Application Security (apsi)

Lecture at FHNW

Lecture 8, 2021

Arno Wagner, Michael Schläpfer, Rolf Wagner

<arno@wagner.name>, <{michael.schlaepfer, rolf.wagner}@fort-it.ch>

Agenda

- 12:15 – 13:00: Introduction to Bug Bounty
- 13:15 – 14:00: Introduction to WebGoat
- 14:15 – 15:00: Exercise Session

Why Bug Bounty?

Digitalization is massively increasing the **complexity of IT.**

Due to the high expectations of **time to market,** products & services are often made available to the public very early.

This leads to a larger **attack surface** and a higher **damage potential.**

Security incidents, service interruptions, lost revenue.

TRADITIONAL SECURITY APPROACHES ARE NO LONGER ENOUGH

What’s Bug Bounty?

Digital / IT
Vulnerabilities

Rewards,
Bounties

Crowd
Community
of Friendly
Hackers

Friendly hackers search for security bugs and are rewarded for the vulnerabilities they find

BB Program

- What has to be defined?
  - Rules: What is allowed, what not?
  - Scope, Targets: What should be tested, what not?
  - Bounties: What bounties are paied out? For which severity?
  - Safe Harbor: Are you allow to hack anyway? Legal Safe Harbor.

- How to standardize severity levels?

Common Vulnerability Scoring System – CVSS

- https://www.first.org/cvss/
- The Common Vulnerability Scoring System (CVSS) provides a way to capture the principal characteristics of a vulnerability and produce a numerical score reflecting its severity. The numerical score can then be translated into a qualitative representation (such as low, medium, high, and critical) to help organizations properly assess and prioritize their vulnerability management processes.
- CVSS is a published standard used by organizations worldwide

Rating	CVSS Score
None	0.0
Low	0.1 - 3.9
Medium	4.0 - 6.9
High	7.0 - 8.9
Critical	9.0 - 10.0

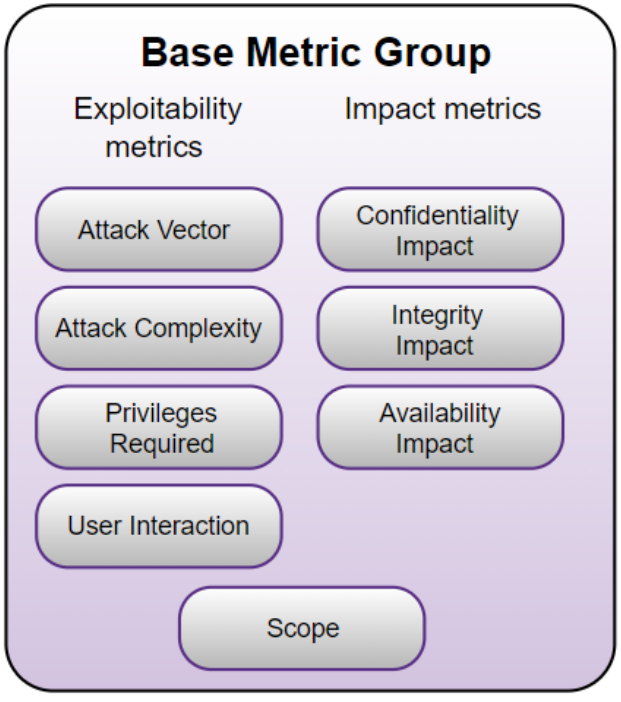
Source: first.org/cvss

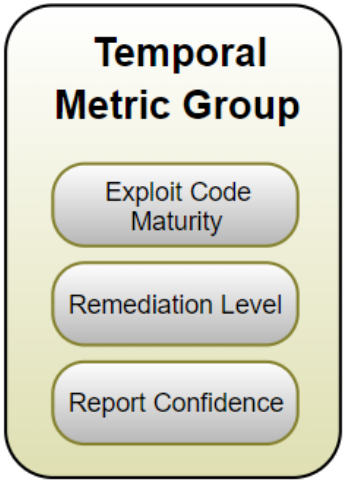
Common Vulnerability Scoring System – CVSS

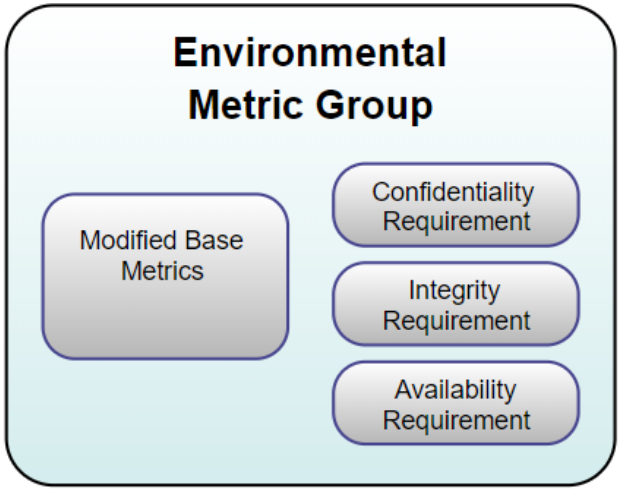
- How to score a vulnerability?

Common Vulnerability Scoring System – CVSS

- How to score a vulnerability?







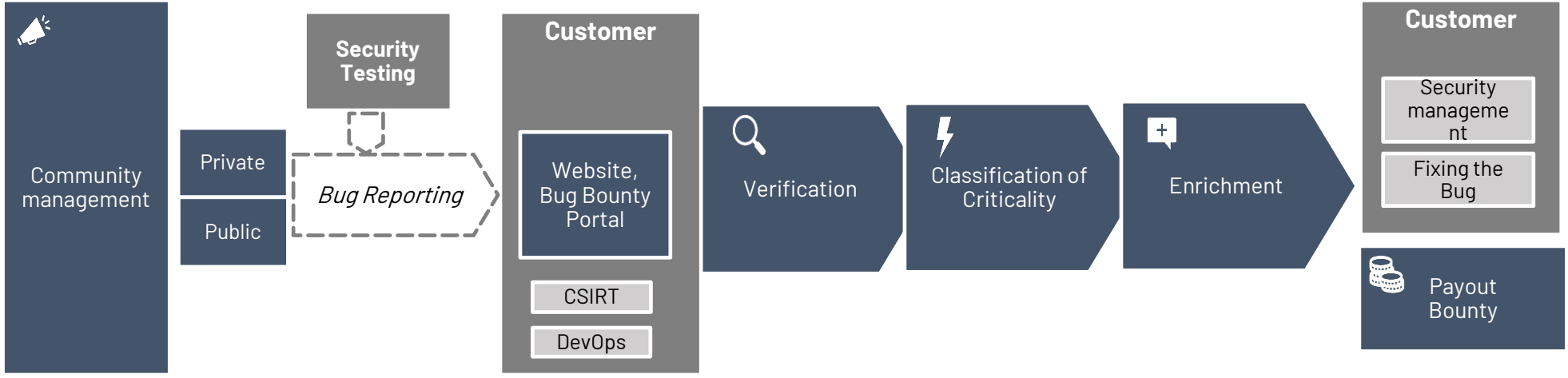
Source: first.org/cvss

More details: https://www.first.org/cvss/specification-document#1-2-Scoring

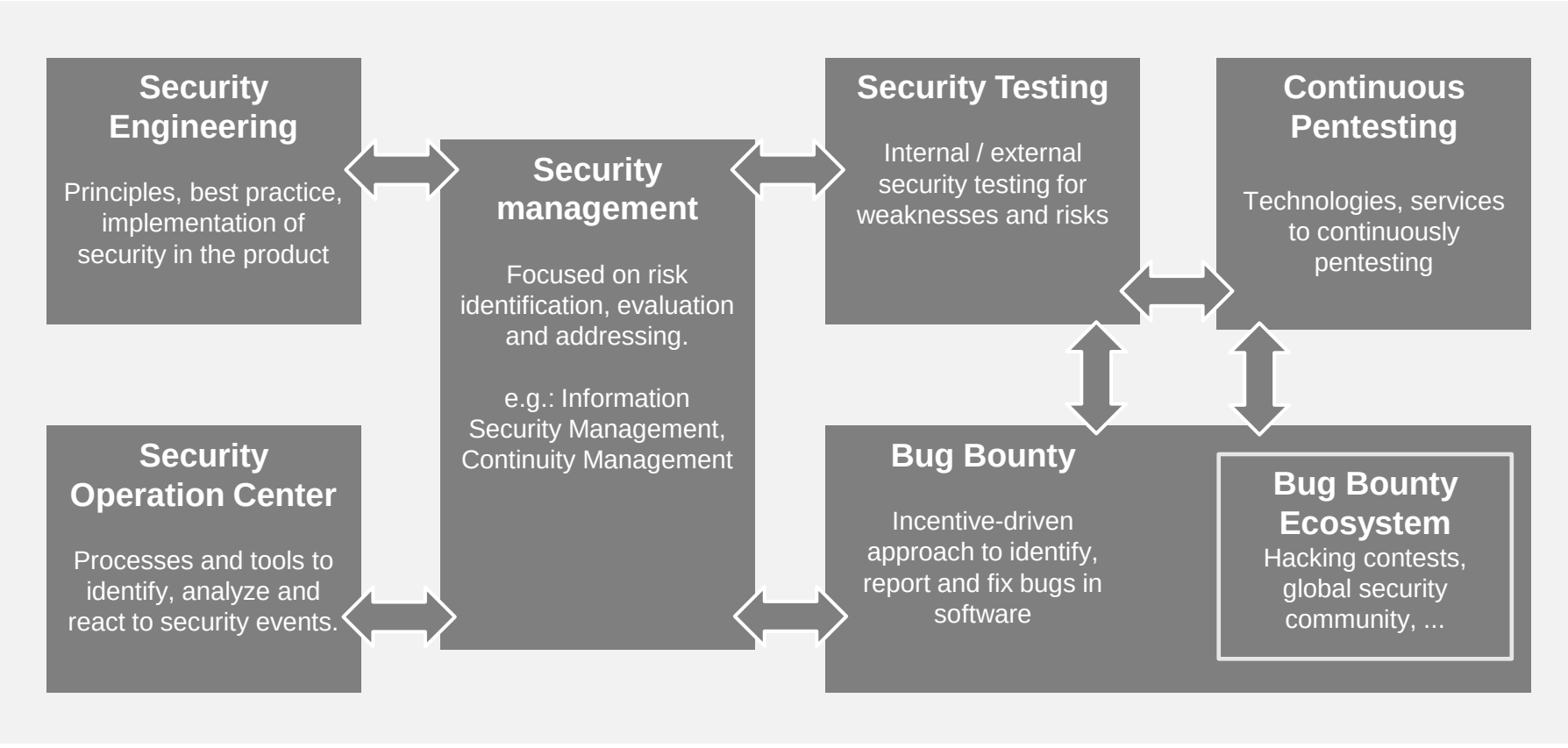
Let's have a look at the calculator: https://www.first.org/cvss/calculator/3.1

! For the exam, you have to understand the metrics.

BB Process


Community management
Private
Public
Security Testing
Bug Reporting
Customer
Website, Bug Bounty Portal
CSIRT
DevOps
Verification
Classification of Criticality
Enrichment
Customer
Security manageme nt
Fixing the Bug
Payout Bounty

BB Overview & Interaction



Bug Bounty Program FHNW @BBHub

- Let's have a look: https://bugbountyhub.com/de-ch/programs/fhnw/protected
  - Rules in General
  - Hacking Methods
  - Qualified vulnerabilities
  - Non-qualified vulnerabilities
  - Legal Safe Harbor
  - Bounties

Remark: You are welcome to participate in the Bug Bounty Program at FHNW. However, you do not have to and it is not part of the exams.
Find out how: https://owasp.org/www-project-web-security-testing-guide/stable/4-Web_Application_Security_Testing/#

Agenda

- 12:15 – 13:00: Introduction to Bug Bounty
- 13:15 – 14:00: Introduction to WebGoat
- 14:15 – 15:00: Exercise Session

WebGoat

https://owasp.org/www-project-webgoat/

Learn the hack - Stop the attack

WebGoat is a deliberately insecure application that allows interested developers just like you to test vulnerabilities commonly found in Java-based applications that use common and popular open source components.

WebWolf the small helper

WebWolf is a separate web application which simulates an attackers machine:

- Host a file
- E-mail client
- Landing page for incoming requests

WebGoat

The WebGoat exercise environment

In the following two lectures you will get hands-on experience using a vulnerable system (*WebGoat*). You will learn to hack the most common vulnerabilities with this respect.

Setup

The setup is pretty simple. The OWASP offers docker images for *WebGoat* and *WebWolf* (which you as an attacker will use for certain exercises).

Install docker-compose on the apsi-host (vagrant)

To start the environment, the OWASP offers a simple *docker-compose.yml* file, which you will in this folder. However, to use `docker-compose`, you will first have to install it. You will find a simple script in this folder which will do the job for you. First make the *install_docker-compose.yml* file executable:

`vagrant@apsi-host:~/FHNW-apsi/Vorlesung/v08$ sudo chmod u+x install_docker-compose.sh`

and then execute it:

`vagrant@apsi-host:~/FHNW-apsi/Vorlesung/v08$ ./install_docker-compose.sh`

P.s. Maybe git fetch, git pull necessary

After installation you may test whether or not `docker-compose` is correctly installed on the demo-machine:

`vagrant@apsi-host:~/FHNW-apsi/Vorlesung/v08$ docker-compose -v`

You should now see the version of your `docker-compose` installation.

Start WebGoat and WebWolf

Now, that you have `docker-compose` installed, you may just have to execute it with the given *docker-compose.yml* file:

`vagrant@apsi-host:~/FHNW-apsi/Vorlesung/v08$ docker-compose up`

Using WebGoat

After *WebGoat* and *WebWolf* are running properly, you may just use your favorite browser on your host system and point it to:

`http://localhost:8080/WebGoat`

or

`http://localhost:9090/WebWolf`

respectively.

If you do so, just register a new user first. You will use these credentials further on to authenticate yourself.

WebWolf

Some challenges requires to have a local web server running.

WebWolf is for you the attacker it helps you while solving some of the assignments and challenges within WebGoat.

An assignment might for example require you to serve a file or connect back to your own environment or to receive an e-mail. In order to not let you run WebGoat open and connected to the internet we provided these tools in this application, called WebWolf.

Files

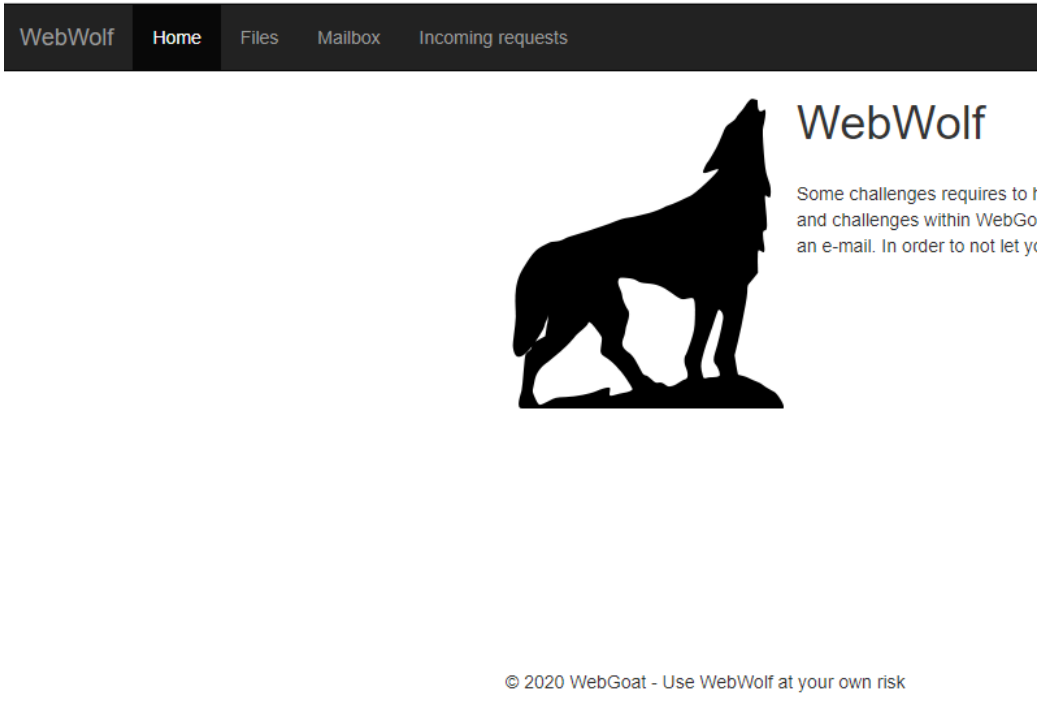
- Upload a file which you need to host as an attacker.
- Each file will be available under the following url: http://localhost:9090/files/{username}/{filename}.

Mailbox

- The mailbox of you as an attacker, all the mail send to {user}@{random} will be send to this mailbox.
- Only the user part is important the domain can be anything

Incoming requests

- Challenges in which you need to call your hacker machine WebWolf offers a simple httpd server functionality which only logs the incoming request.
- You can use the following URL: http://localhost:9090/WebWolf/landing/*



WebGoat

Learning in three steps:

- Explain the vulnerability
- Learn by doing
- Explain mitigation

We expect that you put ~8h effort into WebGoat.
Best Case: All Lessons ;-)
Minimal:

- A1 Injection (Intro & Advanced)
- A2 Broken Authentication
- A7 XSS

Agenda for the next lecture (15. November 2021)

- 12:15 – 14:00: Self-Work on WebGoat
- 14:15 – 15:00: Exercise Session
  - Q&A WebGoat
  - Walk through sample solutions

Agenda

- 12:15 – 13:00: Introduction to Bug Bounty
- 13:15 – 14:00: Introduction to WebGoat
- 14:15 – 15:00: Exercise Session

Exercises last lecture (v07)

Installing and setting up the Vagrant environment (for those who haven't done it yet).

- Instructions on the GitHub https://raw.githubusercontent.com/Fort-IT/FHNW-apsi/master/Demoenv/README.md
- Recommendation: Vagrant Environment

Read and understand the A1-A10 Risks Version 2021 ("Description" & "How to Prevent" chapters): https://owasp.org/Top10/

→ Exam: Questions to check understanding

A1-A10 Risks Version 2021

Questions?

Top 10:2021 List

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 (SSRF)

Go Hunting!

WebGoat (MUST) & Bug Bounty Programm FHNW (Voluntary)