HideZeroOne
INE – Cyber Sec
www.hide01.ir

Web Application Penetration Testing

The Penetration Testing Process

Module 1

v3.0

OUTLINE

1.0 The Penetration Testing Process

MAP REF

1.1 Pre-engagement

1.2 Methodologies

1.3 Reporting

OUTLINE

PRE-ENGAGEMENT

1.1. Pre-engagement

A Penetration test is a complex, cyclical process of both identifying and exploiting vulnerabilities in a system.

The ultimate goal of a penetration test is to identify and assess the client organization's risk of exposure.

1.1. Pre-engagement

There is still not a unified industry standard for performing penetration tests; therefore, efforts have been made by both organizations and the open source community to establish standards. These efforts have been put forth in order to standardize the methodology and operations before, during and after the pen testing engagement.

In the next chapter, we will go through these.

1.1. Pre-engagement

Let’s look at the basic steps every professional penetration tester goes through before the engagement can begin.

1.1.1. Rules of Engagement

During the *pre-engagement phase*, the penetration tester and the client must discuss and agree upon a number of legal and technical matters.

1.1.1. Rules of Engagement

Usually, the paperwork in which all of these agreements are formalized (in writing) and signed is called the **Rules of Engagement**.

1.1.1. Rules of Engagement

This can be one or more documents with the objective to define the following:

The goal and scope of the engagement	The timeline and milestones	The liabilities/ responsibilities
The allowed techniques	The deliverables and expectations	The statement of work

1.1.1.1. The Goal And Scope

Goal and **scope** are two different terms that, during a penetration test, mean two entirely separate things.

Goal

Scope

1.1.1.1. The Goal And Scope

Goal

The **goal** of a penetration test is the very reason the client is hiring you in the first place!

So the very first question to ask the client is:

Why do you want to execute a Penetration test?

1.1.1.1. The Goal And Scope

MAP REF

Goal

You will glean a great deal of valuable information simply from the answer to the previous question; it will determine how you will deal with the entire *Pre-engagement* phase.

Performing a penetration test because the client says "*we got hacked,*" is completely different than performing a penetration test for PCI DSS compliance validation.

1.1.1.1. The Goal And Scope

IMPORTANT

Do not expect that clients know what they want (*this is a standard business rule, that certainly applies here*).

Remember that understanding their goals will help both you and the client define the *Scope of Engagement*.

1.1.1.1. The Goal And Scope

MAP REF

Scope of Engagement

The **Scope of Engagement** is exactly what you will have to (or be allowed to) test; this defines the boundaries of your tests and is critical from a legal perspective because any test beyond the defined scope is a breach of the *Rules* of Engagement and, therefore, could result in criminal charges.

The scope can be defined both logically and physically.

1.1.1.1. The Goal And Scope

MAP REF

Scope of Engagement

While the **Logical scope** is, for example, an entire department within the organization, an entire line of operations or even the whole organization.

You will want to pay attention to these kinds of poorly defined boundaries because you might find yourself considering in-scope what the client thinks is not.

1.1.1.1. The Goal And Scope

MAP REF

Scope of Engagement

The **Physical scope** is simply IP addresses, given servers (identified by IP addresses or name), domains, subdomains, autonomous systems and so on.

This is a well-defined and clear scope that you really should look for during the pre-engagement phase.

1.1.1.1. The Goal And Scope

Scope in a web application penetration test is often defined in terms of domains; therefore, the client usually will want a penetration test against a subdomain, such as www.pentest.this

At this point, you will immediately wonder (and ask) whether subdomains (such as ***intranet***.*pentest.this*, ***email***.*pentest.this*) are included and, if so, which ones.

1.1.1.2. Timetable

The timetable of a penetration test is the same that you would define for any other project, although slightly more complicated.

You will need your client to be aware of what will happen, when and where.

1.1.1.2. Timetable

MAP REF

You can either use a simple time tracking table to convey this information, or you can use GANTT charts.

This depends on your preference; however, be aware that GANTT charts are unable to convey some important information as we explore later on.

For this reason alone, a simple table or spreadsheet is preferable for most pen-testers.

http://www.gantt.com/

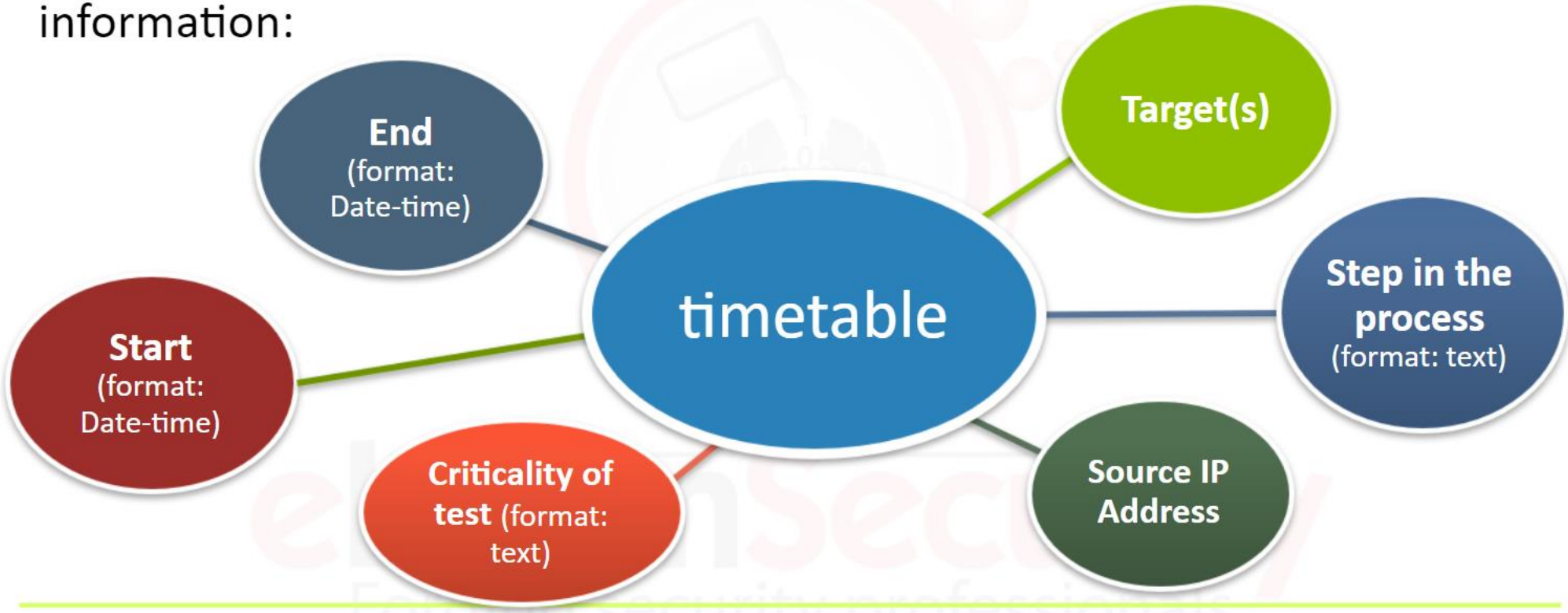
1.1.1.2. Timetable

What you need to remember is:

- Timing is usually suggested by and agreed upon with the client
- Timing of the tests can change during the engagement, as information is uncovered

1.1.1.2. Timetable

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



1.1.1.2. Timetable

As you can imagine, a GANTT chart, although good-looking and professional, will not convey nearly enough information.

1.1.1.2. Timetable

MAP REF

Here is a simple example of a timetable that you could 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

1.1.1.2. Timetable

The table on the previous slide fits best in a spreadsheet as a living document that you modify during the engagement.

Criticality determines whether the tests you are going to conduct on that particular date will pose some risk of denial-of-service, system outage or data loss to the client.

1.1.1.2. Timetable

You and the client should define your test criticality definitions and come up with emergency steps to undertake if things go badly for the client's systems.

Having the criticality level explicit in the timetable will help the client know exactly when someone in the organization should be available to initiate the previously defined emergency plan.

1.1.1.3. Liabilities and Responsibilities

A penetration test poses a number of risks for both the client and the penetration tester.

During a penetration test, things can certainly go wrong, and you will need to ensure that most of the things that you can anticipate might go wrong, are dealt with in the pre-engagement phase.

1.1.1.3. Liabilities and Responsibilities

Regardless of what can go wrong, even in the most perfect of penetration testing engagements, there are responsibilities that you will be accountable for.

Let’s see some example of liabilities and responsibilities.

1.1.1.3. Liabilities and Responsibilities

MAP REF

Possible **liabilities** could be:

- You access sensitive data out-of-scope
- You accidentally remove data
- You accidentally cause unavailability of services
- Other catastrophic event with an impact on the organization

1.1.1.3. Liabilities and Responsibilities

MAP REF

Responsibilities are:

- Keeping the client informed and up to date during your pentest
- Keeping reports and collected data in a safe place
- Following a code of ethics
- Nondisclosure of any information

1.1.1.3. Liabilities and Responsibilities

Liabilities should be dealt with by an attorney.

Your lawyer will try to eliminate any accountability for data loss and other catastrophic events unless, of course, you did them on purpose.

1.1.1.3. Liabilities and Responsibilities

Responsibilities are pretty much simple care and ethics. You will store the reports of your client encrypted and destroy them after you provide them to the client.

You will also never disclose any information you come across during the engagement to anyone.

1.1.1.3. Liabilities and Responsibilities

IMPORTANT

Please note that during the engagement you will come across employees' personal information.

In certain countries, this information is sensitive and falls under strict privacy laws and regulations.

1.1.1.3. Liabilities and Responsibilities

The entity responsible for keeping this data safe from third parties is your client. If, as a result of either exploitation or the information gathering phase, you come across data like this, you will want to have your bases covered with respect to any legal claims made against you by the company or its employees.

You should make the client aware of this and make a properly informed decision, in writing, beforehand.

1.1.1.3.1. Non-disclosure Agreements

A *non-disclosure agreement* (NDA) is part of any engagement.

Basically, the penetration tester guarantees, in writing, that discovered vulnerabilities, exploits used or developed, and, in general, any information related to the organization accessed during the engagement (not previously public), will not be disclosed to any third party.

1.1.1.3.1. Non-disclosure Agreements

This is a critical aspect for the client, and you want to make sure to provide plenty of assurance of both your ethical conduct and your respect for their confidentiality.

1.1.1.3.2. The Emergency Plan

An emergency plan is a good idea for both the pentester and the client.

It shows the client that you care and will save both of you from legal issues, should any arise as a result of your testing.

1.1.1.3.2. The Emergency Plan

An emergency plan is put into action when things go wrong during the engagement, such as: a server failing due to heavy scans, a database table being altered during an exploitation phase, or any other potentially debilitating result of our actions.

Protect yourself, and protect the client!

1.1.1.3.2. The Emergency Plan

MAP REF

An emergency plan simply involves the following factors:

- The timetable
- The contact in charge of responding to the emergency plan
- The solutions to apply to the issue

1.1.1.3.2. The Emergency Plan

The timetable or schedule of the tasks allows the client to know exactly what is going on, where, and what the criticality is for each task.

So, for instance, if *criticality* is high, the client can have a team ready and alert them quickly to act on the emergency plan.

1.1.1.3.2. The Emergency Plan

MAP REF

The emergency plan is meaningless if:

- The pentester does not know who to contact
- The contact is not readily available
- There is no written plan

Make sure to have all of the above in place before you begin the engagement.

1.1.1.4. The Allowed Techniques

Closely related to the emergency plan and how you should avoid destroying their systems, you should agree with the client beforehand which intrusive techniques you are allowed to use.

1.1.1.4. The Allowed Techniques

Defining what is allowed and what is not in advance will greatly decrease the chances of surprising the client with really bad news.

Intrusive techniques are those that not only can cause damage but also they have the possibility for serious embarrassment in the client organization.

1.1.1.4. The Allowed Techniques

The following is a list of the most common *intrusive techniques*:

- Brute force attacks
- Social Engineering
- Data harvesting of temporary internet files and history
- Phishing attacks

1.1.1.4. The Allowed Techniques

MAP REF

Social engineering is not always in scope during a penetration test. The same goes for ***phishing attacks***.

Generally exposing human weaknesses is much more embarrassing than doing the same for technology.

Businesses generally do not feel comfortable exposing these weaknesses and tend to keep this out-of-scope.

1.1.1.4. The Allowed Techniques

In order to overcome this embarrassment, you should ensure the client maximum confidentiality about the names of the employees who fell victim to such attacks during the pentest. That may help to get social attacks in-scope again.

1.1.1.5. The Deliverables

The *deliverables* of a penetration test are reports or "**the report**."

However, the client, especially for much larger engagements, might also require you to provide spreadsheet documents.

1.1.1.5. The Deliverables

Make sure that you agree about formats and documentation and *make sure you start producing it from Day 1 of the engagement*.

In the *Report* chapter, we will go through more information about how you should lay out your web application penetration test report.

METHODOLOGIES

1.2.1. PTES

MAP REF

The **Penetration Testing Execution Standard** (PTES) is an initiative being undertaken by a community of experienced penetration testers to define how penetration tests should be carried out in real-world situations.

You can find documentation produced by the PTES as it is made available: http://www.pentest-standard.org

1.2.1. PTES

MAP REF

Some interesting documents there are:

- The Pre-engagement steps according to PTES: http://www.pentest-standard.org/index.php/Pre-engagement
- The Reporting phase : http://www.pentest-standard.org/index.php/Reporting

1.2.2. The OWASP Testing Guide

The OWASP Testing guide is the effort of a number of web application security specialists to help standardize how vulnerabilities and weaknesses should be tested during an assessment.

https://www.owasp.org/index.php/OWASP_Testing_Project

1.2.2. The OWASP Testing Guide

The document, v4 at the time of writing, targets not only penetration testers but also companies willing to assess their web applications with a solid and comprehensive methodology.

While the OWASP Top 10 is based on the risks associated with web application vulnerabilities, the OWASP Testing guide contains a number of actionable tests.

https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project

1.2.2. The OWASP Testing Guide

Each test is still closely related to the most important web application weaknesses and vulnerabilities, but instead of discussing the risk, the *Testing guide* gives step-by-step instructions on how to test for the existence of the weakness.

REPORTING

1.3.1. What Do Clients Want?

The Penetration Report is the ultimate deliverable from your engagement.

By hiring a penetration tester or a penetration testing firm, a client is usually interest in:

- Knowing the status of the security of the assets in scope
- Knowing what is vulnerable
- Knowing what needs to be fixed first

1.3.1. What Do Clients Want?

Knowing that you used a reverse TCP shell once you managed to exploit an RFI may not be of critical importance to the client.

The judgment of the overall engagement is based solely upon the final report that you turn over.

1.3.1. What Do Clients Want?

MAP REF

You will want your report to be:

- Exhaustive
- Clear
- On-time
- Good looking
- Adherent to client's goals

In the next chapter, we will help you see that it is easy to write an excellent report.

1.3.2. Writing the Report

There is not a prescribed or standard structure to penetration testing reports. However, there are best practices, do's and don'ts and critical aspects that should be taken care of while writing a report.

What follows is a plethora of advice, criteria, and rules, the fruit of years of our experience in the field.

1.3.2.1. The Reporting Phase

The reporting phase begins the moment you sign the Rules of Engagement with your client; this is the right time to put together a few pages describing the engagement and the client's goals.

This Engagement Summary is something that is great to have in the Executive summary section of the report. We will see in a moment what the *Executive Summary* should look like.

1.3.2.1. The Reporting Phase

Reporting is often (mistakenly) believed to be the last chronological step of an engagement.

This is both incorrect and dangerous.

1.3.2.1. The Reporting Phase

When you create your report as the last step, you will forget important details. Also, if the engagement is big enough or time has run out, you might simply not be allowed to perform tests anymore. Now you are not able to double-check this software revision or that open port.

This does not mean that during your tests you have to keep stopping to write your report!

1.3.2.1. The Reporting Phase

This is a more appropriate process:

1.3.2.1. The Reporting Phase

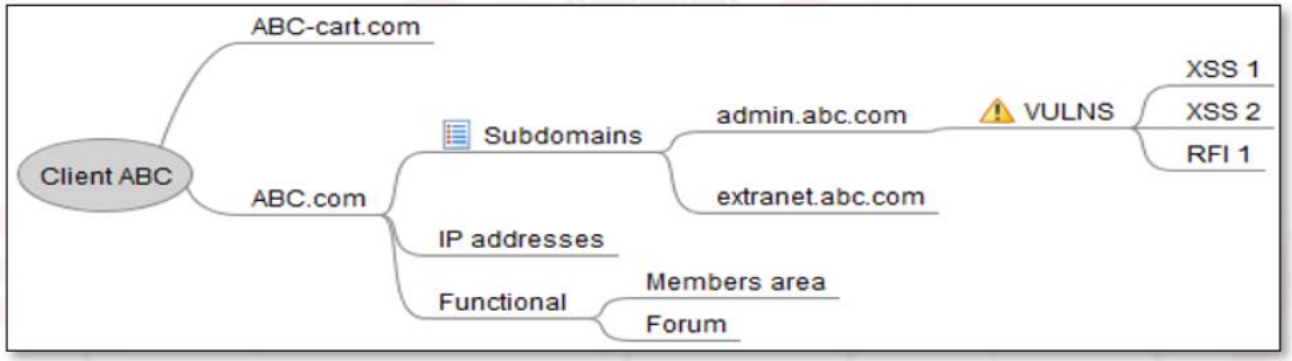
While you perform your tests, you will have to collect and organize information methodically.

This information will be the core of your report. So, by collecting and storing this information in a precise and organized way, you will have already contributed to the reporting phase. At the end, you simply need to gather and present this information in a readable and professional format.

1.3.2.1. The Reporting Phase

Mind mapping tools and spreadsheets are the best two ways to store information with structure and relationships.

The following is an example of how we can keep track of information about the organization:



1.3.2.1. The Reporting Phase

Freemind is a free mind mapping tool that allows you great flexibility in the many ways that you can store this information, still within the boundaries of a hierarchical tree view.

1.3.2.1. The Reporting Phase

MAP REF

You can create a structure that adapts to your needs or to the type of engagement.

Each node can also contain a reference to an external file, which comes in handy when you want to attach the payload of an attack or the XML result file of a Nmap scan.

You can download the tool here.

http://freemind.sourceforge.net/wiki/index.php/Download

1.3.2.2. Understanding your Audience

The report can live in many different structures and formats.

However, remember that there are pretty much 3 types of people interested in reading your report:

1. The C-level
2. The IT folks
3. The developers

1.3.2.2. Understanding your Audience

If the client is a small business, 2 (IT folks) and 3 (developers) may be the same person wearing multiple hats.

In any case, you will want to create as many sections as the types of audience your report needs to have.

1.3.2.2. Understanding your Audience

As stated earlier, the C-level (corporate and managers) will not really care what kind of exploits you used, even though they are the level that promoted (or at least allocated the budget for) the penetration test in the first place.

A CISO might want to know what advancements have been made since the last penetration test, or what needs to be mitigated and what costs are involved. *Money*, *man-hours* and *time* are the metrics at this level.

https://en.wikipedia.org/wiki/Chief_information_security_officer

1.3.2.2. Understanding your Audience

Conversely, a developer will not care about the graphs and the money involved in implementing a patch or a workaround. S/He will only care about what to patch and how.

So, the trick to create a perfect report is to build your report in different sections with different language and terminology as they are understood by each part of the client's organization.

1.3.2.2. Understanding your Audience

MAP REF

Below is a typical set of your penetration test report's target audience groups:

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

1.3.2.3. The Report Structure

Once you understand who will read your report and what kind of information they are actually interested in, you can match your target audience groups with the actual report structure.

1.3.2.3. The Report Structure

A typical structure is laid out like this:

1.3.2.3.1. The Executive Summary

The Executive Summary is where you talk to the corporate types or, in general, give a brief and concise overview about the whole engagement.

If you agreed to use metrics meaningful to the organization, make sure that you make the current level of security, according to these metrics, visible in the first two pages of your executive summary.

The executive summary should be no more than 2 - 3 pages.

1.3.2.3.1. The Executive Summary

An introduction to the Executive summary could read like this:

> "The purpose of this assessment and report is to identify any web application issues that could affect ABC, Inc.'s e-commerce application and the web server hosting it, and to provide solutions to remedy these same issues."

1.3.2.3.1. The Executive Summary

The Executive level of a company has zero time to invest in your philosophical approach to security.

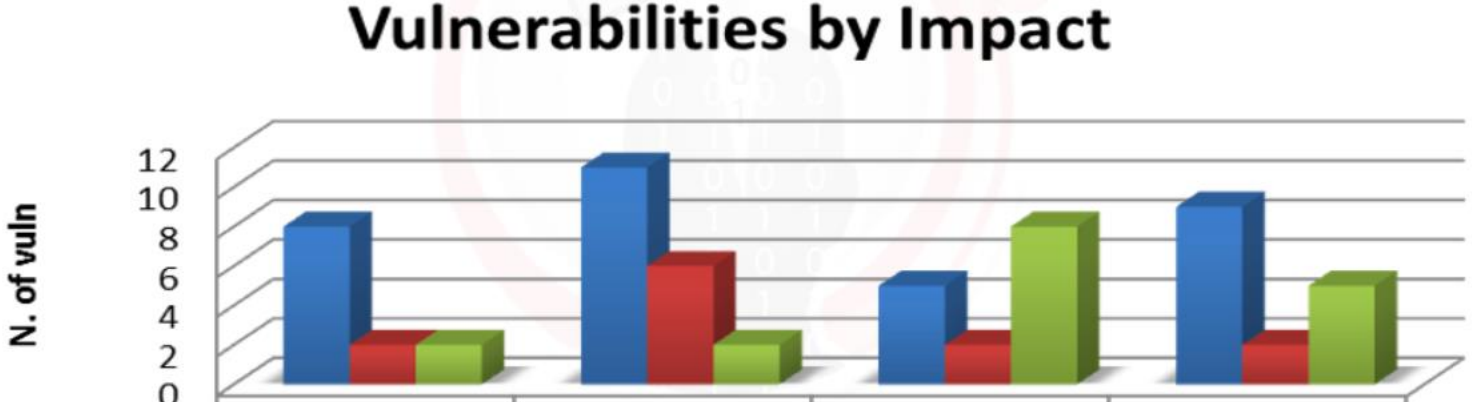
Nor it is interested in which open source "*pwnsauce*" tool you used.

You should, instead, use graphs, charts, stats and tables. Text should only be used to explain your charts and to give a final estimation on the state of security.

1.3.2.3.1. The Executive Summary

MAP REF

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



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

1.3.2.3.1. The Executive Summary

MAP REF

Risk Exposure Over Time

The importance of working with metrics lies in the fact that we can reliably measure the level of security over time and act accordingly.

This is critical to your client, especially at their executive level. They are interested in knowing how effective their security expenditure is and what would be required to lower said risk to an acceptable level.

1.3.2.3.1. The Executive Summary

MAP REF

Risk Exposure Over Time

You should understand what the management level is expecting - concrete and viable proof that by hiring you and following your remediation guidelines, they have lowered their risk.

If yours is an intrusion test to check whether the remediation plan of a previous report has been carried out properly, you will want to include a graph with the level of risk over time.

1.3.2.3.1. The Executive Summary

MAP REF

Risk Exposure Over Time

Of course, you would need to use past reports to produce an estimate of the current risk compared to previous risk level.

This is a critical graph, but you cannot get it wrong. We advise you to include it only if you are familiar with risk assessment principles and terminology.

1.3.2.3.1. The Executive Summary

The executive level may be interested in understanding what kinds of attacks you have been able to carry out against each asset of the organization.

A good classification of vulnerabilities and attacks is the MITRE Common Attack Pattern Enumeration and Classification (CAPEC).

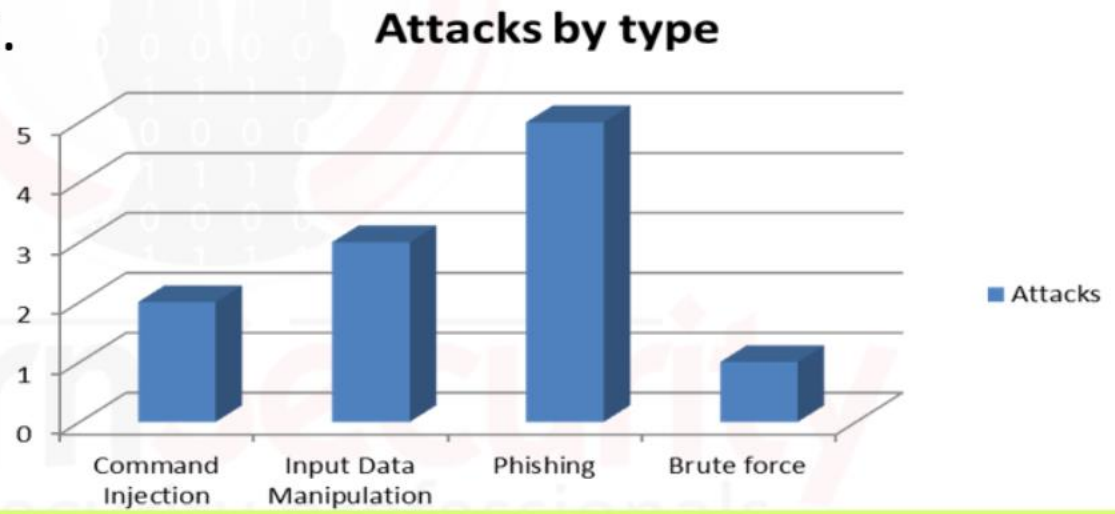
http://capec.mitre.org/data/lists/2000.html

1.3.2.3.1. The Executive Summary

MAP REF

Successful Attacks By Type

It offers a comprehensive dictionary on attacks along with useful information (description, vocabulary terms, reference links) that you can use in your report.



1.3.2.3.1. The Executive Summary

You can create a similar graph charting your vulnerabilities by type.

You can refer to WASC Threat classification for a good list of vulnerabilities and their common names.

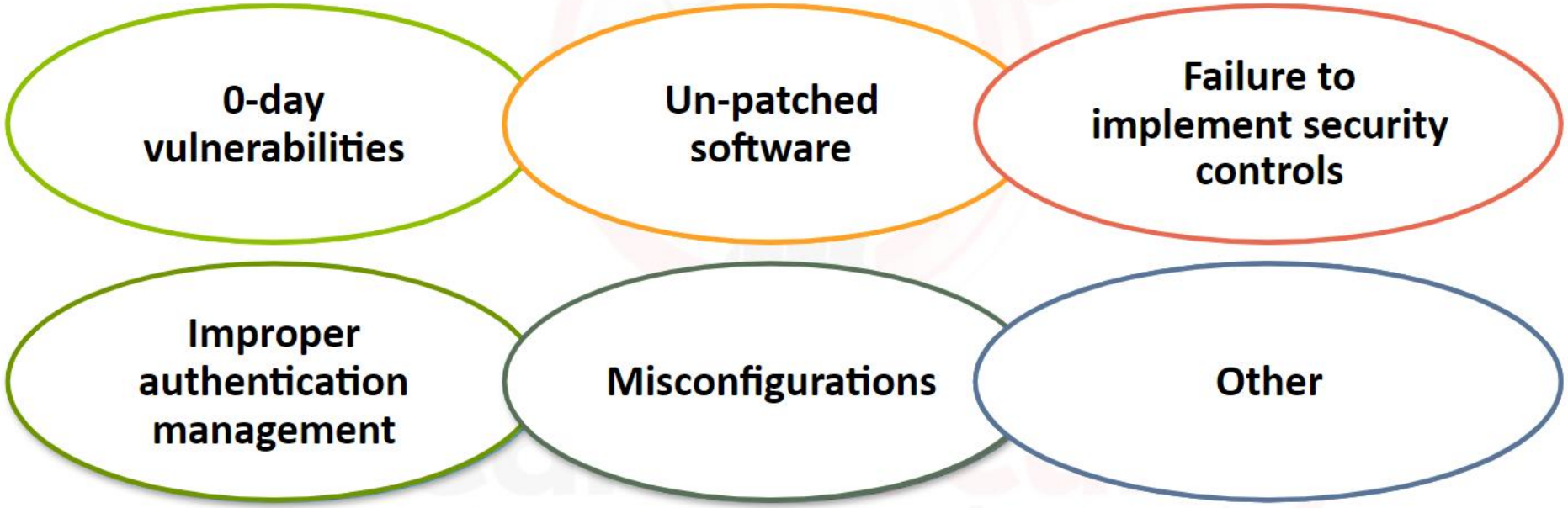
http://projects.webappsec.org/Threat-Classification

1.3.2.3.1. The Executive Summary

MAP REF

Vulnerabilities by Cause

You can exploit a system because of:



1.3.2.3.1. The Executive Summary

Vulnerabilities by Cause

The aforementioned are means that a threat agent (a hacker, an insider, the competition) can use to violate your client's security.

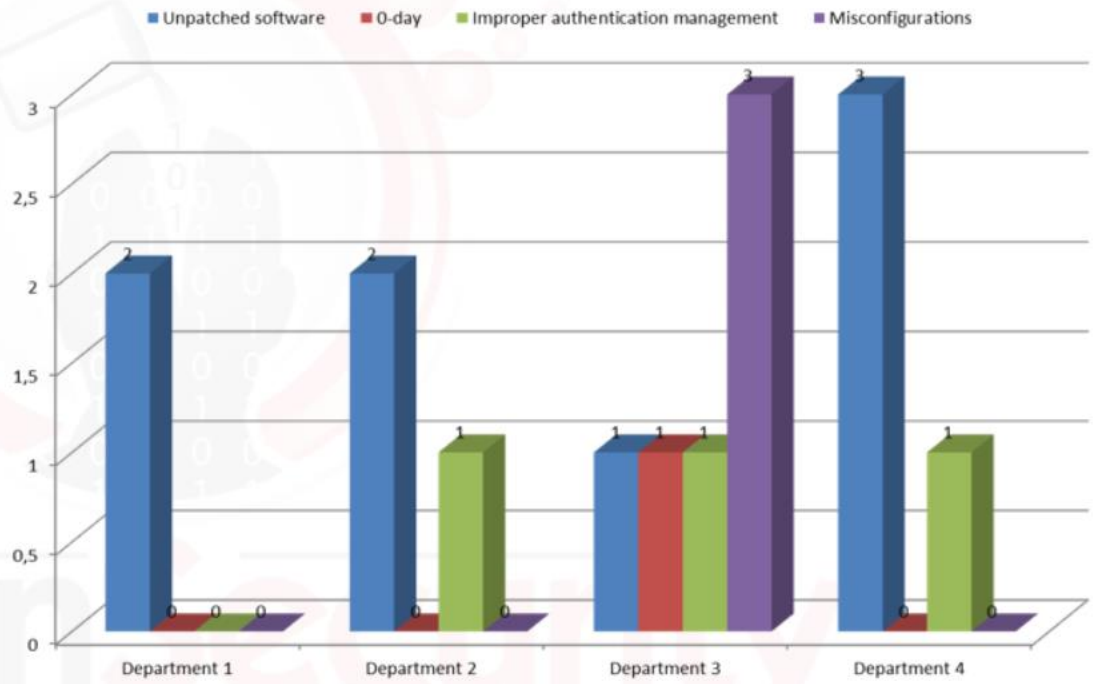
Charting vulnerabilities by their root cause will allow the executives to identify responsibilities and take actions.

1.3.2.3.1. The Executive Summary

MAP REF

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.



1.3.2.3.1. The Executive Summary

Vulnerabilities by Cause

At the end of your charting journey, it is important to touch on the amount of work required to fix all the discovered issues.

Executives and managers think best about work in terms of man-hours.

1.3.2.3.1. The Executive Summary

MAP REF

Vulnerabilities by Cause

If you can estimate this effort, excellent!

However, be wary as you are usually not aware of the inner processes or the difficulties that a fix can cause in an unfamiliar environment.

If you are performing your penetration testing internally (for your own company), you may add a fairly close estimate of the man hours.

1.3.2.3.1. The Executive Summary

MAP REF

Vulnerabilities by Cause

Otherwise, you should probably just provide an overview of the required operations:

- Perform input data sanitizing
- Use stronger ciphers
- Patch software X
- And so on…

1.3.2.3.1. The Executive Summary

Vulnerabilities by Cause

Once again, be certain that the Executive Summary contains no more than 2 - 3 pages of non-technical, professional, managerial-level presentation of your penetration test outcome.

1.3.2.3.2. Vulnerability Report

The Vulnerability Report is often also called the Technical report and is where each vulnerability found in the web application is dealt with in greater detail.

This part will be read by technicians (sometimes even managers) and it is used to explain in specific detail what is wrong with the organization's security.

1.3.2.3.2. Vulnerability Report

You will be able to talk in technical terms about specific vulnerabilities, exploitations, affected hosts (or domains or...anything in-scope), attack vectors, etc.

1.3.2.3.2. Vulnerability Report

MAP REF

To introduce this section you can still use graphs as long as they are relevant to the recipients of the document.

Typical graphs would be:

- Vulnerabilities per item in scope
- Risk level per item in scope

1.3.2.3.2. Vulnerability Report

The names you use in the graphs are critical. We advise that you take these names from a well-known taxonomy or, more specifically, the taxonomy you will use throughout the report.

Good examples for web applications are *OWASP Top 10* and *WASC TC*.

1.3.2.3.2. Vulnerability Report

MAP REF

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

1.3.2.3.2. Vulnerability Report

You have different options about how you want to arrange the information within this section.

In web application penetration tests, you often have a small number of vulnerabilities affecting a large number of different pages on different domains.

1.3.2.3.2. Vulnerability Report

In other cases, you will have only one target in scope affected by a large number of different vulnerabilities.

Be prepared to adjust your layout according to each situation.

In the first case, you would report vulnerabilities by their type.

1.3.2.3.2.1. Vulnerabilities by Type

Reporting on vulnerabilities by type lets you concentrate more on the vulnerability and less on the target IP/Server/Website affected.

1.3.2.3.2.1. Vulnerabilities by Type

MAP REF

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

1.3.2.3.2.1. Vulnerabilities by Type

In the previous layout, you will only have one table for all the SQL injections vulnerabilities found and multiple *"Affected targets."*

So, in this case, there is a one-to-many relationship between the vulnerabilities and the places where you found it.

1.3.2.3.2.1. Vulnerabilities by Type

Please note that in this case, you will have to make room for the detailed proof of concept for each of the affected targets.

We recommend that you insert a reference (#sql.1, #sql.2 and so on) in order to delve into each in a separate chapter or appendix.

1.3.2.3.2.1. Vulnerabilities by Type

Once again, the name of the vulnerability should be one of the WASC Threat Classification, MITRE CWE or, for application-specific vulnerabilities, OWASP Top 10.

Make sure to choose one of the above, according to your scope, and stick with it religiously.

http://projects.webappsec.org/f/WASC-TC-v2_0.pdf
http://cwe.mitre.org/data/lists/2000.html

1.3.2.3.2.1. Vulnerabilities by Type

The description of the vulnerability can be taken from the aforementioned sources, while if the vulnerability is in common off-the-shelf software, you should include its description from NIST or OSVDB.

https://blog.osvdb.org/

1.3.2.3.2.1. Vulnerabilities by Type

Do not forget that it is fine if you want to add further explanation when the description sounds meaningless or is too generic.

Make sure that your description is always relevant to your client environment and that it provides clear information pertaining to the specific situation you encountered during tests.

1.3.2.3.2.1. Vulnerabilities by Type

MAP REF

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?

1.3.2.3.2.1. Vulnerabilities by Type

The *OWASP Top 10* already assigns an impact to each Risk.

Remember to make this impact meaningful for your client by adjusting this metric with the value that the affected asset has in the client's business (Business Impact).

1.3.2.3.2.1. Vulnerabilities by Type

MAP REF

The vulnerability ID is important to your client organization so that they may gather more information or for statistical purposes.

You can find a vulnerability ID for publicly known vulnerabilities into software, devices and OS's.

1.3.2.3.2.1. Vulnerabilities by Type

The Mitre CVE-ID's are "*unique, common identifiers for publicly known information security vulnerabilities.*"

This means that for known vulnerabilities, you should include the corresponding ID from at least Mitre CVE and OSVDB, including a link to their page.

1.3.2.3.2.1. Vulnerabilities by Type

The Exploitation proof should not be too long. It is intended for your audience to understand how the exploitation was carried out, and, more specifically, to the development team to reproduce it.

You should include:

- Snapshots
- Exploit payloads

1.3.2.3.2.1. Vulnerabilities by Type

Exploit payloads can include HTTP request and response headers in the case of web application testing. If you can, only include the vulnerable parameter, the HTTP method and the payload used. Only add full headers if they are required to reproduce the exploitation.

For example: the vulnerability requires an authenticated session or a particular cookie value in order to be successfully exploited.

1.3.2.3.2.1. Vulnerabilities by Type

MAP REF

You basically want to keep redundant information at a minimum, while at the same time, remembering that you will need to further explain the exploitation if it is not straightforward or already documented elsewhere.

1.3.2.3.2.1. Vulnerabilities by Type

Remember that:

Including the 1000 SQL queries that gave you the password of a user in the organization's database is not very relevant.

Instead, you would just prove the existence of a SQL injection vulnerability through a very simple proof-of-concept injection query.

Including many different outcomes for one exploitation is not always considered a value. Just include the most important. For example, a SQL injection can lead to data dump, data modification, XSS, impersonation, access to file system… Just delve into the ones that are most impactful for the business and only mention the others.

1.3.2.3.2.2. Vulnerabilities by Target

If vulnerabilities are heterogeneous, or the number of targets in scope is little, you can arrange the above information on a per-target basis (instead of reporting vulnerabilities by type).

1.3.2.3.2.2. Vulnerabilities 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)

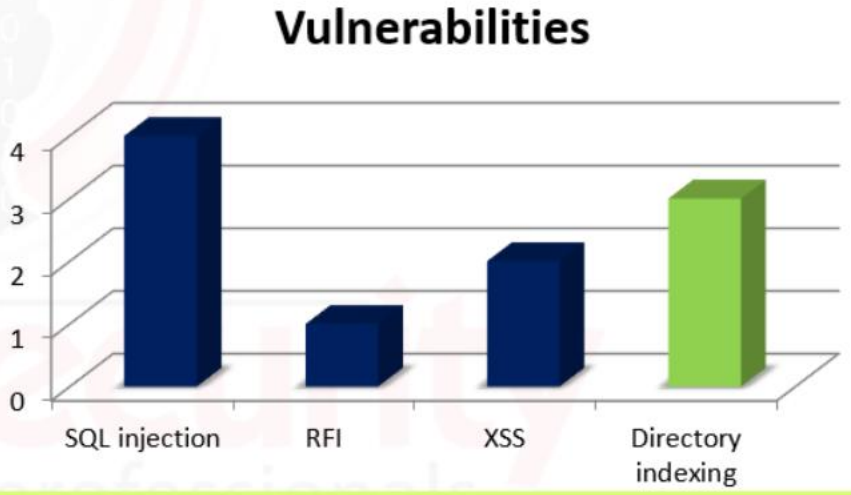
Vulnerability 2

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

1.3.2.3.2.2. Vulnerabilities by Target

You can include a couple of graphs to highlight the types of vulnerabilities found on the single target.

For example, if the target is a web application or the target is a domain, you could show the following graph:



1.3.2.3.2.2. Vulnerabilities by Target

You can use different colors that correspond with the impact level.

Just be sure that you use the same colors for each impact level throughout the report.

1.3.2.3.2.2. Vulnerabilities by Target

When the scope of the test is a web application with many URLs, you will want to pick the Vulnerability by type model (sec. 2.2.2.3).

In this case, you will have a section for each vulnerability and then list all the URLs that are affected by that particular vulnerability.

1.3.2.3.2.2. Vulnerabilities by Target

Consider the following an incomplete skeleton of the schema you can use when your scope includes a web application.

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
…	…	…

1.3.2.3.3. Remediation Report

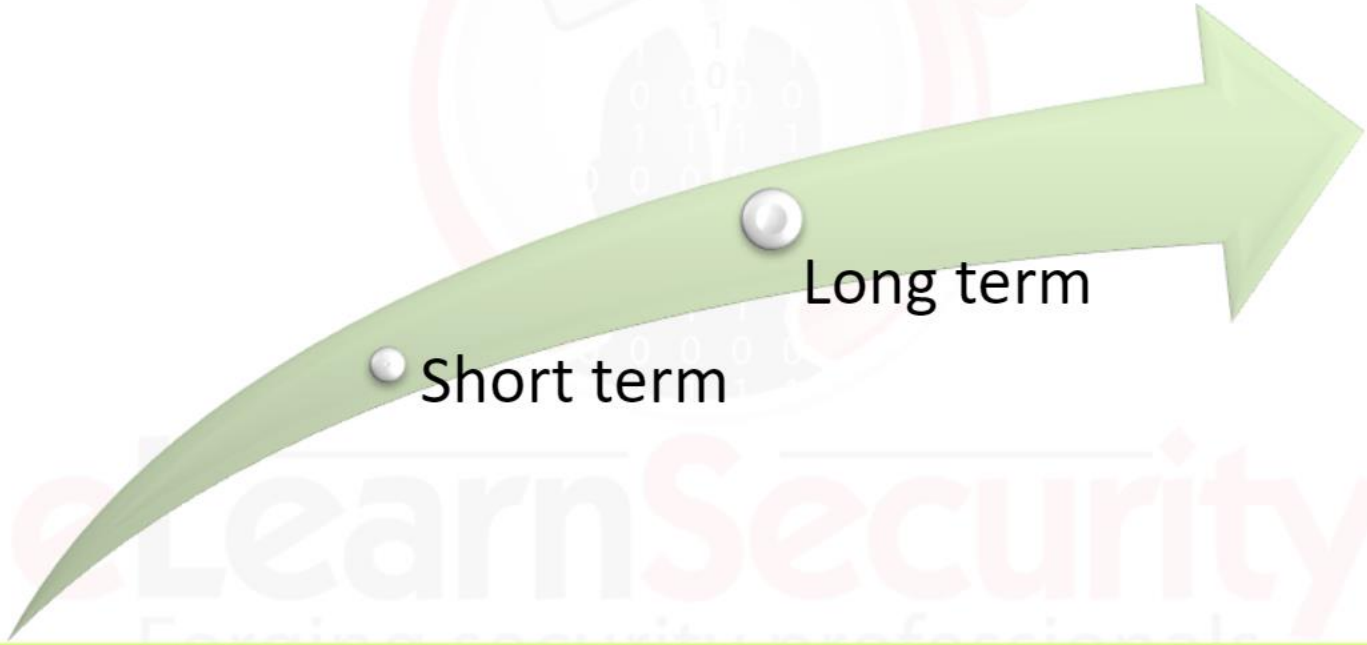
The Remediation report is the place to talk to the developers in charge of fixing the vulnerabilities that you have proved exploitable in the *Vulnerability report*.

This is where you can prove yourself a real professional and not just a hacker; you will help the organization to find the right solution to their issues.

1.3.2.3.3. Remediation Report

MAP REF

In this section, you can work on two different time horizons:



1.3.2.3.3. Remediation Report

In the **short term**, you want the remediation team to address the most important vulnerabilities as soon as possible.

1.3.2.3.3. Remediation Report

You may suggest that your client provides you with an emergency phone number where you can immediately call a developer, should you discover vulnerabilities that put the organization's vital assets at risk.

This is very short-term; sometimes it takes weeks or months from the beginning of the tests to the release of the deliverables. You will want to persuade the client to address these high-impact issues as soon as possible.

1.3.2.3.3. Remediation Report

You can also suggest **long-term** actions, like:

- Implementation of SSDLC (Secure Software Development Lifecycle)
- The employment of security checks early in the business or development processes
- Or the use of different platforms, versions or frameworks

1.3.2.3.3. Remediation Report

Long-term actions will bring benefits in the long run but are generally things the organization will not accomplish in the next 6 - 12 months and certainly not without a good chunk of investment of both time and money.

1.3.2.3.3. Remediation Report

Your job is to increase security (and lower the risk) for your client organization, not to merely exploit their machines.

1.3.2.3.4. The Remediation

Providing suggestions on how to remediate common vulnerabilities is usually pretty trivial.

If the vulnerability exploited was in a publicly available web application, you will just add references to available patches, upgrades, hotfixes or workarounds.

1.3.2.3.4. The Remediation

You usually find all you need within vulnerability databases or in the official security advisories.

If the application is custom-coded, you will have to suggest patches to the code or solutions according to the type of vulnerability and the web application environment.

Example:

An input validation vulnerability can be solved by properly handling input dat

1.3.2.3.4. The Remediation

How you arrange the information in this section should reflect the approach you used during the vulnerability report.

If you have listed your vulnerabilities by type, you will do the same here.

1.3.2.3.4. The Remediation

You will have to **prioritize** your remediation plan according to the impact level that you assigned and stuck within the vulnerability report.

Make sure that the first issues on your list are the most important.

Double-check and adjust your priority based on common sense and use your experience as well.

1.3.2.3.4. The Remediation

MAP REF

Below is an example of a remediation item is in the following schema:

#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

1.3.2.3.4. The Remediation

The previous table is just an example, of course.

You will definitely want to add more details like snippets of code that are relevant to the languages and servers in use on the client environment.

1.3.3. Report templates and guides

Here you can find some useful report templates and guides:

- eLearnSecurity Reporting Guide
- Cure53 - Pentest Report

https://members.elearnsecurity.com/course/resources/name/wapt_v3_section_1_module_1_attachment_Reporting_guide

https://cure53.de/#publications

REFERENCES

References

Gannt chart

http://www.gantt.com/

PTES Main page

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

Pre-engagement

http://www.pentest-standard.org/index.php/Pre-engagement

Reporting

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

OWASP Testing Project

https://www.owasp.org/index.php/OWASP_Testing_Project

OWASP Top 10

https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project

Freemind

http://freemind.sourceforge.net/wiki/index.php/Download

CAPEC List

http://capec.mitre.org/data/index.html

References

Threat classification

http://projects.webappsec.org/Threat-Classification

Open Source Vulnerability Database

http://osvdb.org/

Cure53 - Pentest Report

https://cure53.de/#publications

Threat classification (PDF)

https://members.elearnsecurity.com/course/resources/name/wapt_v3_section_1_module_1_attachment_WASC-TC-v2_0

eLearnSecurity Reporting Guide

https://members.elearnsecurity.com/course/resources/name/wapt_v3_section_1_module_1_attachment_Reporting_guide