1. Module - The Penetration Testing Process (komplett)

http://www.pentest-standard.org/

Steps before the penetesting

Rules Of Engagement: paperwork contains aggreements between client & penetester
Penetester need to know: Goal + Scope

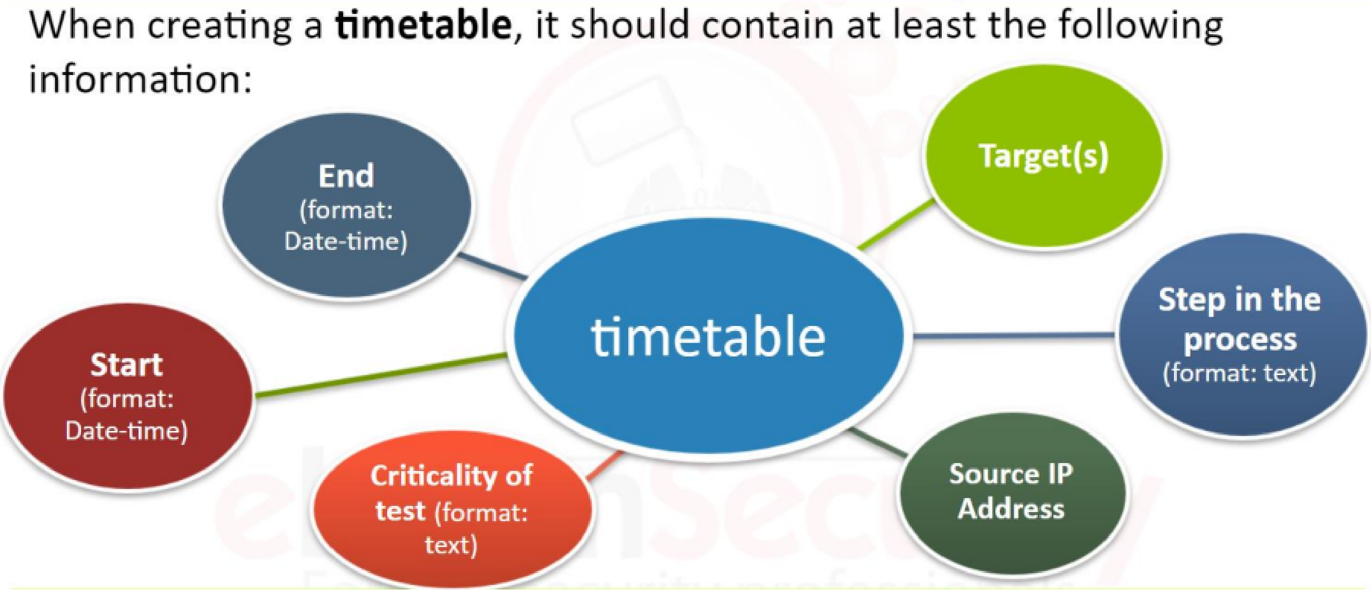
Goal & Scope

Scope: logical + Physical Scope
Logical : a departement or the whole org.
Physical: domain, subdomain, autonomous system and so on.

Penetester have to know and ask what subdomains are included, and what not..

Timetable

-> Make client aware what will happen, where and when!!!
-> timing of the tests can be changed during the engagement, as information is uncovered.



When creating a **timetable**, it should contain at least the following information:

timetable

End (format: Date-time)

Target(s)

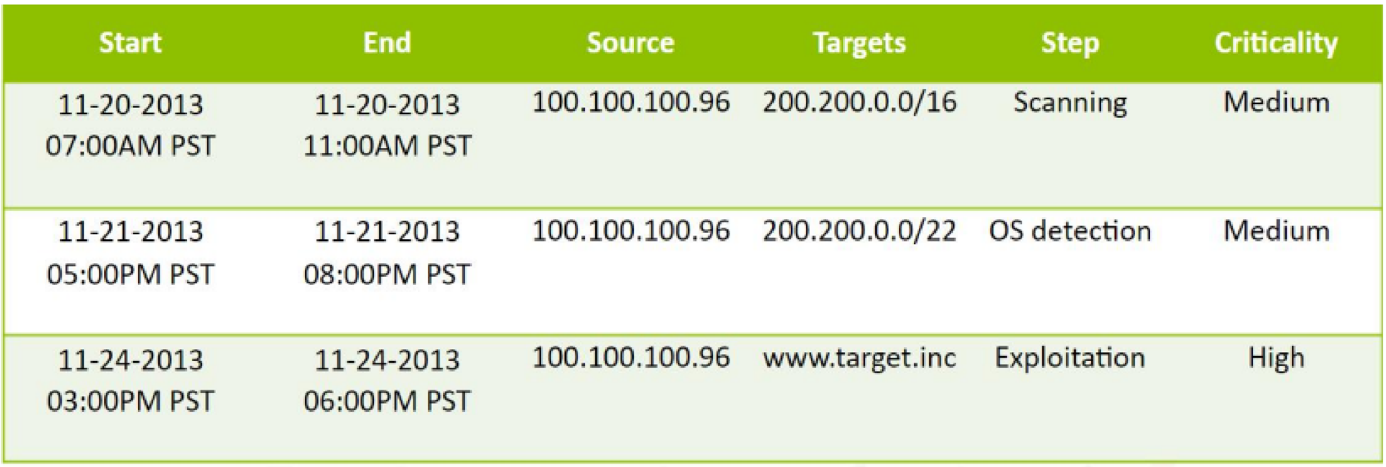
Start (format: Date-time)

Step in the process (format: text)

Criticality of test (format: text)

Source IP Address

-> example of timetable to use



Start	End	Source	Targets	Step	Criticality
11-20-2013 07:00AM PST	11-20-2013 11:00AM PST	100.100.100.96	200.200.0.0/16	Scanning	Medium
11-21-2013 05:00PM PST	11-21-2013 08:00PM PST	100.100.100.96	200.200.0.0/22	OS detection	Medium
11-24-2013 03:00PM PST	11-24-2013 06:00PM PST	100.100.100.96	www.target.inc	Exploitation	High

-> Criticality: determines whether the tests on that particular data will pose some risk of DOS or data loss to the client.
-> Pentester & client should discuss some steps if things goes badly..
-> Pentester must ensure that and dealt with in the pre-engagement phase....

Examples of liabilities & Responsibilities:

liabilities:
-> access data out-of-scop
-> accidentally remove data
-> cause Dos
-> other event with an impact to the org.

Responsibilities:
-> keep client informed up to date during the pentesting
-> keep report and collected data in a safe place (like store report & data encrypted and delete them after giving them to the client)
-> Nondisclosure of any infos

You will come across employees data, so if that happened, inform the client

Non-disclosure agreement(NDA)

it is a part of any engagement. Basically penetester guarantees, in writing, that infos will be not disclosed to any third party.

Emergency plan

to put into action, if something goes wrong. like DOS from heavy scans. So pentester should know, (write a plan better):
-> who to contact, if something wrong happen ?
-> is there a team to alert them ?
-> is there available any time ?

The allowed techniques

Should be defined to not surprise the client with a bad new
intrusive techiquest are not always allowed, so ensure them, like
bruteforce attack, social engineering, data harvesting from internet file and history, phishing attacks

The deliverables

The report, or spreadsheet documents
-> Penetester ensure the format and the documentation and start with it from Day 1

usefull website:
http://www.pentest-standard.org/

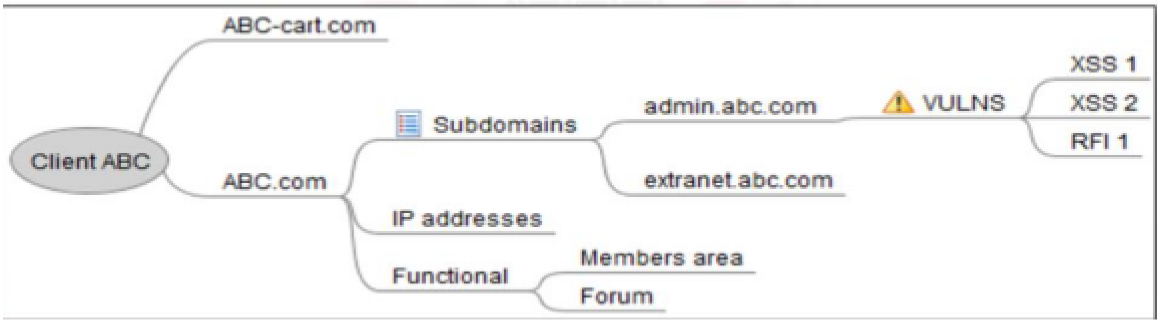
Reporting

http://www.pentest-standard.org/index.php/Reporting

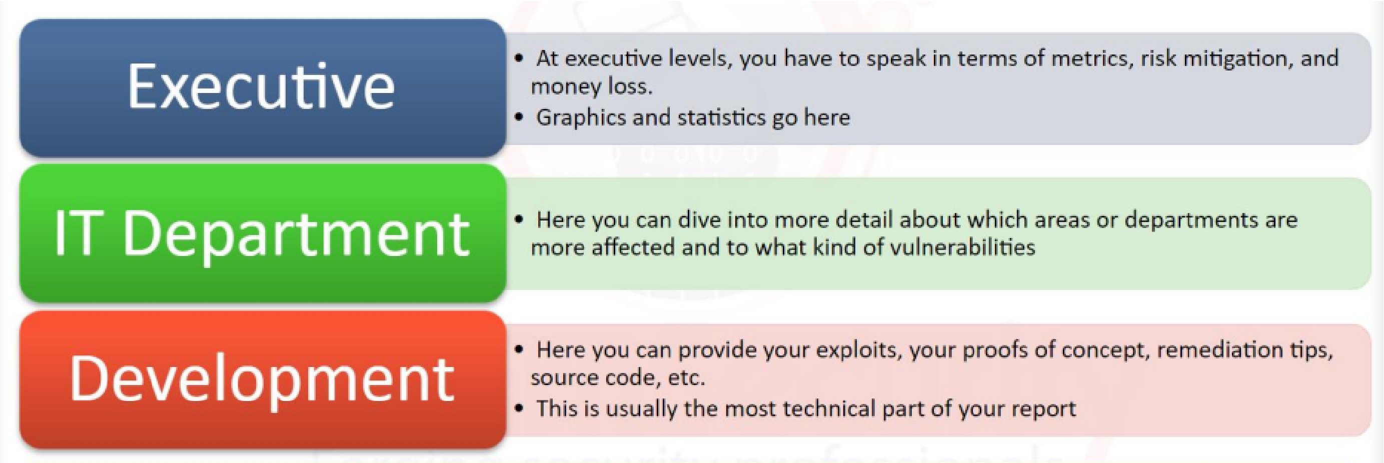
client interested in:
-> Status of the security of the assets in scop
-> what vulnerable
-> what to fix first

Report should be:
exhaustive - Clear - on-time - Good looking - adherent to client's goals

The reporting phase begins the moment you sign the rules of engagement with the client
-> firslty describe the engagement and the client goals.
-> test -> collect infos -> report
-> It is better to write the report while testing (Mind mapping tools | spreadsheets)



-> people interested in your report:
The C-level(corporate & manangers), **The IT folks**, **The developer**



Executive

- At executive levels, you have to speak in terms of metrics, risk mitigation, and money loss.
- Graphics and statistics go here

IT Department

- Here you can dive into more detail about which areas or departments are more affected and to what kind of vulnerabilities

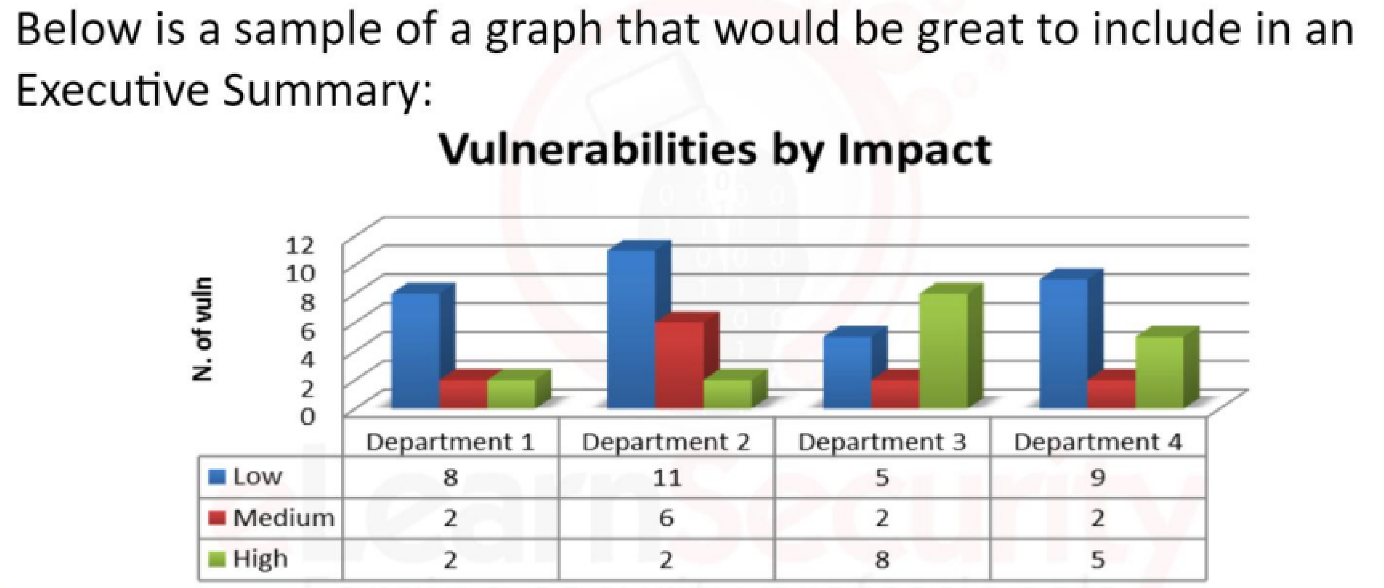
Development

- Here you can provide your exploits, your proofs of concept, remediation tips, source code, etc.
- This is usually the most technical part of your report

The Report Structure:
Executive Summary -> Vulnerability Report -> Remediation Report

Executive Summary

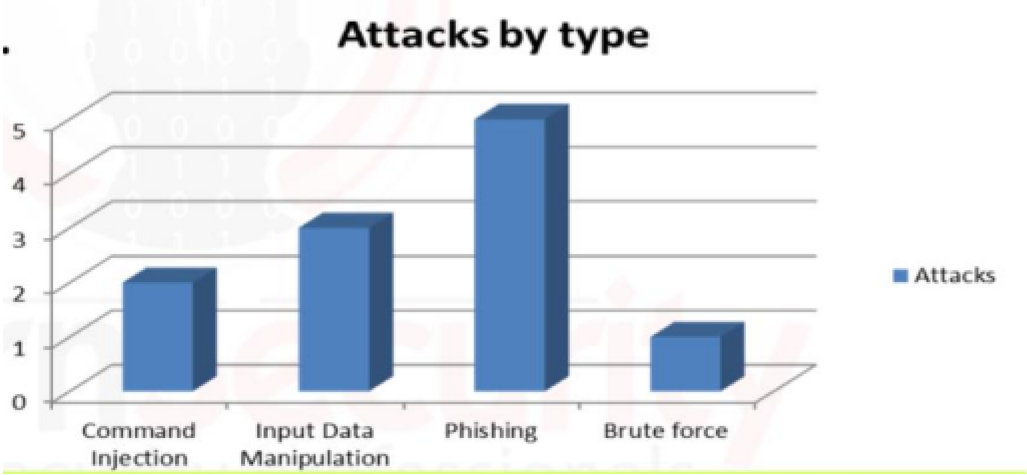
It is 2-3 pages. Here you give a brief overview about the whole engagement using graphs, charts, stats and tables. Text should only use to explain your charts. (no interested how you approach and what tools you used!!!!)



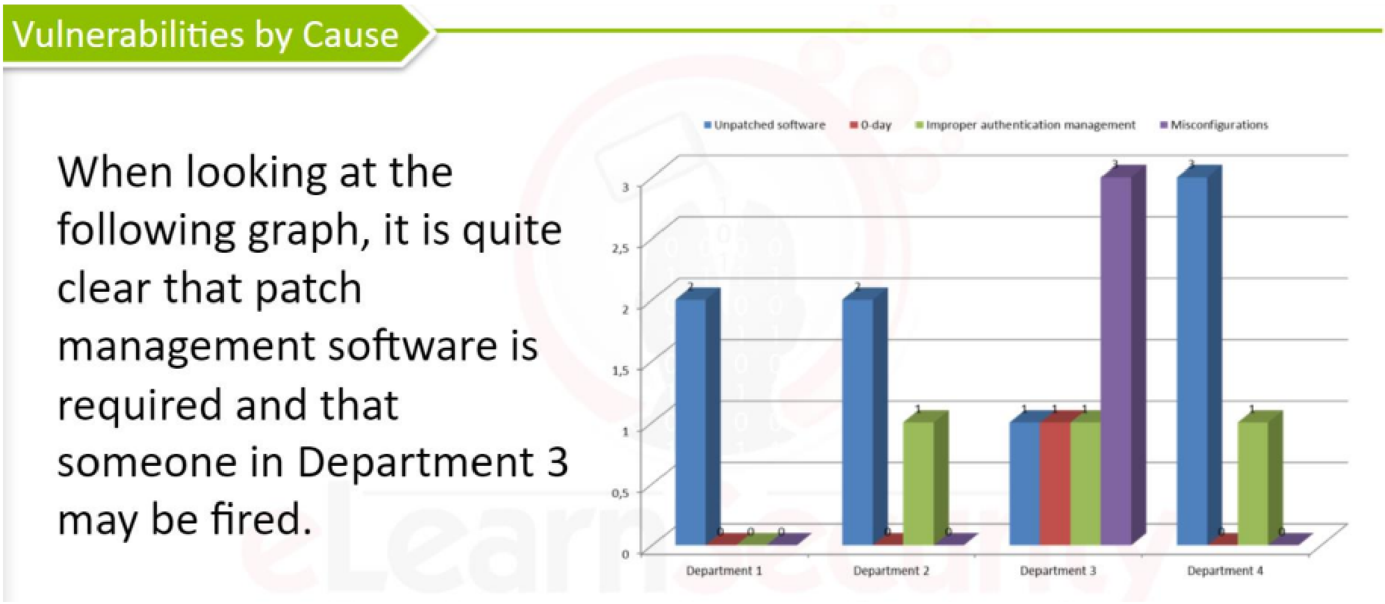
Below is a sample of a graph that would be great to include in an Executive Summary:

Vulnerabilities by Impact

	Department 1	Department 2	Department 3	Department 4
🟦 Low	8	11	5	9
🟥 Medium	2	6	2	2
🟩 High	2	2	8	5



Attacks by type



Vulnerabilities by Cause

When looking at the following graph, it is quite clear that patch management software is required and that someone in Department 3 may be fired.

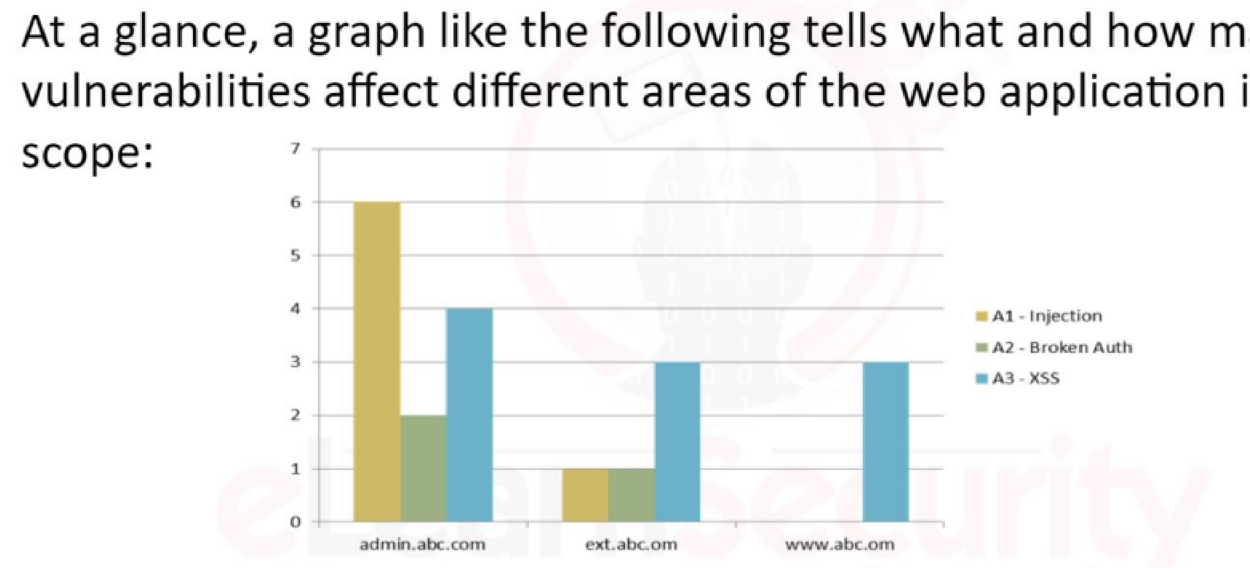
at the end of executive summary, provide an overview of the required operations like:
-> Perform input data sanitizing
-> Use stronger ciphers
-> patch software X and so on ....

Make sure, that the executive summary no more then 2-3 pages of non-technical and non-profissional exptaination level of you tests

Vulnerability Report

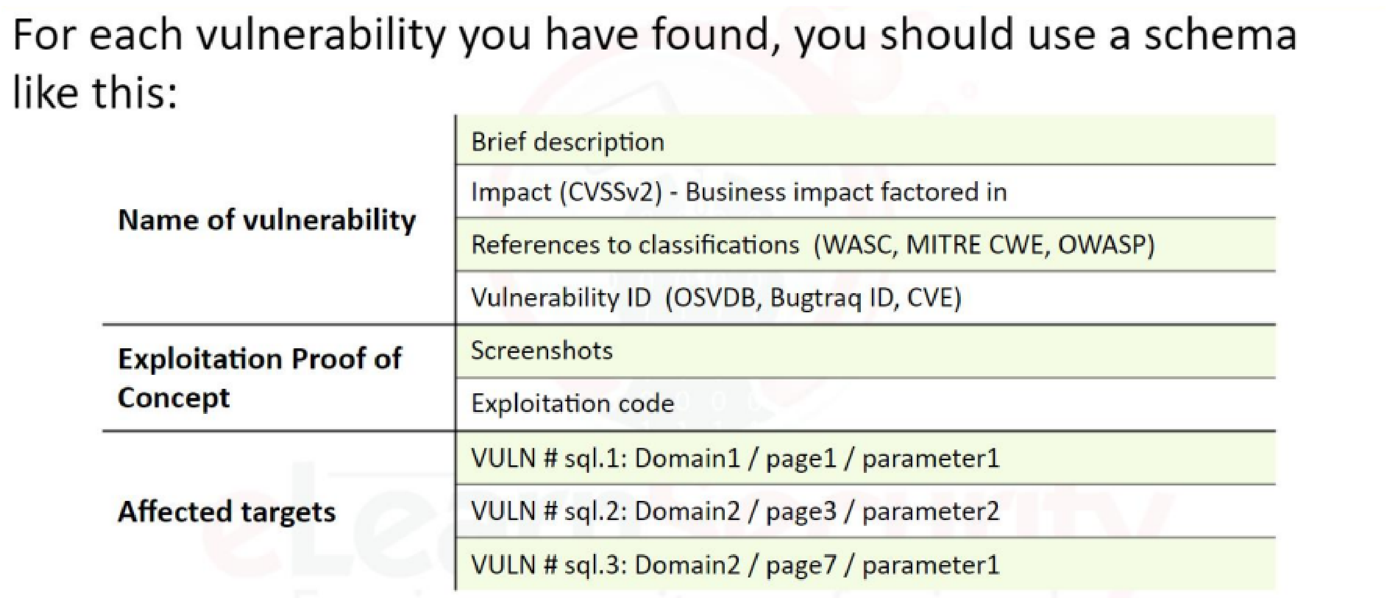
It is a technical report, will be read by technicians. It explains, what is wrong with the organization's security.

-> you can use graphs here like explaining:
vulnerablilities per item in scop or risk level per item in scope



At a glance, a graph like the following tells what and how many vulnerabilities affect different areas of the web application in scope:

Reporting Vulns by type is better, because it lets you concentrate more in the vulns and less on the target ip/server/site



For each vulnerability you have found, you should use a schema like this:

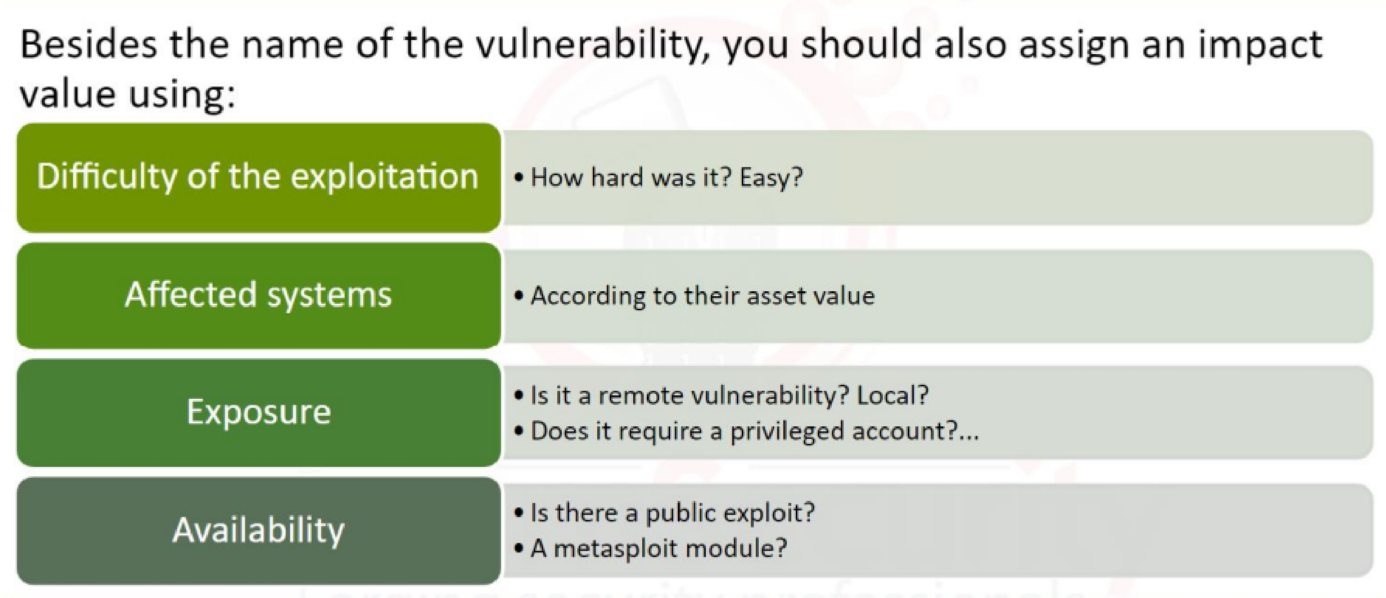
Name of vulnerability	Brief description
	Impact (CVSSv2) - Business impact factored in
	References to classifications (WASC, MITRE CWE, OWASP)
	Vulnerability ID (OSVDB, Bugtraq ID, CVE)
Exploitation Proof of Concept	Screenshots
	Exploitation code
Affected targets	VULN # sql.1: Domain1 / page1 / parameter1
	VULN # sql.2: Domain2 / page3 / parameter2
	VULN # sql.3: Domain2 / page7 / parameter1

above we have a table for all SQLi vulns.

You will make a different room for the detailed POC and inserting the reference (#sql.1, #sql.2 and so on)

-> Vulnerability name from (CAPEC or wasc threat classification)

-> vulnerability description from (NIST or OSVDB) and you can add more infos to that by your self



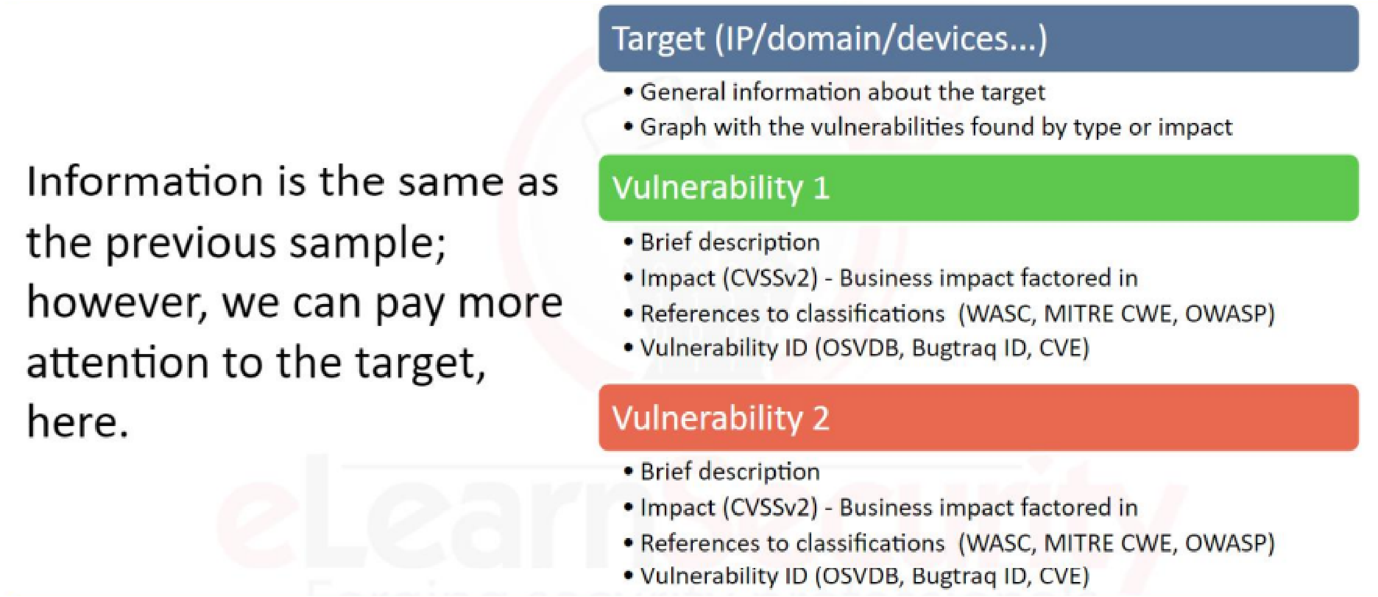
Besides the name of the vulnerability, you should also assign an impact value using:

Difficulty of the exploitation	• How hard was it? Easy?
Affected systems	• According to their asset value
Exposure	• Is it a remote vulnerability? Local? • Does it require a privileged account?...
Availability	• Is there a public exploit? • A metasploit module?

-> vulns id are important for your clients, for them to gather more infos from the internet

-> for POC include screenshots + exploit payloads

Vulns by Target:



Information is the same as the previous sample; however, we can pay more attention to the target, here.

Target (IP/domain/devices...)

- General information about the target
- Graph with the vulnerabilities found by type or impact

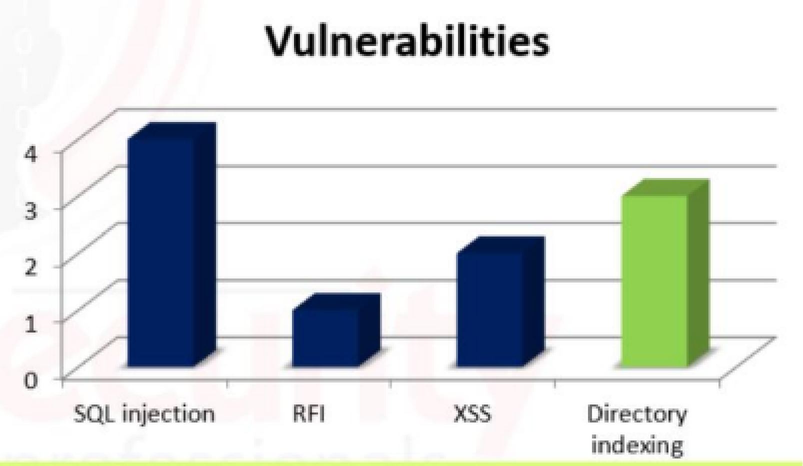
Vulnerability 1

- Brief description
- Impact (CVSSv2) - Business impact factored in
- References to classifications (WASC, MITRE CWE, OWASP)
- Vulnerability ID (OSVDB, Bugtraq ID, CVE)

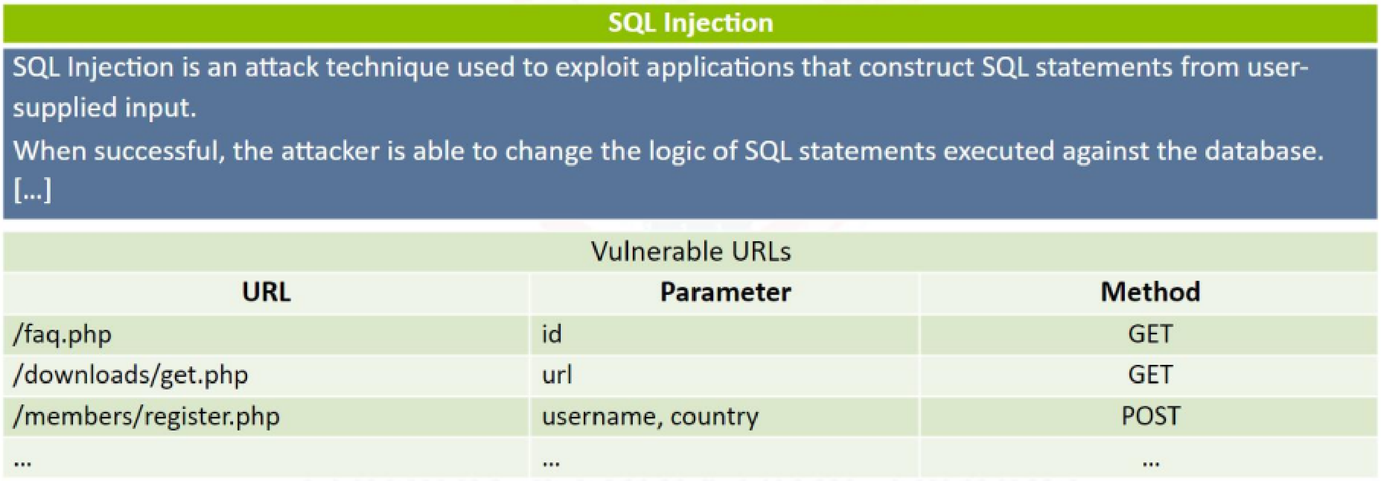
Vulnerability 2

- Brief description
- Impact (CVSSv2) - Business impact factored in
- References to classifications (WASC, MITRE CWE, OWASP)
- Vulnerability ID (OSVDB, Bugtraq ID, CVE)

-> You can include some graphs for the target domain. Use color depends on impact and same color for same impact...



-> if you have many urls infected by the same vuln, it is better to do so:



SQL Injection

SQL Injection is an attack technique used to exploit applications that construct SQL statements from user-supplied input.
When successful, the attacker is able to change the logic of SQL statements executed against the database. [...]

Vulnerable URLs

URL	Parameter	Method
/faq.php	id	GET
/downloads/get.php	url	GET
/members/register.php	username, country	POST
...	...	...

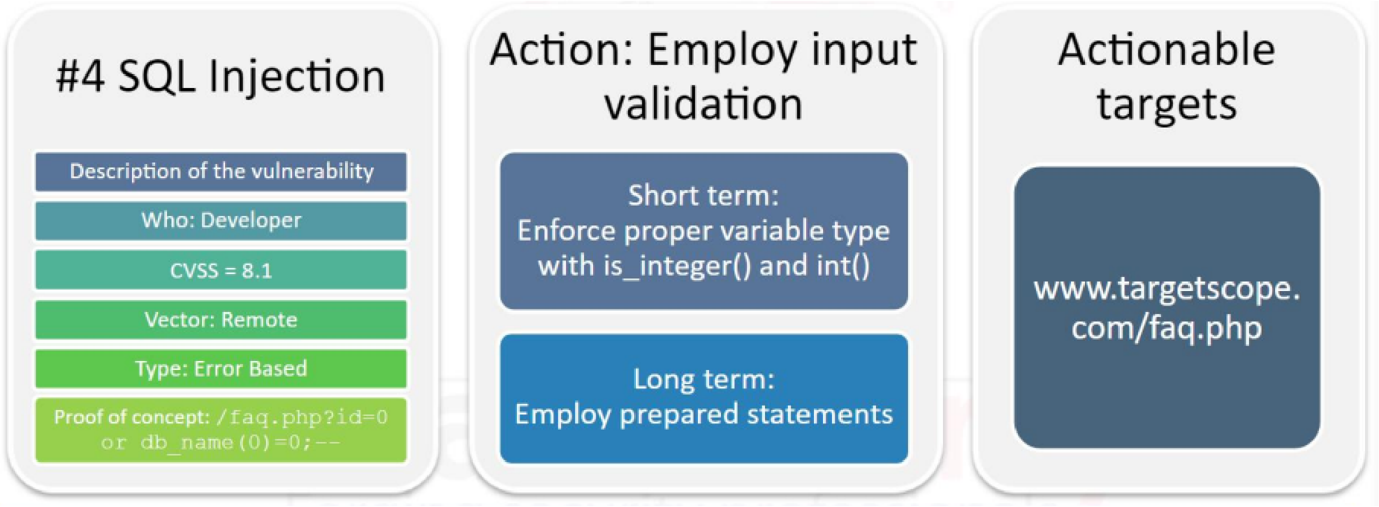
Remediation Report

-> Here pentester talk to the developer in charge to fix the vulns. You give the org. best solution for the issue.
-> Fixing start by most critical vulns. and you pentester should ask for emergency phone number to call

for any critical vuln, that need to be fixed immediately

-> if vuln has public exploit, then add reference to avialable path

-> Example:



#4 SQL Injection

- Description of the vulnerability
- Who: Developer
- CVSS = 8.1
- Vector: Remote
- Type: Error Based
- Proof of concept: `/faq.php?id=0 or db_name(0)=0;--`

Action: Employ input validation

Short term:
Enforce proper variable type with is_integer() and int()

Long term:
Employ prepared statements

Actionable targets

www.targetscope.com/faq.php

see some template in the slide 133