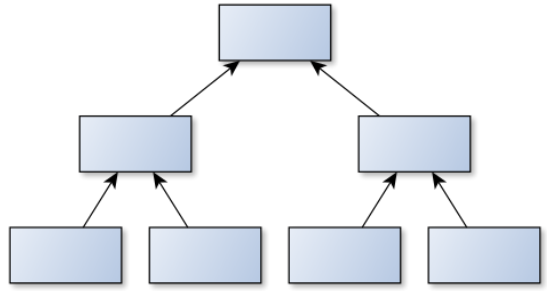
When do I do what?

- At the requirements phase, define:
  - Goals
  - Risks
- At a high-level design phase (i.e. architecture), define:
  - Indicators
  - Some tests
- At a low-level design phase (incl. maintenance), define
  - More Tests
- All the time
  - Track everything and write it down
  - “Bubble up” new risks from new test ideas

Bottom-Up Security Test Planning

- Step 1: Write down a lot of tests
  - Document it in short form
  - Doesn’t have to be complete – just seeds for now

- Step 2: Group those tests into various categories
  - By assets
  - By functionality
  - By CIA consequences
  - By what your team requires to run the test
  - etc.



- Step 3: Revise the categories as a group
  - Are there groups missing?
  - Art there tests missing in a group?

- Step 4: Add more tests to each category

Top-down vs. Bottom-up

- Top-down security test planning
  - Benefit: tied to specific goals
  - Drawback: incomplete within the categories
    - “Just to check it off the list” syndrome
    - Miss out on planning for really creative tests

- Bottom-up security test planning
  - Benefit: gives you freedom to write your best tests immediately
  - Drawbacks: easy to miss stuff
    - Entire goals/categories/assets can get missed
    - Requires security expertise in the first place