PenTest magazine

WRITING AN EFFECTIVE PENETRATION TESTING REPORT

ISSN: 2084-1117

Managing Editor: Anna Kondzierska
anna.kondzierska@pentestmag.com

Proofreaders & Betatesters: Lee McKenzie, Daniela, Jay Kay, Ayo Tayo-Balogun, Paul Oyola, Jonus Gerrits, Hammad Arshed, Mateo Martinez, Christopher Pedersen

Special thanks to the Betatesters & Proofreaders who helped with this issue. Without their assistance there would not be a PenTest Magazine.

Senior Consultant/Publisher: Pawel Marciniak

CEO: Joanna Kretowicz
joanna.kretowicz@pentestmag.com

DTP: Anna Kondzierska

Publisher: Hakin9 Media Sp.z o.o. SK 02-676 Warsaw, Poland
ul. Postepu 17D
Phone: 1 917 338 3631 *www.pentestmag.com*

DISCLAIMER!

The techniques described in our articles may only be used in private, local networks. The editors hold no responsibility for misuse of the presented techniques or consequent data loss.

Contents

Dear PenTest Readers,

We would like to proudly present you the newest issue of PenTest. We hope that you will find many interesting articles inside the magazine and that you will have time to read all of them.

We are really counting on your feedback here!

In this issue you will see different approaches to writing an effective penetration testing report. You can read both technical articles and those which provide key insights. You will see 10 different approaches and an interview with Mihai Raneti- CEO of CyberFog. We will give you answers to questions:
- why is a penetration testing report so important?
- who is it for?
- and the most important: what should the report contain?

The main aim of this issue is to present our publication to a wider range of readers. We want to share the material we worked on and we hope we can meet your expectations. With free account you have access to all the teasers and open issues, but we fully believe that you'd like to take this one step further and enjoy our publications without limits. Our premium subscription contains access to our whole archive.

The virtual doors to our library are open for you!

We would also want to thank you for all your support. We appreciate it a lot. If you like this publication you can share it and tell your friends about it! every comment means a lot to us.

Again special thanks to the Beta testers and Proofreaders who helped with this issue. Without your assistance there would not be a PenTest Magazine.

Enjoy your reading,
PenTest Magazine's
Editorial Team

Approach by Lorenzo Vogelsang

Table of Contents

Introduction

A report “is a statement of the results of an investigation or of any matter on which definite information is required”(Oxford Dictionary). The penetration testing in particular is the ultimate outcome you can deliver to a client after the “technical” penetration testing process is completed. An effective pentest report should document all the security discoveries and a thorough remediation plan so that the client's overall security could be improved at a later stage.

Some questions from a client’s perspective could be:

- *How could my business be impacted from an external/internal attack?*
- *What kind of information could attackers steal?*
- *How much money could the business lose if attackers gain access to some of the critical services/data?*

These kind of questions and their respective answers are related and subjective to their Threat Model, and how and why core assets are defined in the company. Defining and talking about Threat Modeling processing and policies is not in the article’s scope but I personally think it’s important to keep that in mind because it’s inseparable from business impacts on the one hand and from the pentest report on the other.

In fact, the final report is not just a “piece of paper” but instead, it’s a real and tangible commercial and business object; a report is the demonstration that the penetration tests were worth the cost.

In order to make it as effective as possible, we have to communicate to the client its real value and in order to achieve that, we should have always in mind these general guidelines:

- The pentest is what the client has paid for, so we should treat the report as the “final product” delivered to the Client (also because the client has no clue about the pentest “behind the scene”)
- As a general report contains “definite information”, we should set up the report in a more scientific and deterministic way documenting exactly every security test performed on targets and what we did to exploit them. We should also mention the methodology used to conduct the audit and what tools we have used and how. Everything has to be documented and always repeatable.

What do clients want?

Assuming the prerequisite of technical perfection, the major aim of a successful pentest report is to answer, as much as possible, the critical motivations that are the real cause and motivation of the final report. Usually, almost every client wants to know the status of the security of their critical assets (they could be systems, applications or data) and, if applicable, any improvement compared to the results of previous pentests. Once the overall security status is identified, it’s important to communicate what core systems or applications are susceptible to attack, adding an effective remediation plan that defines what systems/applications need to be fixed first.

In summary, we need to effectively communicate to the client their actual security/risk level, what is impacted and how to successfully remediate. The final goal in the long term is to make a significant contribution to the overall client's security.

The five Ws

As a story is not complete without answering the five Ws (some would also add the one H[ow]). I personally think that a professional report isn't complete without them.

In fact, keeping in mind the five Ws should help us while building and designing our Pentest Report:

Who did that? Who is the report for?

With this question, we carefully define who will perform the tests and all the people involved. In the Pre-Engagement (http://www.pentest-standard.org/index.php/Pre-engagement) phase, for example, we will define a Project Manager as the preferred interface with the client and also a Pentest Team comprised of a Team Leader and components. This is an important task and should be included in the final report. In addition to that, we have to design our report for our target audience that in most cases will be on technical and on the management side.

Where did it take place?

Where were the tests actuated? This could include different information depending on the typology of tests performed. For a standard pentest, for example, we will include the asset, the systems in scope, but we can also add where we conducted the tests from (we are always answering the same question). Was it on site or from a VPN? This kind of info seems useless but not from a client's perspective: they are synonym of professionalism and scrupulosity.

When did it happen?

It's important to include a precise timetable in which is defined the start and the end of every security test performed. Furthermore, we can add info regarding the importance of the actions performed onto the targets in order to be able to activate the relative emergency plan (this allows to pre alert people in charge of specific systems/applications/department). The definition of what system is the most critical is determined with the client during the pre-engagement phase.

What happened?

As we previously said, technical perfection is a prerequisite, so we should report what tests were performed, what tools we have used and how we were able to exploit systems and gain access to privileged data. Moreover, we should include any methodology we have used in order to conduct the tests (like OSSTM or OWASP).

Why did it happen?

This is maybe the real question that the client is interested in: how and why did it happen? Why did the client hire a pentest firm to test their systems? Why, after the pentest process, were several high risk vulnerabilities found? And linked to that: how can I remediate the issues discovered? That's the ultimate

goal of a Pentest Report and in order to achieve that we must design a professional document, technically perfect and with effective communication.

The last question and its answer is intimately connected to the goal of a Penetration Test that is to help the client in the short/long term decisions in order to improve the overall security.

These kind of decisions could be for example could be attested on two different depth levels:

- Short term
- Mid to long term

The short term decisions are the ones regarding immediate changes in order to address the vulnerabilities that were found during the pentest; this could be source code modification in order to mitigate a Login Bypass or a patch in order to avoid a Remote Command Execution on a critical system.

The mid to long term area depends on decisions and changes in a company's organization and its internal processes. For example, if during a pentest, a RCE vulnerability of a year ago is found on a critical system, the boss could fire someone or develop new processes in order to make the patching process more agile.

These two depth levels reflect also the motivations and root causes behind a successful attack: very often the "how" an attack was performed reflects the "why" an attack was successful. In fact, if we as pentesters are able to influence the client in the long term this is a more persistent and permanent fix rather than an OS update or major version patch: the unsecured software could be patched but without complete and stable policies, a new vulnerability could be spotted (this, in fact, reduces over time the company's window of exposure).

A Pentest Report dissection

> *"If you do not document it, it did not happen"*

After a brief introduction to the needs of writing an effective Pentest Report we will dive into a "Pentest Report dissection" in order to design together an effective report.

One wrong assumption is that the report is the last step in the Pentest Report process. In fact, as outlined by Mansour A. Alharbi the Pentest Report involves at least four different development stages such as Report Planning, Information Collection, Writing the first draft and Final Review and Finalization of the report.

Figure 1 The continuous development of a Pentest Report

The methodology

In order to deliver a pentest report as professionally as possible, we have to carefully follow, throughout the whole process, a methodology like the Open Source Security Testing Methodology Manual (OSSTM), the Penetration Testing Execution Standard or PTES and the Open Web Application Security Project (OWASP) for WebApp pentests. The use of such a methodology is useful for both the pentester and client. In fact, it helps us during the pentest project with technical recommendations and project management guidance and at the same time, it shows the client our professionalism and that we have complete governance over the whole pentest process.

As we previously stated, a Pentest Report should be “a scientific process, and like all the scientific processes it should be repeatable by an independent party”. In addition to that, remember that the report is the ultimate and the only tangible output of the whole pentest process: if something is not reported, or badly documented, it never happened.

The target audience

A complete, comprehensive and effective final Pentest Report should be both technically perfect and able to make the company’s management aware of the discovered issues and be a trigger for them to start the necessary processes for the threat mitigations.

It’s not always feasible to exactly define who will read our report but it’s safe to assume that it will be viewed by at least three kind of audiences: the Management, generic IT technicians and the developers:

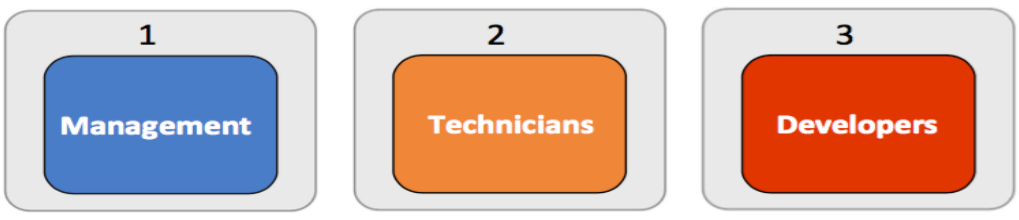


Figure 2 The three typical audiences

The company's IT Senior Management/Corporate level want to know if they are secure and if something bad is discovered, they want to discover who is responsible; it could be a person, a department or even a procedure/policy.

The technicians will be held responsible for fixing any vulnerabilities found during the security audit. To achieve that, they will search in the report for detailed information about what systems are vulnerable, how serious the threat is and a remediation plan in order to fix it. As a plus, we can suggest what vulnerabilities have to be fixed first. This is key information as it impacts both the business and technical field:

- The Management/Corporate area wants to know what needs to be fixed first in order to protect their business
- This information also has to be translated and communicated to the technicians who are responsible for the fix

The developers will fix the vulnerabilities in their code/applications; we will have to be as technically detailed as possible in the report. If the test includes a Code Review, we will have to make sure that the finding provides enough detail about the problem that anyone can understand the vulnerability and also attack scenarios. We would also add the risk related to the issue and suggest a remediation plan (as a plus, we can try to include the effort required for the fix).

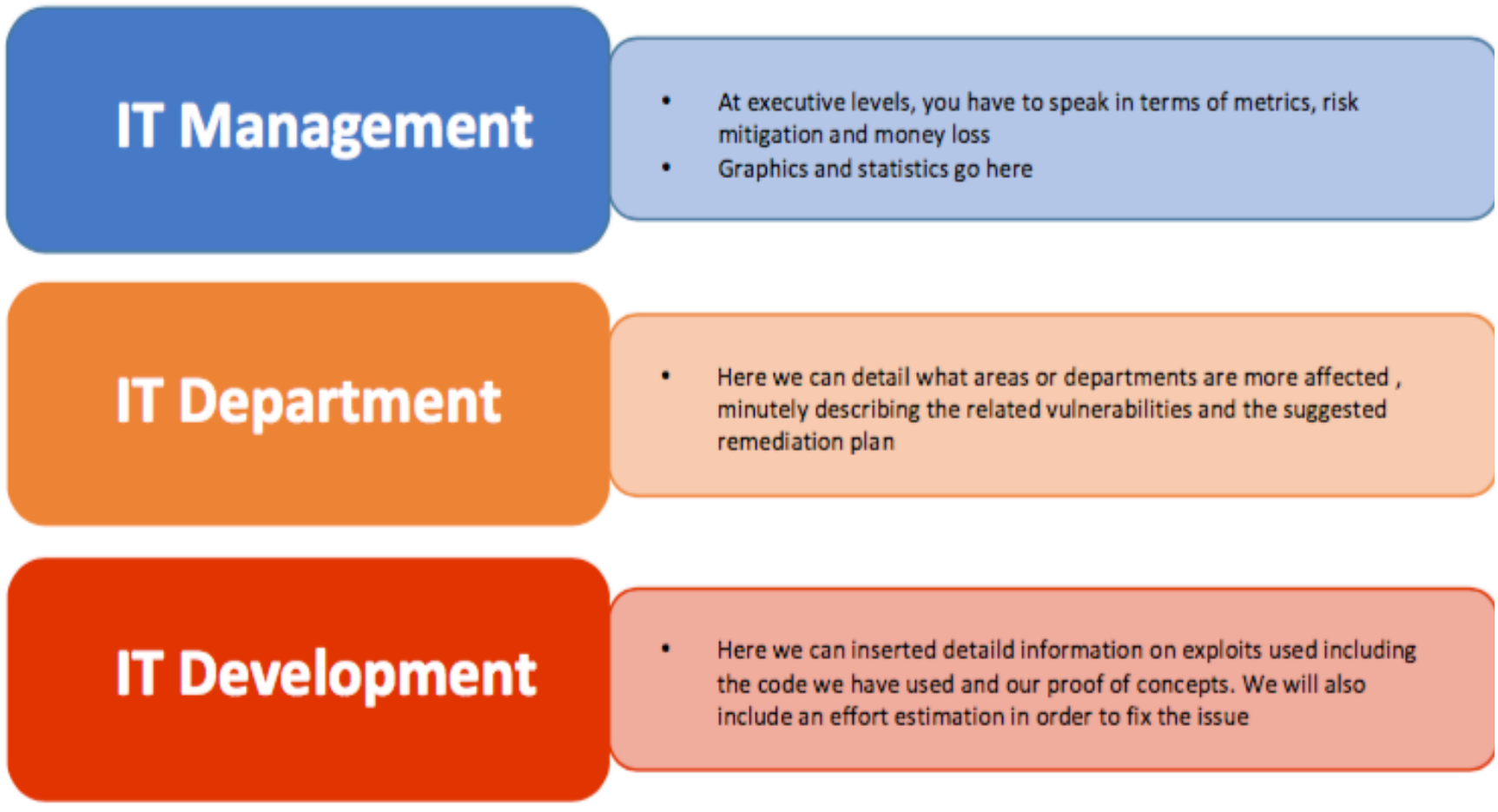


Figure 3 The three different audiences' needs

To summarize, we have to create at least as many sections as the different type of audiences that will read/consult the report.

In order to address different communication and language for different audiences we will have to consider:

- The position in the organization
- Knowledge of the IT security topic
- Authority or responsibility to make decisions

Cover Page

By Cover Page, I mean the initial page of the document with basic info like the document title, the client's logo, date and eventually a number of protocol:

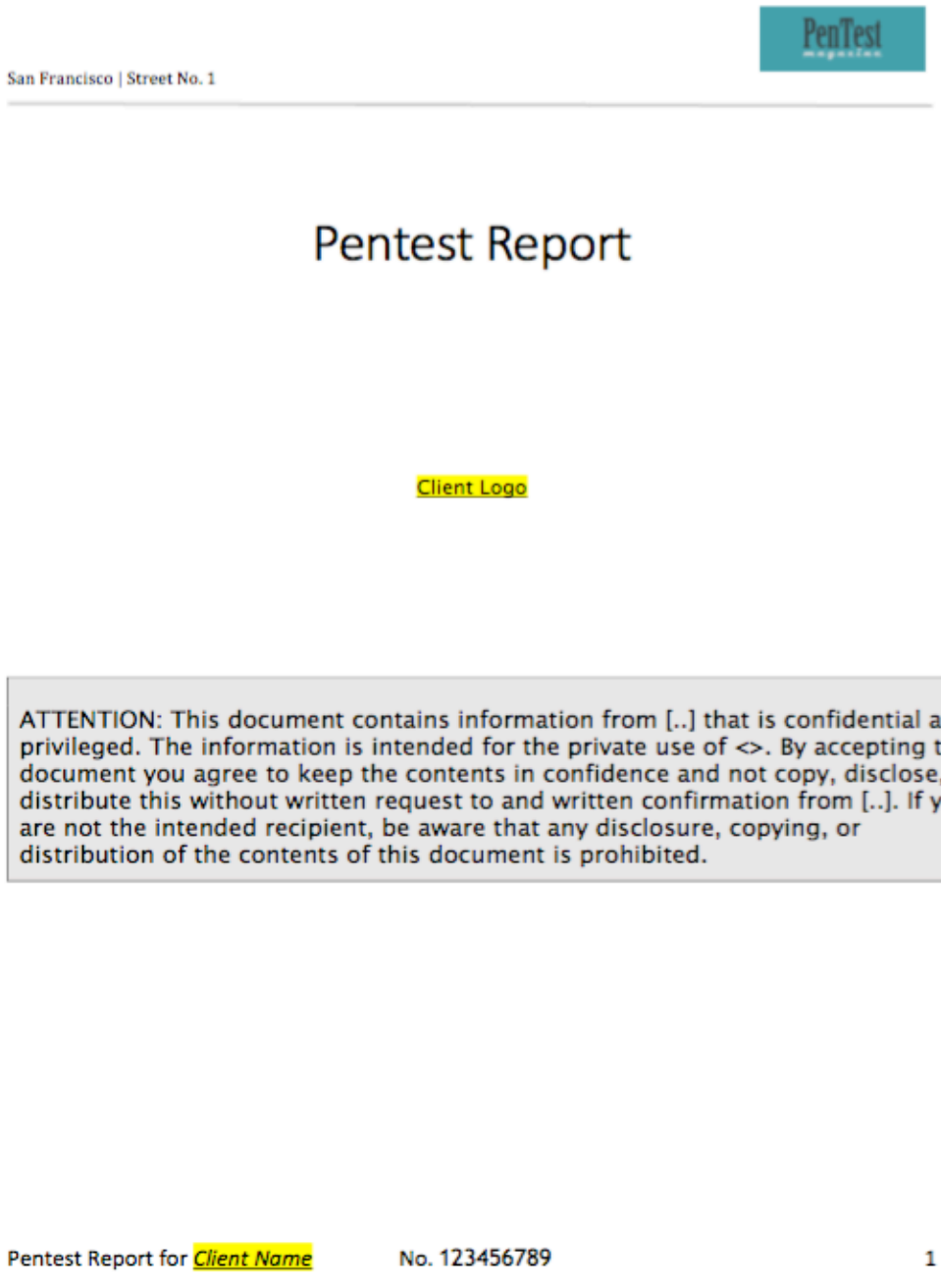


Figure 4 Pentest Report Cover Page

After the cover page, I would also include a table with information on document details, recipient, document history and the pentest team members:

Document Details	
Company	ACME Inc.
Document Title	Penetration Testing Report
Date	3/30/16 10:28:00 PM
Ref	No. 123456789
Document Type	Pentest Report

Figure 5 Pentest Document Details table

Recipient		
Name	Title	Company
PentestMag	Pentration Test Report	N/A

Figure 6 Recipient details

Document History			
Date	Version	Author	Comments
3/30/16 10:28:00 PM	v. 0.1	Lorenzo Vogelsang	Initial Version

Figure 7 Document History

Pentest Team Details	
Name	Role
Lorenzo Vogelsang	Team Leader

Figure 8 Pentest Team Details

Executive Summary

“Write this after you’ve completed writing the report. Think of what you’d say if you ran into an executive in the elevator and had one minute to summarize your findings.”

Introduction

The first thing to insert in the Executive Summary is a brief introduction briefly depicting the scope of the pentest. An example of a typical Introduction would be:

“Pentesting Company Inc. was contracted by ACME Inc. to conduct a penetration test of the internal network in order to determine its exposure to an insider targeted attack. All activities were conducted in a manner that simulated a malicious actor engaged in a target attack against ACME Inc.”

In the Executive Summary, we talk directly to the management and so it’s very important for us to able to concisely communicate an effective overview about what issues we have found and how secure their organization is. In this section, it’s not important nor useful to describe the vulnerabilities in the most technical terms but instead we have to be able to communicate technical findings to a non-technical audience. The challenge here is not merely technical but to use language that communicates to the non-technical manager, so in the Executive Summary, our job will change from a Pentester ninja to a great Communicator Evangelist. The main client questions we will focus on will be:

- What’s the impact and risk factor with the discovered issues?
- How bad is it for the client’s company? How could vulnerabilities impact the company?
- What does the client have to do in order to fix the vulnerabilities and protect their business?
- How much effort is required in order to do that?

If you’ll keep in mind all those questions, the Executive Summary will achieve its goal and that is to be able to speak with the relevant Executive/IT Management people.

In this section, we would add a lot of graphs, charts, and tables, and we will limit text descriptions just for essential things (the Executive Summary should be no longer than three pages).

In this section, since we are speaking with the managers, it’s always a good idea to also add a Time table with Start, End, Targets, type of security test and the associated risk, as you can see in the following picture:

Start	End	Targets	Step	Risk
03/04/2016 18:00 CET	03/04/2016 19:00 CET	10.0.0.1-20	Enumeration	Medium
04/04/2016 07:00 CET	04/04/2016 10:00 CET	10.0.0.1-20	Exploitation	High

Figure 9 Pentest time table example

Vulnerabilities graphs

Vulnerabilities by Impact

It’s great to include Vulnerabilities by Impact and the kind of successful attacks that were performed. For example, a “Vulnerabilities by Impact” graph could be similar to the following image:

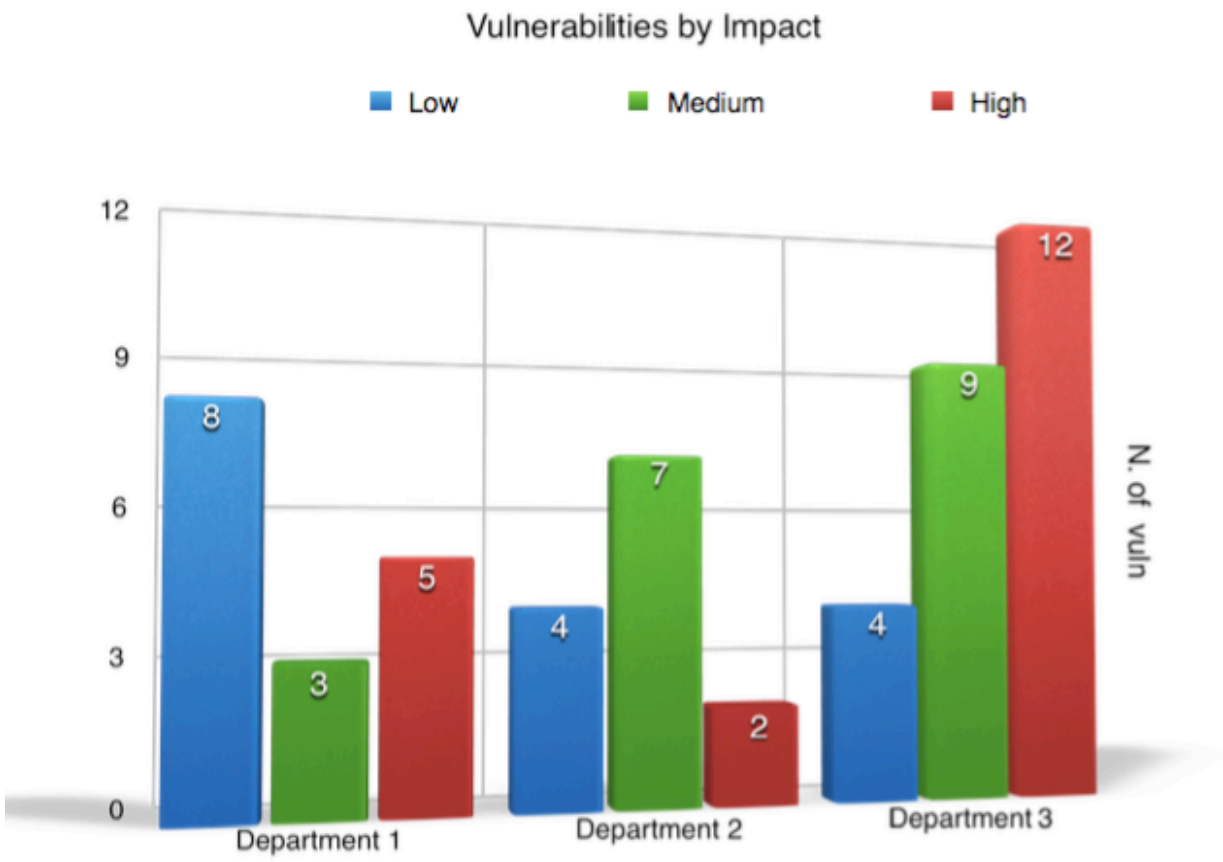


Figure 10 Vulnerabilities by Impact

This kind of chart enables managers and corporate personnel to quickly answer questions 1 and 2. This graph is neither theoretical nor a technical exercise: it's a business accelerator from a manager's perspective because it enhances the future client's reaction in order to fix the vulnerabilities. On a communication perspective, we are emphasizing the need of the report, we are saying to the client: "Look. Here is why you hired us"

Vulnerabilities by cause

The Vulnerabilities by cause is an important graph for the managers because it highlights the link between the cause of vulnerability and the departments who are responsible for it. A phishing scam, physical access, un-patched software or an SQLi in internal Web Applications involves different departments and at the same time impacts different business processes. With this kind of graph, we are giving the tools to the managers so they can analyze the company's overall security and, upon on our remediation plan, predispose the necessary steps in order to fix the vulnerabilities.

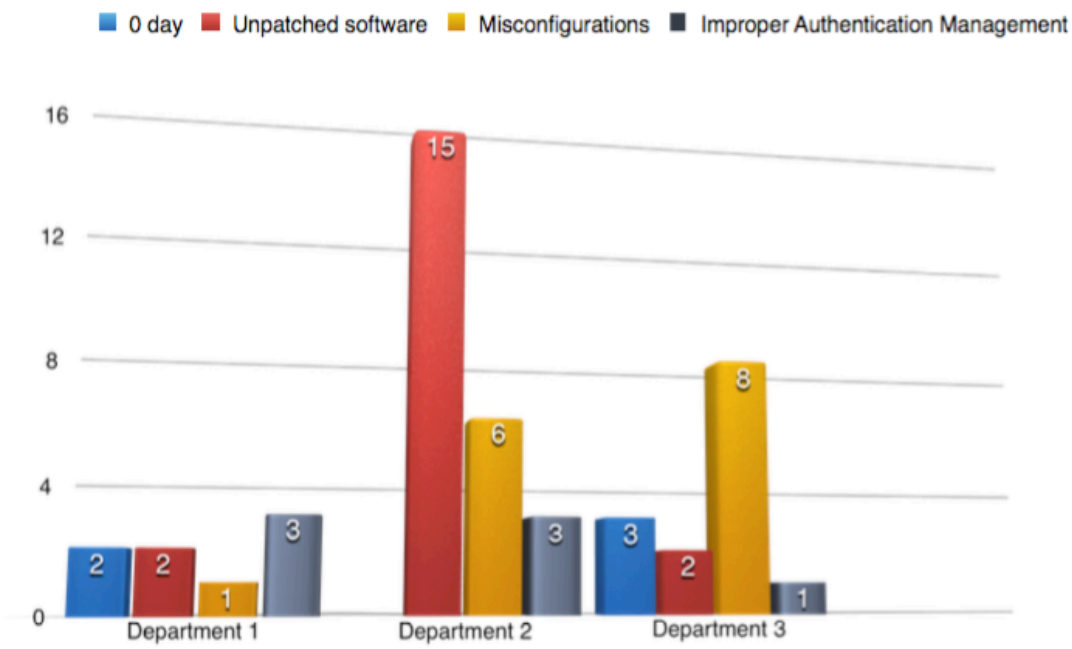


Figure 11 Vulnerabilities by cause

We can also add a more “technical” version of the graph detailing just the type of successful attack and not the involved departments:

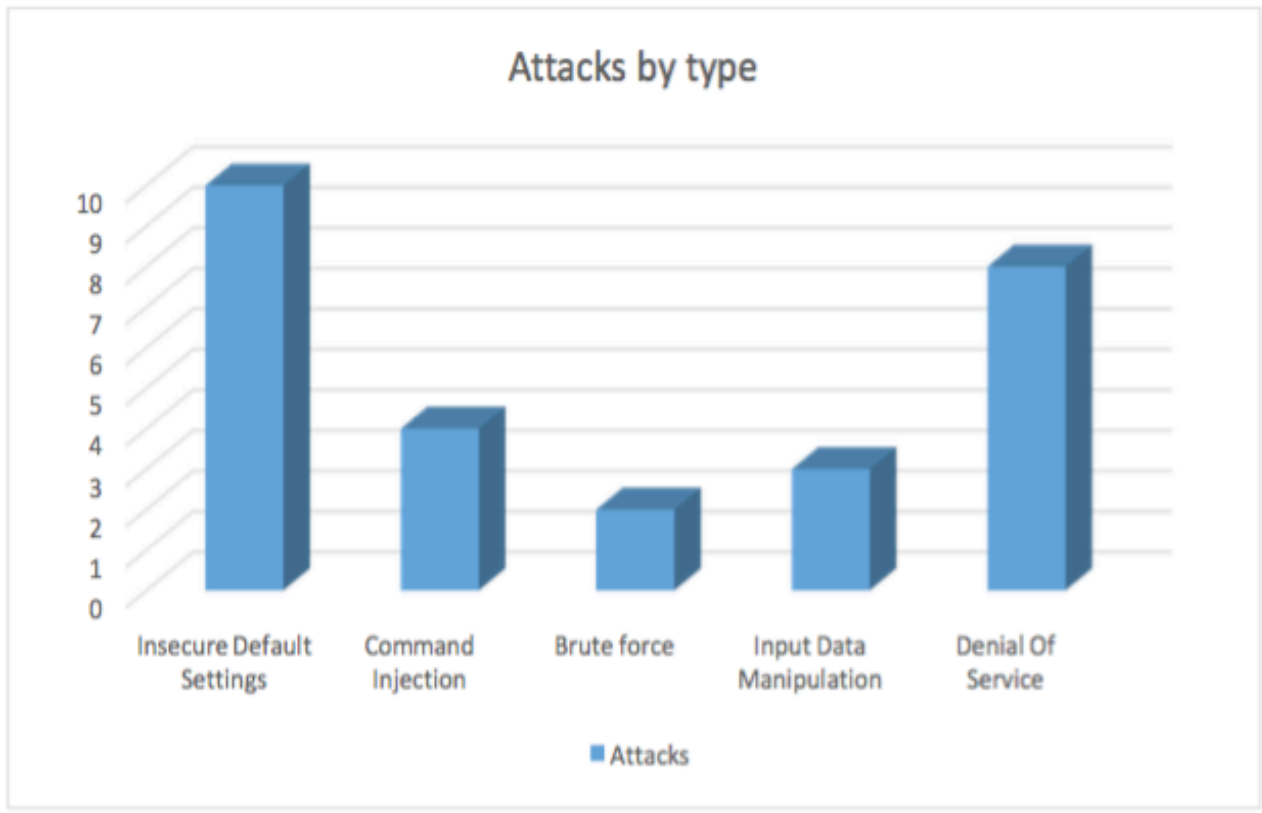


Figure 12 Attacks by type

A good, well known classification of attacks we could refer to is the MITRE Common Attack Pattern Enumeration and Classification (CAPEC) or the Web Application Security Consortium (WASC) Threat classification.

Recommendations

The recommendations section is a snippet, a preview, of what the Remediation Report will be.

We will include a simple but effective overall view of possible remediation for each target/vulnerability, adding the amount of work required (if we are able to estimate it):

- Patching is required on server x.x.x.x
- Implement stronger data sanitation
- Harden the server x.x.x. changing its default configuration
- And so on…

Vulnerability/Technical Report

As we previously stated, the reporting phase begins at the moment of the initial Rules of Engagement (we can add an Engagement Summary in the Executive Summary) but now it's time to present all the information collected in the most professional and coherent way possible. Here we will detail the targets in scope, representing all the info we have gleaned during the initial footprinting/enumeration Penetration Testing phase.

During Web Application Pentests, we often have a small set of vulnerabilities that affect a large number of different pages/URLs on different domains.

In this case, for example, we can opt for a "per vulnerability" approach and draft the following schema:

XSS		
Cross Site Scripting (XSS) is a type of security vulnerability typycally found in web applications. XSS enables attackers to inject client-side scripts into web pages viewed by other users. A cross-site scripting vulnerability may be used by attackers to bypass acccess controls such as the same origin policy. [..]		
Vulnerable URLs		
URL	**Parameter**	**Method**
/search.php	item	GET
/index.php	language	GET
…	…	…

Figure 13 Web App vulnerability details

Information gathering/service Enumeration

We can add in the Vulnerability Report introduction or, in a separate section, detailed information about the targets. We will include an aggregation of all the data collected during initial footprinting/scanning/enumeration phases:

IP Address	System Type	OS Info	Port	Protocol	Service
192.168.1.24	Web Server	RHEL 7	22	TCP	SSH
			80	TCP	Apache
			8080	TCP	Custom Web App

Figure 14 Target details

We can also add another column adding the service specific version.

By Vulnerability

In this section, we will include a Vulnerability Identikit for each of the discovered issues:

- A brief description of the vulnerability
- The Impact factored in CVSS v2
- The vulnerability type with reference of known classifications (MITRE CWE, WASC, OWASP)
- Screenshot and details about the exploitation including an “Attack Narrative” section

Affected targets with reference to details about the specific target in the document and internal number of vulnerability

Name of vulnerability	Short description of vulnerability
	CVSSv2 and business impact
	References to classifications like WASC, MITRE, CWE, OWASP
	Vulnerability ID (CVE, OSVDB, Bugtraq ID)
Exploitation POC	Screenshots of every explotation step
	Exploitation code
Affected targets	Vuln No # Buffer Overflow: IP, Domain / page reference/ parameters
	Vuln No # SQL Injection: Domain / page reference / URLs / Vuln parameters

Figure 15 A Vulnerability Identikit

For simplicity and coherence purposes, we can build an internal index of the vulnerabilities discovered and assign them a unique ID in order to build a valuable Identikit for each vulnerability discovered. This is useful when dealing with an unknown or undisclosed vulnerability with no publicly available CVE or OSVB ID.

The MITRE CVE-IDS are “unique common indentifiers for publicly known information security vulnerabilities” and this means that for known vulnerabilities, we should include at least the corresponding ID from CVE and/or OSVDB including a link to their page.

This will also improve searching for a specific vulnerability and trying to see which target it affected, for example. It's very important, as pentesters, to build our own formula to assign an impact for each of the vulnerabilities discovered. We can consider several factors:

- Difficulty of the exploitation
- The affected systems
- The exposure
- The availability

Of course we can add other questions and build our custom formula in order to reuse that and tune it with the client's needs.

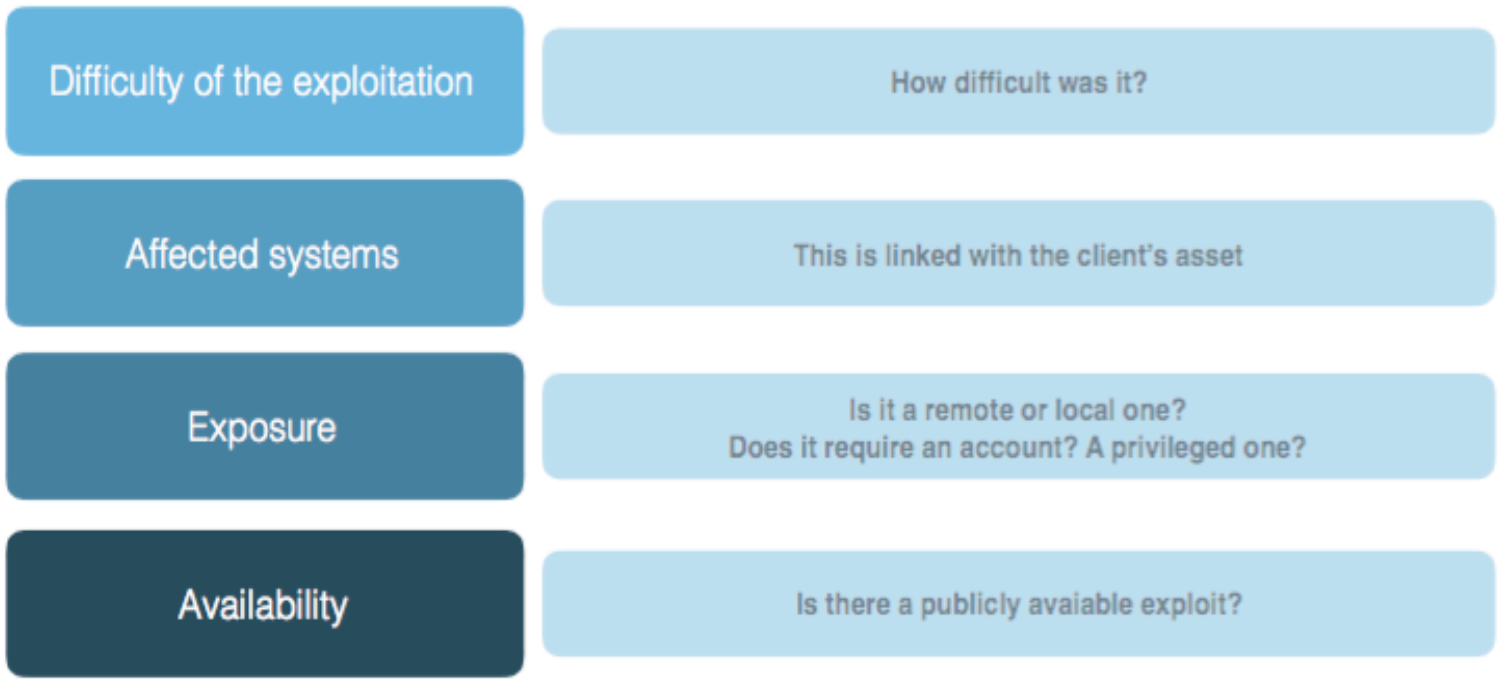


Figure 16 Vulnerability Impact factoring

If our scope is a Web Application, we can also add the following OWASP TOP 10 vulnerabilities discovered in target applications/URLs:

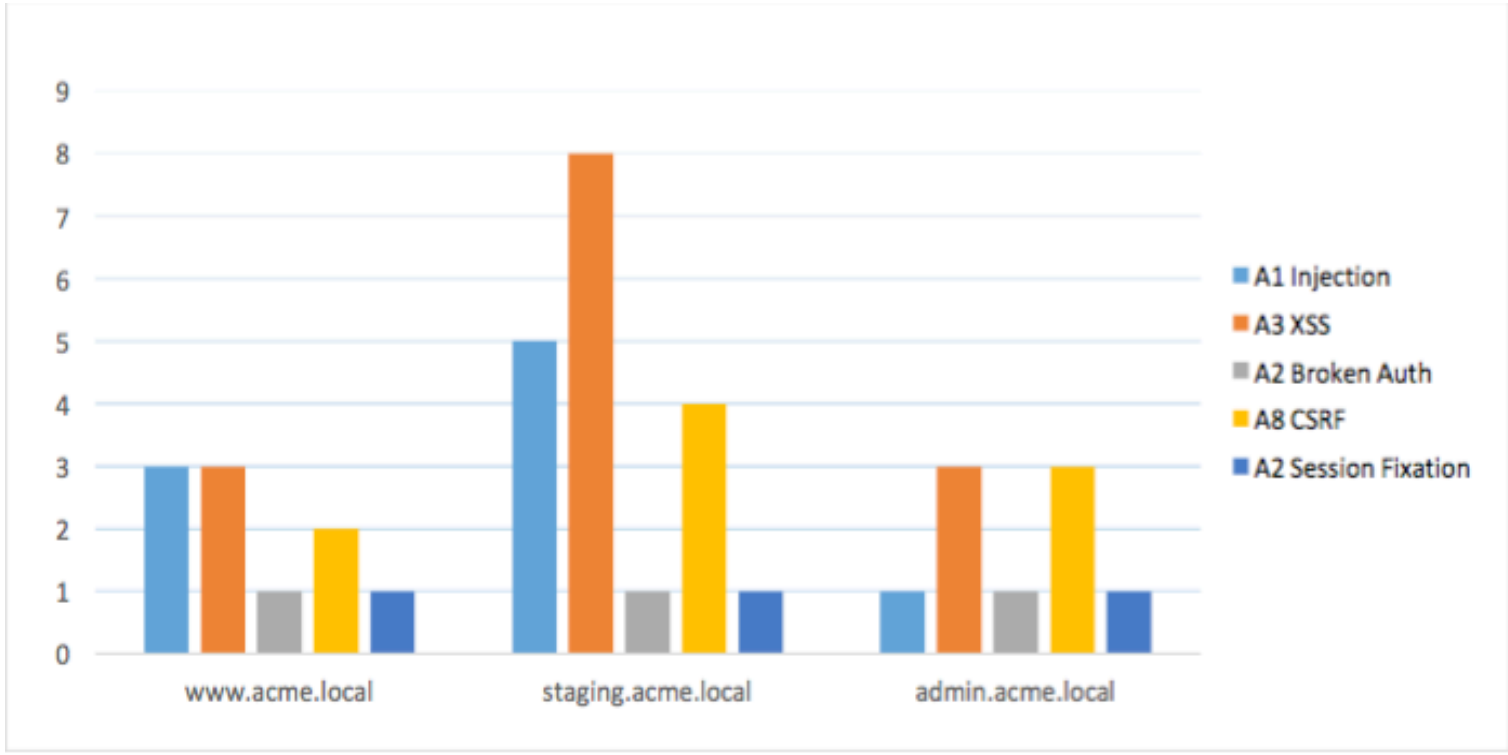


Figure 17 Vulnerabilities by OWASP Top 10

By target

In this section, we will document the very same information but now we will focus our attention on detailing the targets:

- General info about the target
- IDs of the vulnerabilities discovered and their severity

  -Brief description of the vulnerability

  -Vulnerability ID with OSVDB, Buqtraq or CVE

  -Severity of the vulnerability
- Graph with vulnerabilities discovered by type, cause, or impact
- Attack narrative section with a step by step description on the exploitation process

Remediation Report

We will design our Remediation Report depending on what approach we have previously followed (the per vulnerability or per target approach). For each item, we will include at least the following information:

- The vulnerability Identikit (including also the vulnerability vector and PoC)
- The mitigation fix on short/long term perspective
- The target that needs to be fixed

To be as effective as possible, we would also have to prioritize what vulnerabilities need to fixed first. This task could be accomplished considering the following factors:

- Severity and criticalness of the vulnerability
- How many targets are affected
- The business relevance of the affected targets

Appendixes

In order to add completeness to the report, we can add one or more Appendices in order to add details regarding the vulnerabilities encountered and the tools we have used. We can add, for example:

- A Reference section with all the references and links we mentioned in the Report. This is useful because the client can find in one place all the external information they want
- A Tools with a list of all the tools used during the tests. This is important as many clients want to know exactly what was used to attack their systems
- A Glossary section with all the relevant terms used in the Report

Conclusion

Writing an effective Penetration Testing Report isn't an easy task, it involves experience and a little bit of art to mix technical skills to language ones. This article could be useful for all the people who never wrote a report or even for the experienced pentester as a repository of useful information to use during the Penetration Test Reporting phase. If you want to know more about the topic in order to build your own personal formula for Report planning, I encourage you to visit these web sites with a very good list of Penetration Testing samples:

https://cure53.de/#publications

www.offensive-security.com/offsec-sample-report.pdf

http://www.sans.org/reading_room/whitepapers/bestprac/writing-penetration-testing-report_33343

http://www.vulnerabilityassessment.co.uk/report%20template.html

http://www.niiconsulting.com/services/security-assessment/NII_Sample_PT_Report.pdf

Author: Lorenzo Vogelsang

He has a diversified academic background in both the humanities and ICT Security; he has a degree in Pihlosophy and a Master's in Information Security. He is very passionate about Penetration Testing related topics and he's always looking for new and creative methods to circumvent system's security. He is a OSCP professional and a CyberArk specialist. He actually works in the Banking/Finance field.

References:

https://www.offensive-security.com/information-security-training/penetration-testing-training-kali-linux/

https://www.elearnsecurity.com/course/web_application_penetration_testing/

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

Smith, B., LeBlanc, D., & Lam, K.(2004). Assessing network security. USA: Microsoft Press

Mansour A. Alharbi,(2010). Writing a Penetration Testing Report.

https://www.giac.org/paper/gpen/2164/writing-penetration-testing-report/119556"https://www.giac.org/paper/gpen/2164/writing-penetration-testing-report/119556

https://www.sans.org/reading-room/whitepapers/bestprac/writing-penetration-testing-report-33343"https://www.sans.org/reading-room/whitepapers/bestprac/writing-penetration-testing-report-33343

http://www.isecom.org/research/"http://www.isecom.org/research/

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

https://www.owasp.org/index.php/OWASP_Testing_Guide_v4_Table_of_Contents"

http://resources.infosecinstitute.com/writing-penetration-testing-reports/"http://resources.infosecinstitute.com/writing-penetration-testing-reports/

Snedaker, S.(2006). It Security project management. Canada: Syngress. ISO (te Internation Organization for Standardization) and IEC (the International Electro-technical (2005). Information security management systems – Requirements BS ISO/IEC 27001:2005. International Organization for Standardization.

https://www.offensive-security.com/reports/penetration-testing-sample-report-2013.pdf"https://www.offensive-security.com/reports/penetration-testing-sample-report-2013.pdf

https://capec.mitre.org/"https://capec.mitre.org

http://projects.webappsec.org/f/WASC-TC-v2_0.pdf"http://projects.webappsec.org/f/WASC-TC-v2_0.pdf

https://nvd.nist.gov/CVSS/v2-calculator"https://nvd.nist.gov/CVSS/v2-calculator

https://en.wikipedia.org/wiki/Open_Source_Vulnerability_Database"https://en.wikipedia.org/wiki/Open_Source_Vulnerability_Database

https://cve.mitre.org/about/faqs.html"https://cve.mitre.org/about/faqs.html

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

https://www.offensive-security.com/information-security-training/penetration-testing-training-kali-linux/"https://www.offensive-security.com/information-security-training/penetration-testing-training-kali-linux/

https://www.elearnsecurity.com/course/web_application_penetration_testing/"

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

Interview with Mihai Raneti

Mihai has a degree in Psychology and various certifications in the field of IT and cybersecurity. He is passionate about quantum physics, math, history and technology. He has a critical thinking, always sees the bigger picture and is keen on problem solving. He is the brain behind a pioneering cybersecurity technology.

Can you tell us something about yourself?

Ever since I can remember, I have been passionate about sciences, particularly Math and Quantum Physics. I still keep the observation diaries I made in middle school as a reminder of who I was back then. As I grew up, my interests expanded and I immersed myself into the study of IT and all its connected subfields. At some point, I realized my day job was not enough anymore, so I quit and decided to start my own business, because I genuinely thought I could do more for people that way.

That was back in 2011, now fast forward to 2016, I run a successful cybersecurity startup that is preparing to officially launch its first product on the market. The road that led to this moment was difficult, it took much effort and many sacrifices, but we made it in spite of the people who did not believe in us, or worse, tried to discourage us.

Five years ago, I started a company that offered pentesting and ethical hacking services. It was among the first and few such companies in Romania. It was a completely new experience for me, and it ended up being the catalyst behind the cybersecurity concept we turned into palpable technology these days.

Can you tell us what does your company do?

Based on the conclusions from our pentesting activities, I came up with the idea of a proactive software solution that I developed over the course of two years. The package was complete, or so I thought back then. On one hand, we had the process that assessed the level of security. On the other hand, we also provided a solution for those who wanted it. Then we took things to the next level. During an accelerator, someone gave us the most valuable piece of advice we could have received: to create our own hardware. This is what we did, after spending months studying possibilities to combine the software with the right hardware elements, which was based on ARM technology. The advantage was that there were no third parties involved anymore, and the company could vouchsafe for its product.

We provide a pioneering cybersecurity technology which detects, and protects, against any unauthorized attack attempt. We collaborate with several heavy players in the field of hardware and we use customized components, providing additional protecting measures against outside attacks.

Presently, CyberFog offers protection through real time malware analysis, real time detection of advanced persistent threats, and previously unknown attacks, as well as deflection of DDoS attacks. Our machine learning driven algorithms ensure the entire system adapts against the attacks. Our servers do not interrupt or increase traffic within the network, and they are able to work even offline, since they represent a secure and independent communication protocol. In addition, our SPI has the ability to integrate IoT interconnected devices existing at the moment and other devices that will appear in the future.

We have not given up the pentesting side, it is still a vital part of the bigger company.

What do you think about cybersecurity nowadays?

One thing is clear, we can only speak about cybersecurity by adding cyberattacks and breaches into the equation. I noticed a few different trends after analyzing the clients' behavior and problems. Cybersecurity poses a massive danger to any company or organization, it is always on the rise, but it is not considered important enough by management boards. Unfortunately, this is a common case, because the damage a cyberattack causes may not be immediately apparent. It may take months or even years before a breach is discovered.

Secondly, traditional security solutions such as firewalls and antivirus fail most of the time, since they rely on information about previous attacks. Thirdly, even those companies that are aware of their vulnerabilities and allocate substantial financial resources to protecting their data are in danger, as attackers also invest money and effort into developing more sophisticated malware. In addition, one of the highest risks is represented by the human factor. Employees are the greatest source of vulnerability. Passwords like 123456 are still widely used in large enterprises, and let us not forget about BYOD policies or working from home on certain days. Last, but not least, is social engineering and how easy it is for someone to steal data this way.

Do you think writing an effective PT report is important and why?

The report is perhaps the most important part of pentesting, because it should deliver the correct message to different categories of people. It is the tangible document, or written proof, if you want, that states the degree of vulnerability of an organization. It goes without saying that details should remain confidential. If the message and language are clear, the benefits are tremendous. On the contrary, if it uses too much specific vocabulary and advanced technical terms – jargon - without explaining them, the audience might lose interest and not deem it properly important.

When writing a pentesting report, one must consider that it will reach not only the IT department or CISO, but also the CEO, and other members of the management board, who do not have that much knowledge in the field. Those people need to be convinced that they are facing real threats, because ultimately they are the ones who make decisions and who pay for the testing. IT staff are the most interested in the clear organization of the information (what system is affected, how seriously it is affected, and how they can fix the vulnerability), since they are directly responsible for fixing the issues

that were found during testing. While this is not common practice, some companies might present pentesting reports to their customers as proof of their level of security.

What should the perfect report contain?

There are two categories of clients: the ones that are very specific about what the report should contain, in terms of both content and layout, and the ones who have no idea about it and leave it up to you.

The major elements of a report should comprise the services that were provided, the methodology that was adopted, the results, and further recommendations on how to mitigate the risks.

Our reports generally contain a cover sheet with specific details, an executive summary, the scope and objectives of the work, the tools and methodologies we used, a summary of vulnerabilities followed by their detailing (how they were detected, how they could be exploited, how likely they are to be exploited, how they can be remediated), and the details of the team that worked on the test. The last part is very useful in case there are further questions about the testing or about the report. Testing is performed by a team, and all the members contribute to the report and to its final reviewing and editing.

What should we avoid?

One should avoid making assumptions. It is something I tell my team almost every day. In this particular case, we do not have the certainty that the multiple layer audience speaks the same technical language. If they do not understand our meaning, our relationship will be rocky and in the future they are not likely to maintain the collaboration. It will mean spending more time and effort clearing details that should have been understood from the very beginning. Ultimately, it means the pentesters are only doing half of their job right.

What do you think the common mistakes are?

Testers dedicate most of their time to the process itself, thus they have very little time left for writing the report, and the consequence is something rushed, that has not been reviewed or is not clear enough. Sometimes the results of the scan are copy/pasted from the scanning tool, without further details or explanations. To make things even worse, they do not offer suggestions on how to fix the problems they brought to light. Therefore, the value of the report decreases. If we bring into discussion smaller companies that do not have an IT department or a CISO, the impact is even more negative. Someone identifies their problem, makes them aware of it, and then leaves them to simply fend for themselves, which is completely unethical. Also, testers should keep in mind that they are dealing with a very sensitive subject, therefore they need to support each and every affirmation with concrete evidence.

At CyberFog do you use a specific methodology? Please tell us more about it.

We use the classical approach made up of three steps: pre-engagement, engagement and post-engagement. We gather as much intelligence about the company as possible, then we agree on a procedure that works best for our clients for testing itself. We decide on the tools and methods we will use based on the info we gather. It is more objective when they allow us to go completely undercover compared to when they inform the staff a test is going to take place. During a conference last year,

someone spoke about his experience with an American organization and how he had a schedule to get in, how he was allowed only in certain areas, and he was forbidden from talking to the staff and lying to them. At the end of the day, he still managed to obtain the data he set out for and proved they can be hacked. This should be eye-opening for all companies. Someone physically trespassing into your headquarters and stealing data is the ultimate proof of the risk you are exposed to.

In the post-engagement stage, we offer advice on how to remediate vulnerabilities and we offer the possibility of redoing the test after a period of time for those clients who require it. Also, we present them with our technology if they are interested in further protection.

Do you think that it is better to use one methodology or to switch from one to another?

I would say that there is no general rule, except to personalize each test so that it best responds to the needs of the organization. If the situation requires it, I see no wrong in combining two or more methodologies, as long as the results are accurate and objective, our goal has been reached.

If there is one thing that you would like our readers to remember, about writing an effective report, what would it be?

I would like to reiterate the importance of having enough time for writing the report. We have all been there, reports are boring and difficult to write, especially for multiple audiences. But in order for it to be effective, as you put it, you have to coin the best info, evidence, ideas, and vocabulary in such a way that everyone who reads it will understand it.

Have you got any final thoughts? Is there anything you would like to add?

As a matter of fact, I want to underline the increasing threats that are proportional with the growing number of interconnected devices people use. This is something I have noticed in all the companies we have tested. People focus on improving their lives with the help of IoT and apps, while paying little to no attention to how vulnerable this can actually render them. They do not consider themselves important enough to be a target of hacking, and that is the biggest mistake one can make. I see great opportunities in the future for cybersecurity developers, especially those who will also incorporate hardware, because IoT is about hardware.

Approach by Chrissa Constantine

A penetration test, or pentest, is a means to evaluate the security of a system by simulating an attack with the end goal of discovering issues before attackers and reporting them to the organization. Pentesting can be internal or external, and each has advantages and disadvantages.

External testing addresses the ability of an attacker to access an internal network. Internal testing addresses the ability of an insider to attack the network. The target is the same, but the attacker has some authorized access or is starting from within the organization's network.

Every penetration test should have a goal of identifying the particular vulnerabilities that impact the organization and of determining risk with an adverse organizational impact. Many companies undergo testing to meet requirements like Payment Card Industry Data Security Standard (PCI-DSS) or regulatory requirements, but the primary goal of testing should not be compliance; rather, it should be about security. The impact of not meeting a compliance requirement and the impact of data theft or loss of sensitive corporate data together can form the basis of testing.

The penetration tester carefully documents results and states the risks and findings in a final report. The final report contains an executive summary and technical details presented as the culmination of testing efforts. Pentesting can include automated and manual testing techniques and report issues identified via screenshots, tool outputs or other artifacts how the attack occurred or what the attack yielded. For example, if testing passwords, a pentester may show screenshots of cracked passwords and then show output from a tool that used the password to access a system.

How the findings are presented will be the determining factor on successful reports. Many pentest reports fail due to poor grammar or spelling, illogical flow or unclear remediation instructions. Clients want to know how vulnerable the organization is and what requirements exist to reduce overall security risk. The formatting and tone of a report can provide positive or negative reactions. Critical security flaws can be ignored due to language that does not convey how essential it is to have corrective action. Also, understand what the customer requirement is for meeting industry regulations or standards.

Reporting must be planned before testing. If during testing, a pentester discovers a critical exploitable vulnerability, the tester may alert the company before completing the test. A summary of exploitable issues can be sent, so the client can immediately correct the issue. These initial findings can be either documented in a summary report containing corrective actions or can be sent via email, depending upon deliverables defined within the statement of work (SOW) or agreement. Whatever method of delivering these urgent or critical issues, ensure they are sent with confidentiality in mind.

Since these are exploitable issues, ensure appropriate measures are taken in sending information securely to the client.

To effectively plan writing a report, ensure that during all aspects of testing, the capture of tool outputs and notes occur simultaneously. Below are some points to keep in mind for effective reporting:

1. Identify the scope and purpose of the pen test.

    - How far the pentester goes in any area of the test will be defined in the statement of work. Subjects are discovery, mapping, vulnerability assessment, application testing, network assessment, exploiting and then reporting based upon the focus of the engagement.

2. Determine if there is a customer requirement to meet industry standards and ensure the report is tailored to meet that requirement.

    - For example, a client may need to meet Health Insurance Portability and Accountability Act (HIPAA), Federal Information Security Management Act (FISMA), Payment Card Industry Data Security Standard (PCI DSS), or Sarbanes-Oxley Act (SOX)

3. Identify the target audience. Design the report to consider who will be reading it. The audience may be technical or may include managerial audiences.

4. Identify client contact information.

    - Pick a point of contact (POC) within the organization and document various methods for contact – mobile number, office number, email, etc.

5. Identify and describe the methodology used to conduct the penetration test.

    - There are several to choose from, such as Penetration Testing Execution Standard (PTES), Open Web Application Security Project (OWASP), Open Source Security Testing Methodology Manual (OSSTMM), or STRIDE/DREAD.

6. Identify a risk rating scale, provide a list of tools and a brief description.

    - Risk ranking helps clients understand how to react to a finding. Commonly, summary charts and details for all individually identified items are in a report. Ensure references are included to back up findings.

7. Provide detailed testing procedures. During the test, take copious notes, gather information, identify weaknesses, misconfigurations, and vulnerabilities.

    - Capture tool or script outputs, screenshots, or reports from automated tools.

8. Organize information and list all sources.

9. Evaluate the information. Double check the work.

    - Explicit descriptions of the source, the exploit and the fix actions are needed.

10. Explain how to remediate or take corrective steps. Emphasize and rank critical information by most critical to least critical, and provide relevant extracted data to help the client understand how the vulnerability was exploited.

Every penetration test should have a goal of identifying the particular vulnerabilities that impact the organization and of determining risk with an adverse organizational impact.

11. Include relevant details related to the final findings in an appendix. The appendix is for reference purposes, and can include screenshots or other results.

Reports contain sensitive information. Ensure the report and all data affiliated with the report is protected according to the report sensitivity. The classification of the report will be based on the target organization's information classification policy.

- When sending the report, use secure methods of delivery. A secure portal, encryption or other methods can be used to deliver the report to the client.

Testing has a clearly defined scope and rules of engagement and includes a final report documenting all activities performed during the test including technical details about vulnerabilities and the means in which they were exploited. The results have a metric for risk and give the company ways to identify the likelihood of an occurrence of the attack. The final report must document all activities performed during the test with technical details about vulnerabilities or issues, the means in which they were exploited, and remediation steps.

Present test findings clearly and use language geared for the intended audience. For each finding describe the threat level, rating, analysis of the issue, the IP address, risk rating and remediation or corrective action. For vulnerabilities, describe the source, impact and likelihood it will be exploited.

When providing vulnerability or other risk or threat information give credit or provide details about the work from other researchers or authors. This list of references can be in line with the findings or aggregated at the end of the report.

Some reports provide an appendix that contains output from scanning tools, vulnerability test results, or other information that is related to testing, but not essential to the main finding. Testing results will also validate existing controls and risk mitigation techniques deployed in the environment.

Pentest reports usually have repeatable attack methods and use a methodology that identifies all phases of testing. If the client needs to reproduce the results to validate findings, the report should be written in such a manner that the issue can be recreated by a different tester. Reporting needs specific details about how systems and applications were compromised during the test. Provide a list of tools used during testing with brief descriptions.

Once the report is ready for delivery, ensure that it is transmitted securely to the client. Since the report contains details about vulnerabilities and exploits for the customer's network, it is important to make sure the report is not sent in an insecure manner. If sharing samples of pen test reports with other companies, ensure reports do not contain any details of a real organization. The responsibility to protect the report does not stop at the end of the engagement, but due diligence must be performed to ensure the confidentiality of all test results.

Author: Chrissa Constantine

Chrissa is an Information Security Analyst and has a Master of Science in Information Security, CISSP and CE|H certifications. She held positions as a consultant at Apple and for a Silicon Valley start-up as a penetration tester. Chrissa enjoys hacking competitions, meeting new people, and learning new things.

Approach by Mattia Reggiani

Penetration testing is a method of evaluating the security posture of a computer system, network or application by simulating an attack from malicious users. The process involves an active analysis of the system for any potential vulnerabilities and their exploitation. This could result from improper system configuration, known and unknown hardware or software flaws, or operation weakness in process or countermeasures.

The phases of a Penetration Test depend on the engagement, scope and limitation, but generally consist of information gathering, vulnerability analysis, exploitation, post-exploitation and reporting. One of the most important activities of this process is the stage of reporting, because if it is not properly created or if it is imperfect, it could devalue the entire engagement of Penetration Testing, even if it was done in the best way. Indeed, the technical execution of the assessment can be considered the first part of the overall engagement: the final product is a well-written report with all of the supporting evidences, because the report is the only thing that is going to represent your work.

Reporting of scientific process

The report should be oriented to the scientific method and to the reproducibility of the results, because the Penetration Testing is a scientific process. It must been done in this way, to explain and to document every action undertaken and any tool used in order to (re)perform technical analysis at a later time, in the Galilean sense of "repeatable". Furthermore, if a client does not agree with the results of a test, he has every right to seek a second opinion from another tester using the same report and the same procedures.

Audience

A report should easily be understandable both to senior management and the technical staff, and should highlight all the risks identified during the assessment phase. In fact, the final report of the activity should be addressed to management, which is interested in overall security of the organization because it is responsible for infrastructure management, and IT technical staff for detail about the vulnerabilities detected and how to fix them.

Structure

Generally, the report consists of two streams, to various audiences, in order to communicate the objectives, methods, and results of assessment: executive report and technical report.

Executive report

This section, addressed to the senior management, will communicate to the reader the specific goals of the Penetration Test and high-level findings with related effective risk and generic recommendations. The intended audience will be those who are in charge of the oversight and strategic vision of the security program. This section should show the overall security posture with attack surface exploitability.
The recommendation section of the report should provide to reader the high-level understanding of the tasks needed to resolve the risks identified and the general level of effort required to implement the resolution path suggested. In addition, it could insert a possible strategic and tactical road map to follow, which would include a prioritization plan for remediation that takes business objectives into account.
The most interesting part of the deliverable are the findings and remediation, but a penetration testing report should also document the approach used for the assessment, showing and explaining the various steps, and methodology to calculate the risk of vulnerabilities and misconfiguration with related real exploitability.
Generally, the risk level is calculated through the correlation of threats and vulnerabilities. Once a vulnerability is detected, the related risk is evaluated according to the possible impact it could have on the IT systems and according to the technical skills that are required to exploit the vulnerability.

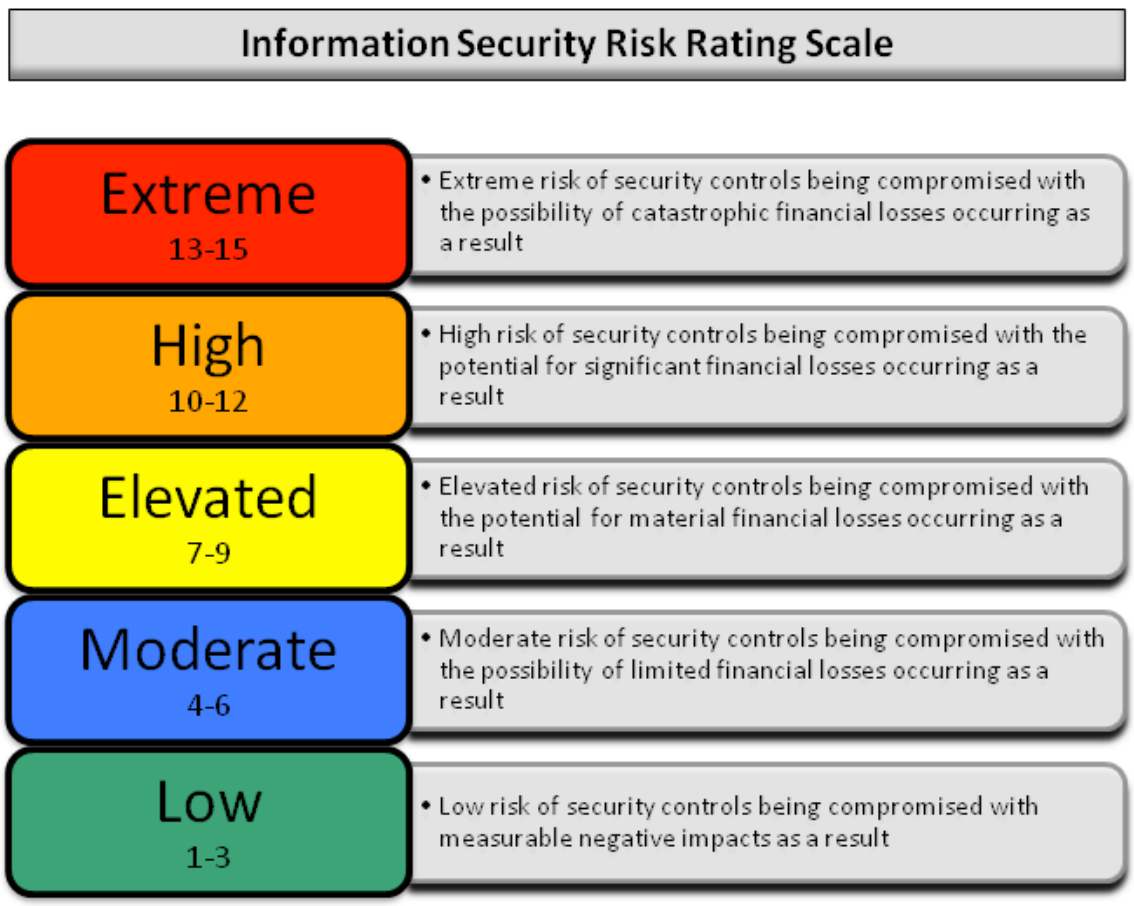


Figure1: Risk rating, based on PTES Technical Guidelines

Finally, even the limitations or out of scope areas should be reported (e.g. social engineering), in order to give the right value to the engagement and the attack scenarios assumed.

Technical report

The detailed technical report should consist of the comprehensive list of vulnerabilities found, with the evidence to support them, the description of any successful attack and possible containment and resolution actions, in the short and long run, that can contribute to the reduction of risks and enhancing system security.

For each vulnerability or misconfiguration detected, you should describe:

- Description: The penetration tester should describe a root cause of the vulnerability by exactly highlighting the provided environment. The description should also include the family and categories of vulnerability with technical references to components or vulnerable code.
- Proof of Concept: Following the description, a proof of concept (PoC) of the vulnerability exploited should be presented. Generally, include a screenshot of a successful exploitation or detailed logs of console input and output.
- Exploitability: A great value component is exploitability, because it represents the plausible risk of the vulnerability, which depends on complexity, code availability and successful exploitation.
- Impact: This field should explain the impact of the vulnerability. It will depend on how severe the outcome may be that it is going to create. Therefore, it is very important to analyze and represent the impact of the attack based on tested environment.
- Likelihood: This area is for explaining the likelihood of the vulnerability, which depends on ease, availability and possibilities of attack. In addition, whether or not the interaction with human or authentication is required for successful exploitation.
- Risk evaluation: The final level of risk should be determined as to vulnerability, according to methodology of risk analysis previously shown. In many engagements, the pentester won't have enough information to perform a proper risk analysis, especially regarding the risk related to business view. In this case, the collaboration with the client is very important to assign a correct and business-oriented risk value. Otherwise, the pentester could give an initial risk value, independent of context, from his perception.
- Recommendation: Should be provided as a series of technical recommendations in order to fix the vulnerability, showing the patch for code, hardening tips or possible workaround.

Report writing process

Report writing follows a specific life cycle process, similar to the below flow:

Planning

Planning is the first step of the reporting process; it defines the basis on which to write the report and communication of the results to the client. The ponts to focus on in this phase are:

- Objectives: to provide an overview of the scope of the assessment, project goals and how the achievements should be documented.

- Time: the results obtained should be considered time-limited, because new vulnerabilities are discovered every day. In fact, deciding with a time lapse at the beginning of the test is a very important point because the whole test and report will depend on it.

- Audience: in this step, the structure of the report has to be decided in relation to the addressee so that it can satisfy all the stake holders, from senior management to the IT manager and technical staff.

- Classification and distribution: the report document contains confidential information that may be protected by law and it is intended solely for use of the recipient. Indeed, the access to the document by other parties should be strictly prohibited. Any comment or advice contained in the report should be subject to the terms and conditions expressed in the contract between pentester and client. Finally, even distribution management plays an important role in making sure that the document should be handed over to an authorized person within a proper time line.

Information collection and organization

In this phase, the information and evidence of the penetration test are collected and organized, in order to provide proof of every action taken during a penetration testing activity because it's required to document every step executed by a penTester. The most important evidence should be:

- Screenshots: the use of screenshots as evidence is widespread because it can provide a good representation of the results obtained, even to non-technical staff.

- Logging: logging all activities performed is very important, also as a result of potential damage to identify a source and exonerate the tester.

- Scripts and exploits: scripts and exploits written ad hoc should be attached to the report to allow the repeatability of results.

Draft Report

Following the initial stages, the pentester can start to write a first draft of the document, paying attention to the following three points:

- Structure: A good structure helps the reader in reading a report in a much smoother. manner The first part should be generic and high level, and then go into detail in the final part. It should be composed of the executive summary, technical report and list of findings.

- Content: Content is the essential part of the report, which should be written with a great deal of information, supported by the evidence, to make the reader understand the steps performed and the related results obtained.

- Formatting: Formatting refers to the text style and size of text to use in the report. The headings, content, source code and various other texts in a report should have completely different style of formatting based on conditions to create a clean and clear report.

Quality Assurance

After the drafting of the report, a review of the quality and correction of any errors or imperfections should be performed. This phase should be done with the assistance of an expert in languages, to correct grammatical errors and shape, and a technical expert to validate the concepts listed in the report.

Review and finalization

At the completion of the phases described, the deliverable should be reviewed by more people, at various levels within the team, as well as to provide further improvements and suggestions. In addition, if possible, it is useful to have the report reviewed by people outside of the engagement, in order to simulate the reading of the final client and discover specific errors that may be identified by a user who is not involved in the project.

Conclusion

This article has described a possible way to write an effective Penetration Testing report, with some guidance, advice and a possible writing process.

The final deliverable is a component of great value within a Penetration Testing because it highlights all the activity performed and is the only thing that can represent your professional work. In addition, it is the only product that the client will receive.

Therefore, it is very important. Spend a good portion of time on this task, compared to the entire project, and heavily invest on training and improving communication and expression skills.

Author: Mattia Reggiani

Mattia is an offensive security enthusiast, certified Ethical Hacker and graduated summa cum laude from a Master's degree in Information Security at the University of Milan.

He is interested in ethical hacking, forensics analysis and cyber intelligence.

Currently, he works as IT Security consultant for a wide range of clients: banking, insurance, industrial, automotive and energy.

Approach by Bruce Williams

Red Queen Pen Test Reporting– A View from the Classroom

When Alice meets the Red Queen in Lewis Carroll's "Through the Looking-Glass," she runs as fast as she can, but then realizes they remain under the same tree as they started. The Red Queen explains: "Here, you see, it takes all the running you can do, to keep in the same place."

http://www.brandchannel.com/2015/05/01/logistics-050115/

Executive summary

Writing a great pen testing report requires both an understanding of the range of pen testing tools and client expectations. The agreement between the client business and the pen tester deals with expectations of both parties. This article covers the flow of this process. I train students to write pen test reports. I use the saying a man with two watches will never know the right time. With so many pen test tools, you are like that man with two watches, never too sure of the right number of vulnerabilities present in your website. This article is designed to help you know which will reveal the correct answer. The correct answer is a happy client.

Introduction

Pen testers do a great job of seeing risks in websites and servers. How do you train a good pen tester? Can you train someone or is it something that already exists within your nature? Well, I have to train students not only in the art of pen testing but also in dealing with client expectations and, more importantly, writing an informed pen test report. The balance of finding exploitable holes in a company's defense versus the cost to perform these tests is a balance well known to pen testers. The client would say "I want 100% bulletproof web server" but then say "I do not want to pay a lot". The pen tester would reply "If your business is only online I would expect you to spend a reasonable level to protect your main business asset". A pen test is a delicate art form of explaining technical risk identification and keeping your client happy. A Pen Test Report is the embodiment of this explanation. It is part of a risk assessment process.

The above graphic showing the Red Queen shows how pen testers need to run at least twice as fast to avoid the hackers. The Red Queen is the pen tester while Alice is the business client who is going along in this strange new world. I call the Pen Test Report the Red Queen Pen Test Report. My alternative was Deathworld Pen Test, based on Harry Harrison's SF novels. The link is the co-evolution of predator and prey (his world develops attackers at the same rate as defenders), but it is not a great marketing name.

I teach IT networking students about pen testing and report writing for a business audience. This fits into a diploma with units that cover risks to the continuity of any business. Disaster recovery and client understanding are parts of this course. Pen testing has an important place in this training as it allows an organization to reduce risks on its servers and protect their business. The students need to write a good penetration report. I expect great things to happen every so often but I am happy with good.

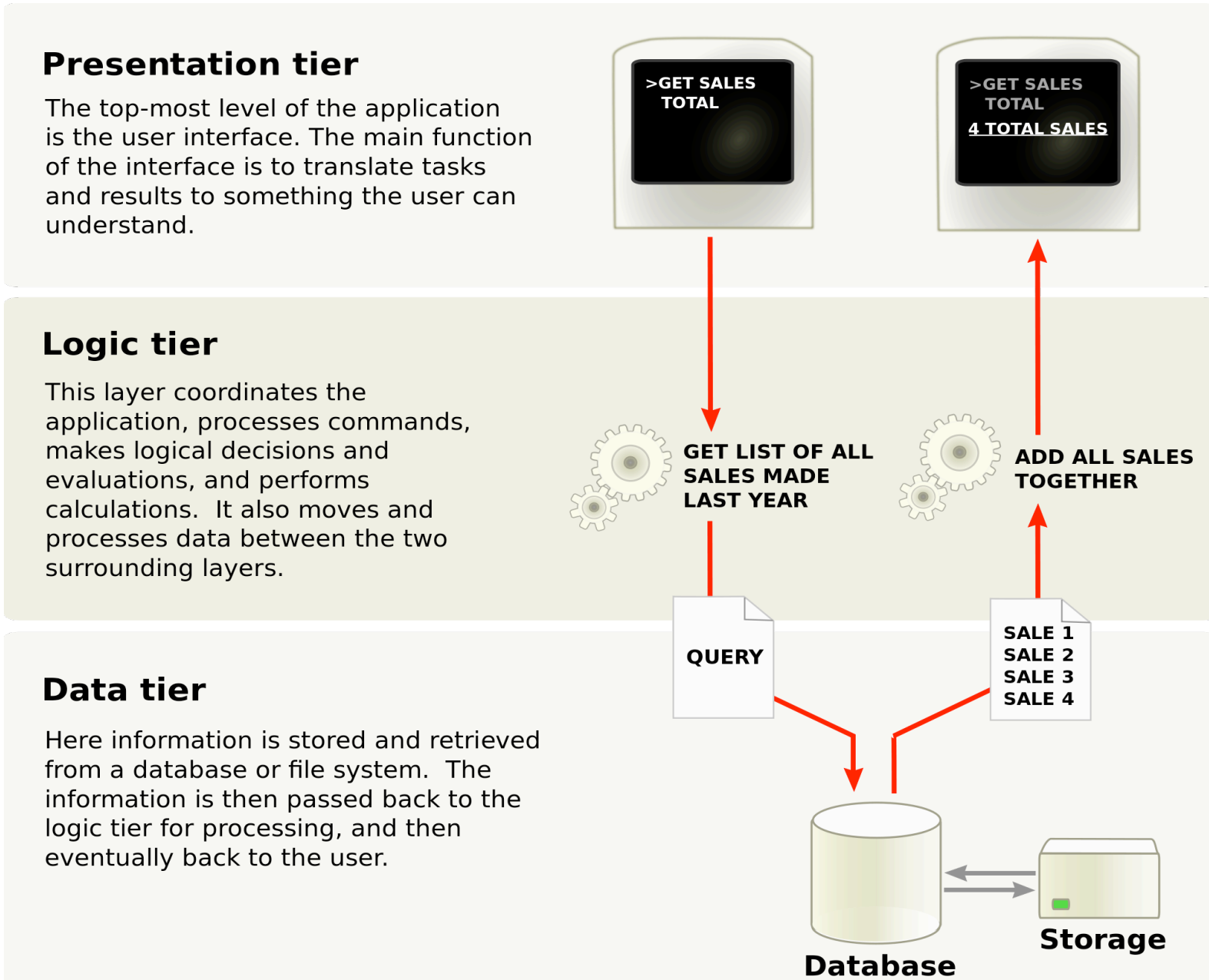
So what makes a good penetration report?

I look at the pen testing from a system's viewpoint (a viewpoint which is common in other units).

Pen testing was defined to be "A test methodology in which assessors, typically working under specific constraints, attempt to circumvent or defeat the security features of an information system"

The system for a web server consists of three layers. This is a technology stack approach. The top layer is the web pages/CMS sitting in the middle layer (mysql/php) which then sits on Apache or similar HTTP server platform. The tools that are directed against this system include Kali and Samurai, which are Web Test Frameworks (WTF) that show weaknesses and therefore represent risks to business continuity. Each layer has its own weaknesses and joins between these layers are a separate problem. A pen test is a snapshot of these layers at a certain point in time. Each layer will change.

I use this three layer model to explain web servers but the three tier architecture shown below is the best one to explain "injection" with the most common a SQL injection. In a simple case, the Presentation layer bypasses the Logic layer to manipulate data in the Data tier. Replace data with credit card details and businesses respond with 'Protect me!'



The training in this pen testing process has a certain flow. It follows a simple project structure of initiation, planning, execution and closure.

Initiation: The Contract between Business owner and PenTester.

The first is the business legal side of authorizing access. Unauthorized access is illegal. A contract proves that access has been authorized. The next step is expectations of the test. These are found in the deliverables at the end of the contract along with a payment schedule. I use the OWASP framework to cover this part. This contract is for software development and needs modifications for a Pen Test. A contract is needed (unless it is an internal test group) that specifies the obligations of each party. The website and range of IP addresses with a check on ownership is the first part of the contract. The section that is of interest is the use of standards or methodology:

(e) Security Analysis and Testing

Developer will perform application security analysis and testing (also called "verification") according to the verification requirements of an agreed-upon standard (such as OWASP Testing Guide).

The Developer shall document verification findings according to the reporting requirements of the standard. The Developer shall provide the verification findings to Client.

> *The software shall not be considered accepted until the certification package is complete and all security issues have been resolved.*

The standard is the main problem. I will discuss standards a little later. If you follow standards, it gives both sides protection.

The other part of the contract that I ask about is the section on acceptance.

11. SECURITY ACCEPTANCE AND MAINTENANCE

Acceptance

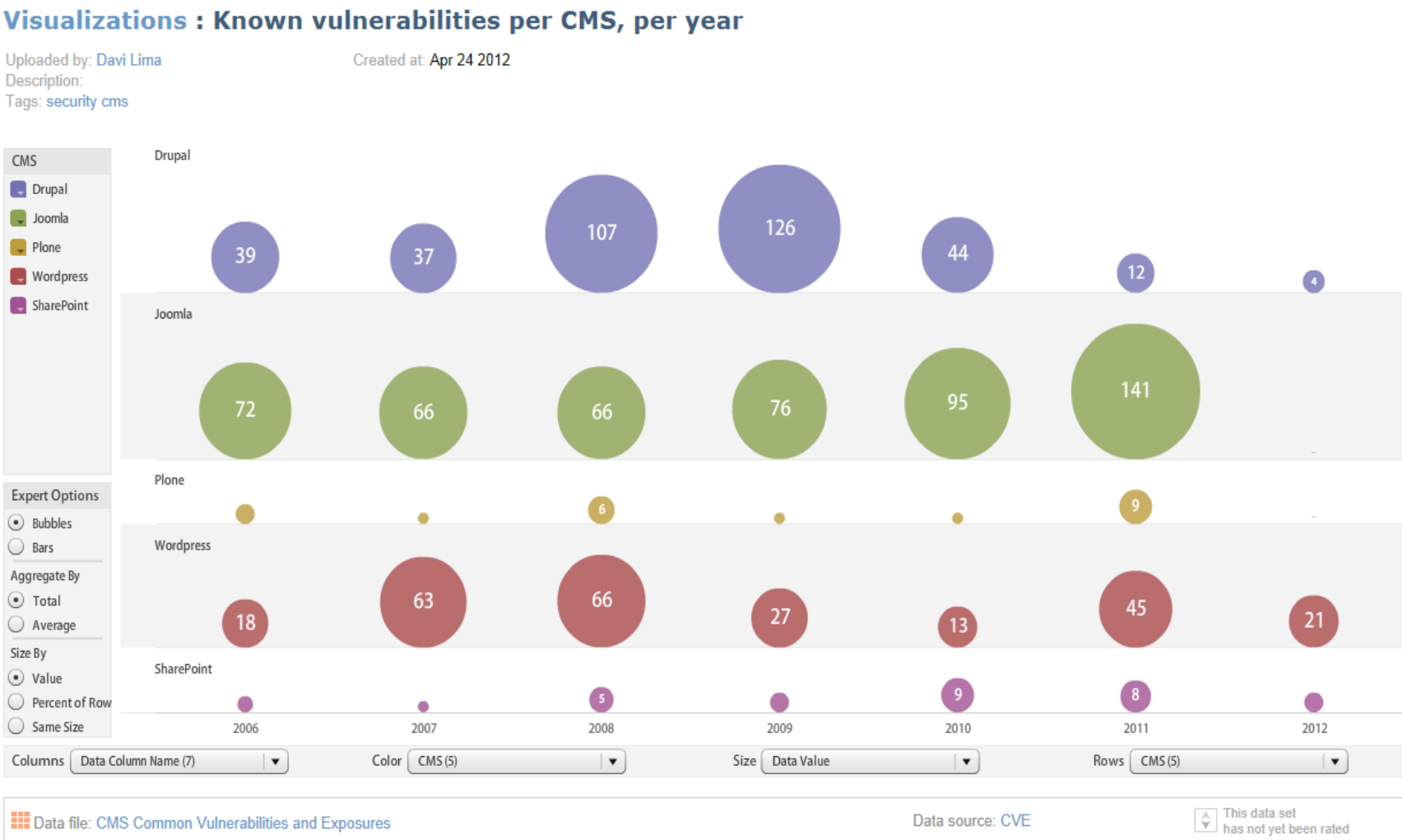
I ask the students if they will give me 100% guarantee that the risks are identified. So what am I testing and what report is needed for the client in order that they be paid?

The contract is then debated and expectations are better understood. I use the OWASP Top Ten. This is a dynamic list of the top ten risks to business. The usual suspects of cross site scripting floats XSS from number 3 to number 2 depending on the attacks that year. It is an industry driven list, which means it is not academic. So this OWASP Top Ten is the basis of contract for the pen test.

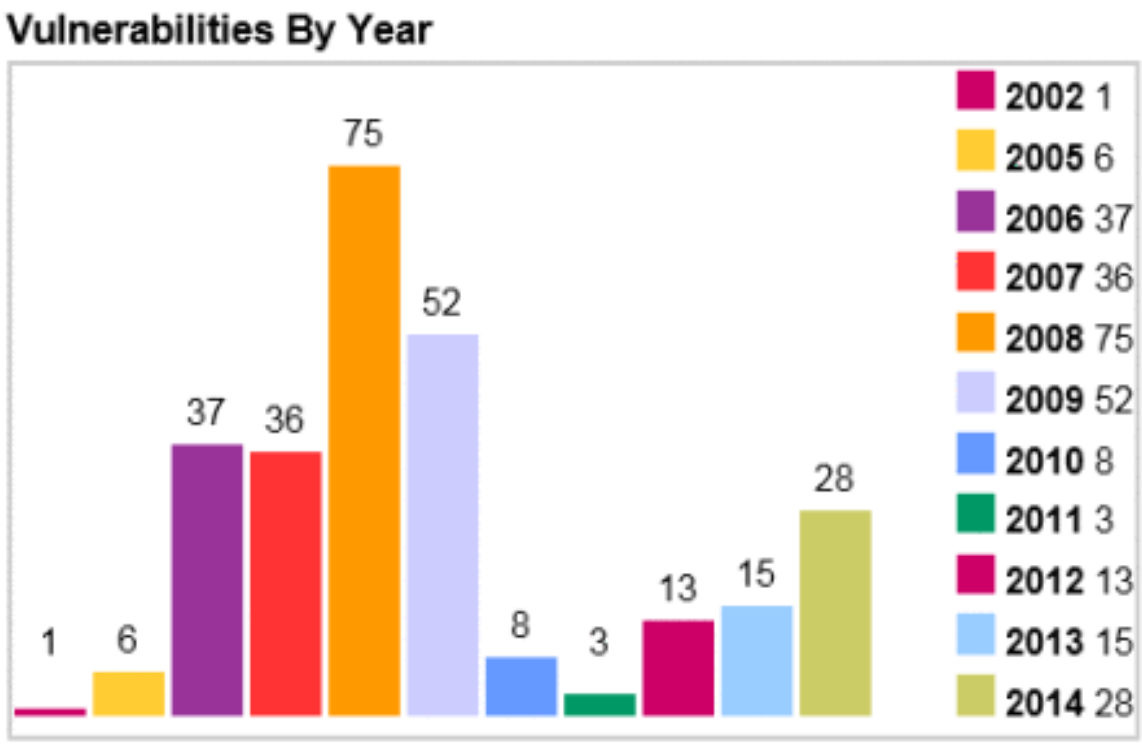
The client wants to reduce risk. Are they prepared to pay for an extensive test or is it a quick health check? The OWASP Top Ten represents an easy benchmark. Scan for the top ten and report. My students then ask what happens if the website is attacked the next day with an attack not on the top ten. Risks are risks. The OWASP Top Ten represents the likelihood of the risks. It does not cover all the risks.

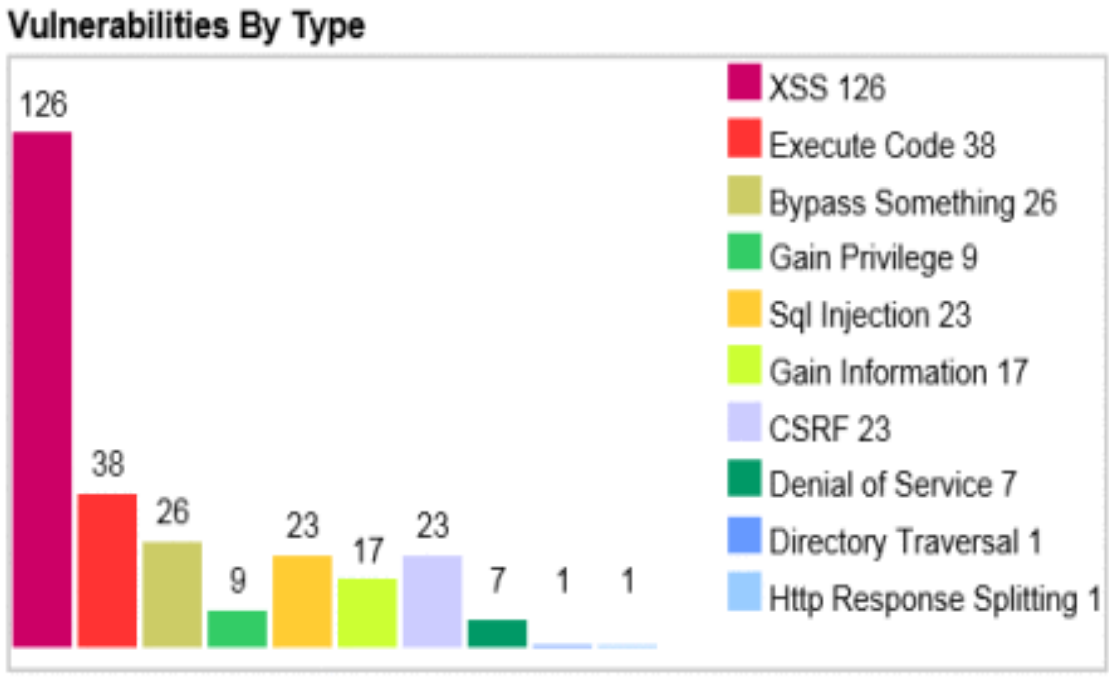
At this point, I show my students the list of bugs detected by platform. This is available at the National Vulnerability Database. There might be 600 vulnerabilities for a Wordpress website and 5 for Magento. If it is Magento then check these separately but if it is a Wordpress check only the more recent.

A comparison of CMS security rankings is shown below:



Drupal: Vulnerability Statistics





Note the charts do not match up. I would trust the National Vulnerability Database but clearly, from the last figure, Drupal has had a major XSS problem. I did a search on the National Vulnerability Database 13 March 2016 for the past three months for Drupal as XSS seems to be still a problem. A good pen test tool would look for these first. Qualysguard representatives would not say if this was one of their vulnerability databases. There is a 3-6 month lag in such publications to allow patches to be applied to US servers.

So the contract sets out expectations and this is where pen testing gets a poor name. I asked for a pen test and my website crashed the next week. Did they break my website with their tests?

The following process of pen test reporting should reduce this problem.

Planning: Automated versus manual testing.

I then show them the article written by Semi Yulianto, Writing an Effective Penetration Testing Report, which appeared in PenTest Magazine 04/2015. This is both an excellent and comprehensive article written from a project manager's viewpoint. As I am a project manager, this report appealed to my sense of order. I show it to explain that there are many choices and the training flows past many other good choices.

- The section Tools of the Trade has a section Vulnerability identification & investigation
- Nmap with NSE (open source)
- Nessus (commercial)
- eEye Retina (commercial)
- Qualysguard (commercial)
- OpenVAS (open source)

Semi also has a good section on the Business Case for pen testing. My objective with this training is to take you through the training process by using the best guides possible. The report by Semi is a good introduction.

Standards

As a pen tester you might want to follow a standard. This allows fewer problems in the contract. The list provided by Semi is shown below:

- NIST SP 800-115, Technical Guide to Information Security Testing and Assessment
- OISSG ISSAF, Information Systems Security Assessment Framework
- ISECOM OSSTMM, Open Source Security Testing Methodology Manual
- OWASP Testing Guide, Open Web Application Security Project
- SANS Institute, Conducting a Penetration Test on an Organization
- PTES, Penetration Testing Execution Standard

I provide the students with NIST Guidelines on Securing Public Web Servers and turn to Sections 6 and 7. It now is a good time to look at running tests on the backup copy of the website. Sections 6 and 7 explain that active content can be a problem so that the actual testing may cause additional problems.

Here is where we can discuss Automated vulnerability scanning versus Manual Pen Testing. Again, the NIST document is a good place to start the discussion:

(Automated) Vulnerability Scanning

However, vulnerability scanners have some significant weaknesses. Generally, they identify only surface vulnerabilities and are unable to address the overall risk level of a scanned Web server. Although the scan process itself is highly automated, vulnerability scanners can have a high false positive error rate

(reporting vulnerabilities when none exist). This means an individual with expertise in Web server security and administration must interpret the results. Furthermore, vulnerability scanners cannot generally identify vulnerabilities in custom code or applications. Vulnerability scanners rely on periodic updating of the vulnerability database to recognize the latest vulnerabilities. Before running any scanner, Web server administrators should install the latest updates to its vulnerability database. Some databases are updated more regularly than others (the frequency of updates should be a major consideration when choosing a vulnerability scanner). Vulnerability scanners are often better at detecting well-known vulnerabilities than more esoteric ones because it is impossible for any one scanning product to incorporate all known vulnerabilities in a timely manner. In addition, manufacturers want to keep the speed of their scanners high (the more vulnerabilities detected, the more tests required, which slows the overall scanning process).

Therefore, vulnerability scanners may be less useful to Web server administrators operating less popular Web servers, OSs, or custom-coded applications.

A list of vulnerability scanners are available at https://www.owasp.org/index.php/Category:Vulnerability_Scanning_Tools

(Manual) Penetration testing

“Penetration testing is security testing in which evaluators attempt to circumvent the security features of a system based on their understanding of the system design and implementation” [NISS99]. The purpose of penetration testing is to exercise system protections (particularly human response to attack indications) by using common tools and techniques developed by attackers. This testing is highly recommended for complex or critical systems.

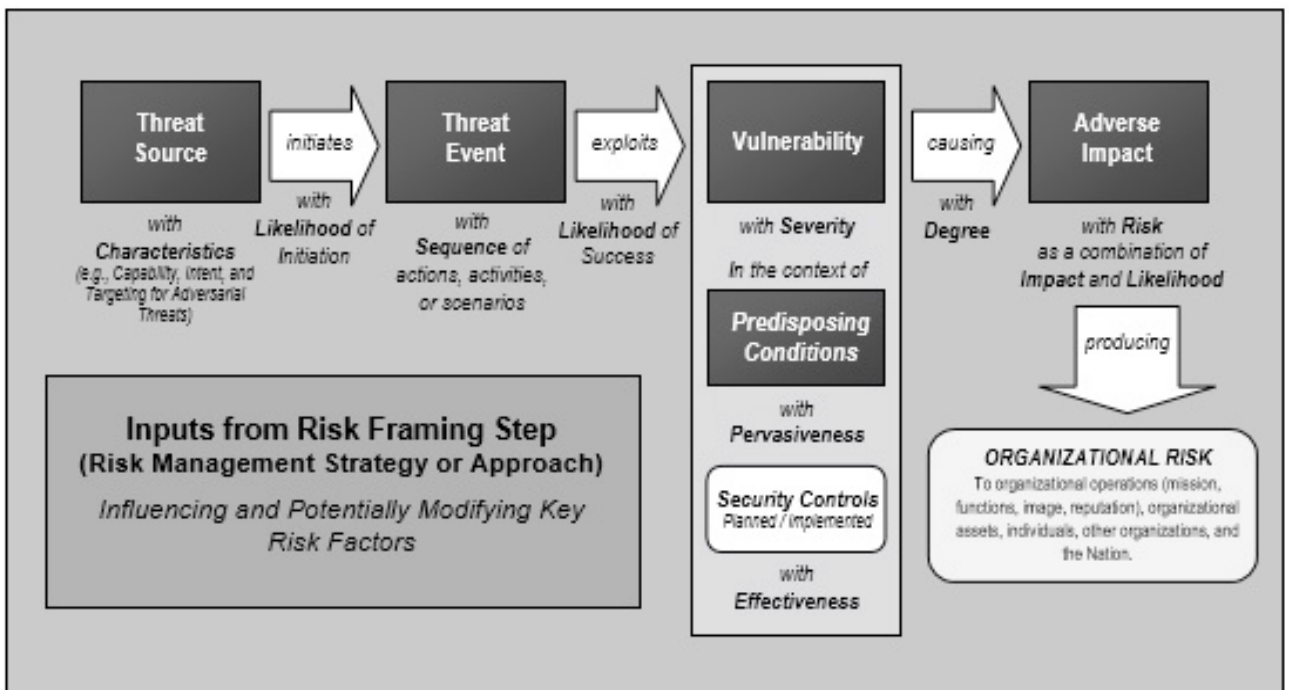
Penetration testing does offer the following benefits [NIST02b]:

- Tests the network using the same methodologies and tools employed by attackers
- Verifies whether vulnerabilities exist
- Goes beyond surface vulnerabilities and demonstrates how these vulnerabilities can be exploited iteratively to gain greater access
- Demonstrates that vulnerabilities are not purely theoretical
- Provides the “realism” necessary to address security issues
- Allows for testing of procedures and susceptibility of the human element to social engineering.

So is Vulnerability scanning really Pen Testing?

NIST sees a distinction. Does it matter to the client? NIST says risk is the net negative impact of the exercise of a vulnerability, considering both the probability and the impact of occurrence. So in our contract should we explain which procedure we will use?

The figure below may assist:



So what is Pen Testing? If we use an ISO27001 viewpoint, which looks at risk the same as NIST (Impact, Likelihood and Threat), the pen test is a snapshot of the business's security at a specific point in time and pen testers use their expertise to identify the Threat source and Threat Event. An automated vulnerability scan misses this expertise in the input section. So you get two snapshots (or watches) and to tell which is correct, you must understand what the snapshot means.

Remember the list of CMS vulnerabilities? You already know which vulnerabilities are a problem for Drupal. Should we select a test tool for better checking XSS? This is a question to be answered in another article.

Planned tests

Qualysguard (automated) and Openvas (manual - in the Kali Linux distribution)

Client would like The OWASP Top Ten (client receives a quote).

I would then use the OWASP Top Ten as my basis of the report. So OWASP Testing Guide is my choice.

Execution Pentesting

Qualysguard this is available from https://www.qualys.com/smb/qualysguard/express-lite

The Freescan https://www.qualys.com/forms/freescan/ (Screenshot of Qualys)

Perimeter scanning detects security vulnerabilities across the entire network. Web application scanning detects vulnerabilities in web applications of all sizes. Malware detection scans websites for malware infections and threats. FreeScan is a free vulnerability scanner and network security tool for business networks. FreeScan is limited to ten (10) unique security scans of Internet accessible assets. FreeScan provides a detailed report that can be used to correct and fix security threats proactively.

Typical results

The format is Threat Impact Solution, which follows NIST layout. There is also the option for a vulnerability scan or an OWASP Top 10 scan.

There were 43 vulnerabilities found in the scan. (Refer to screenshot of Qualys Freescan 2015).

I did not fix any vulnerabilities and in the OWASP there are six vulnerabilities. (Refer to screenshot of QualysFreescan Oct 2015) WTF! Web Test Frameworks generate different reports.

What would you do? The contract asked for an OWASP Top 10 (is it the latest version of the Top 10?) but there are other vulnerabilities. Should a pen tester fix all of these vulnerabilities? Only the high or critical ones?

A good contract will state fix critical and high vulnerabilities.

The difference is the ease of generating reports with threat levels clearly shown. OpenVas has a High Medium Low approach. Openvas spotted 51 vulnerabilities in the same scan.

A typical OpenVas report is attached of a previous scan. (Openvas Forest4trees)

Closure The PenTest Report (Red Queen)

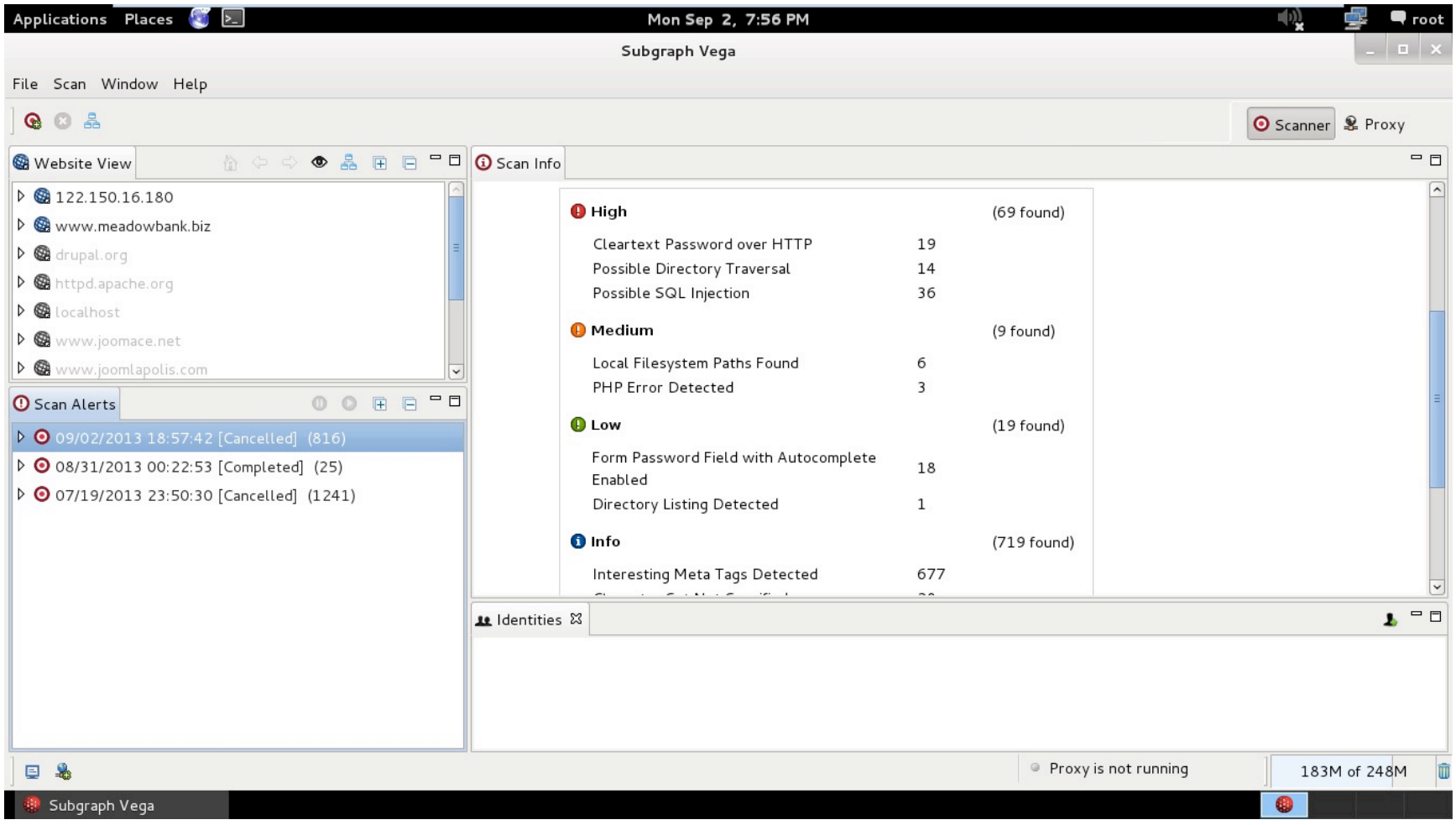
My point from the execution phase is that different pen tools with different settings (OWASP Top Ten) give different numbers of vulnerabilities. It is all about risk level. How do you show that your action reduces this level of risk?

You can produce the test reports from Qualysguard with your company name. There would be an executive summary saying that the following tests were carried out on a certain day with the attached results. It is a snapshot of the system.

A before and after pen test will show improvements and reduced risk. The quality of automatic test reports from Qualys and IBM have been improving over the years.

Or it could just be a series of emails:

> *Hi Bruce,*
>
> *I have attached the results for Joomscan and Openvas for www.forest4trees.biz but we are still working on www.meadowbank.biz. The link below has some security modules for Joomla to be installed.*
>
> *http://extensions.joomla.org/extensions/access-a-security/site-security/site-protection*
>
> *I would sug http://extensions.joomla.org/extensions/access-a-security/site-security/site-protection/12731 gest you install Macro's SQL InJection and install any other firewall from the first link.*



Lessons learned

I ask what makes a good pen testing report. What is the correct pen testing tool to spot the more critical vulnerabilities or meet client expectations? Is it a just a risk report which should be part of a system?

Which tool gives you the best results? Some of my students had answers such as this comment below:

Nearly all VA solutions depend upon version checking as their primary method of assessing the relative vulnerability of network hardware or software. VA solutions typically look at the response header and from the version data there, they deduce whether the hardware or software is vulnerable. If an old version is known to have five vulnerabilities and the header says that the old version is in use, then it is assumed that all five of those vulnerabilities exist.

Version checking has many advantages for the vendor and one key disadvantage for the customer. It is easy to program and claim '45,000 tests'. Also, a version analysis scan that finds an old version can produce a long and impressive list of vulnerabilities. This makes the solution look good.

The disadvantage: poor accuracy misses real problems and list dozens, if not hundreds, of vulnerabilities that don't actually exist. Version information contained in a header doesn't reflect the presence or absence of a security issue with high accuracy.

Client - Concern from client: Is Pen Testing with good tools enough?

We are inclined to think that testing the security of web applications with these tools is not enough. The tools are not capable of copying the versatility and intelligent decisions of a human auditor. The more fundamental problem is that the tools always lag behind. New attacks will surface all the time and some attacks are specific only to the application being tested.

Also, all of the tools assessed take a black-box approach to testing. They do not know the inside mechanisms employed by the application. A security expert doing a traditional audit can have inside information of the application and an attacker can have it as well.

The current offering of tools clearly lacks a white-box web application security testing tool that would make use of inside knowledge of the application. Attacks can come from within the organization.

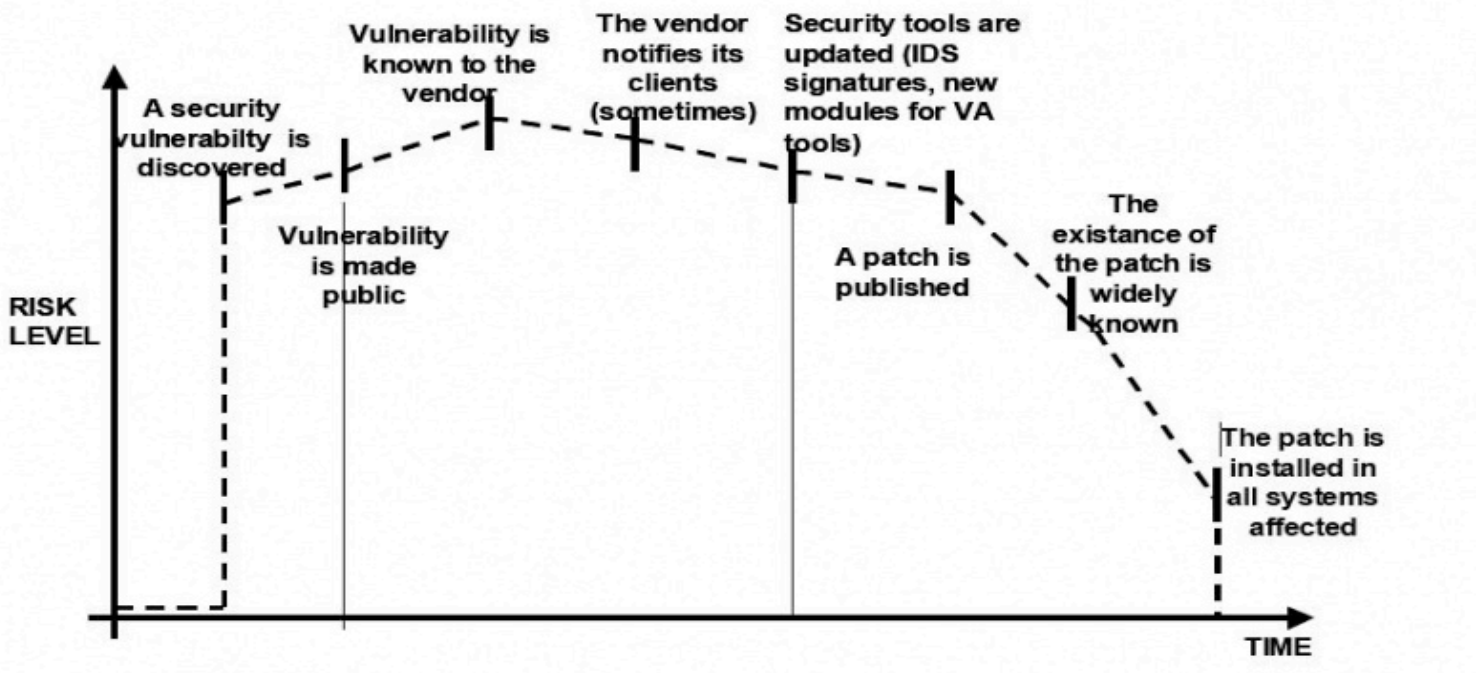
New types of security tests should be available to test tools as early as possible. Having a separate vulnerability and attack database for every scanning tool is a major source of problems. None of these databases can ever contain all vulnerabilities. It adds to the problem if the databases are closed and it is impossible to be sure which vulnerabilities are included.

The students most probably downloaded this off the internet. The end point is that they saw that the pen test tool limitations must be understood for a risk report.

Their verbal answers were:

- Automated testing is only a good starting point.
- Manual frameworks are needed to cover vulnerabilities the automated testing systems do not detect.
- The risk level changes over time as shown below.

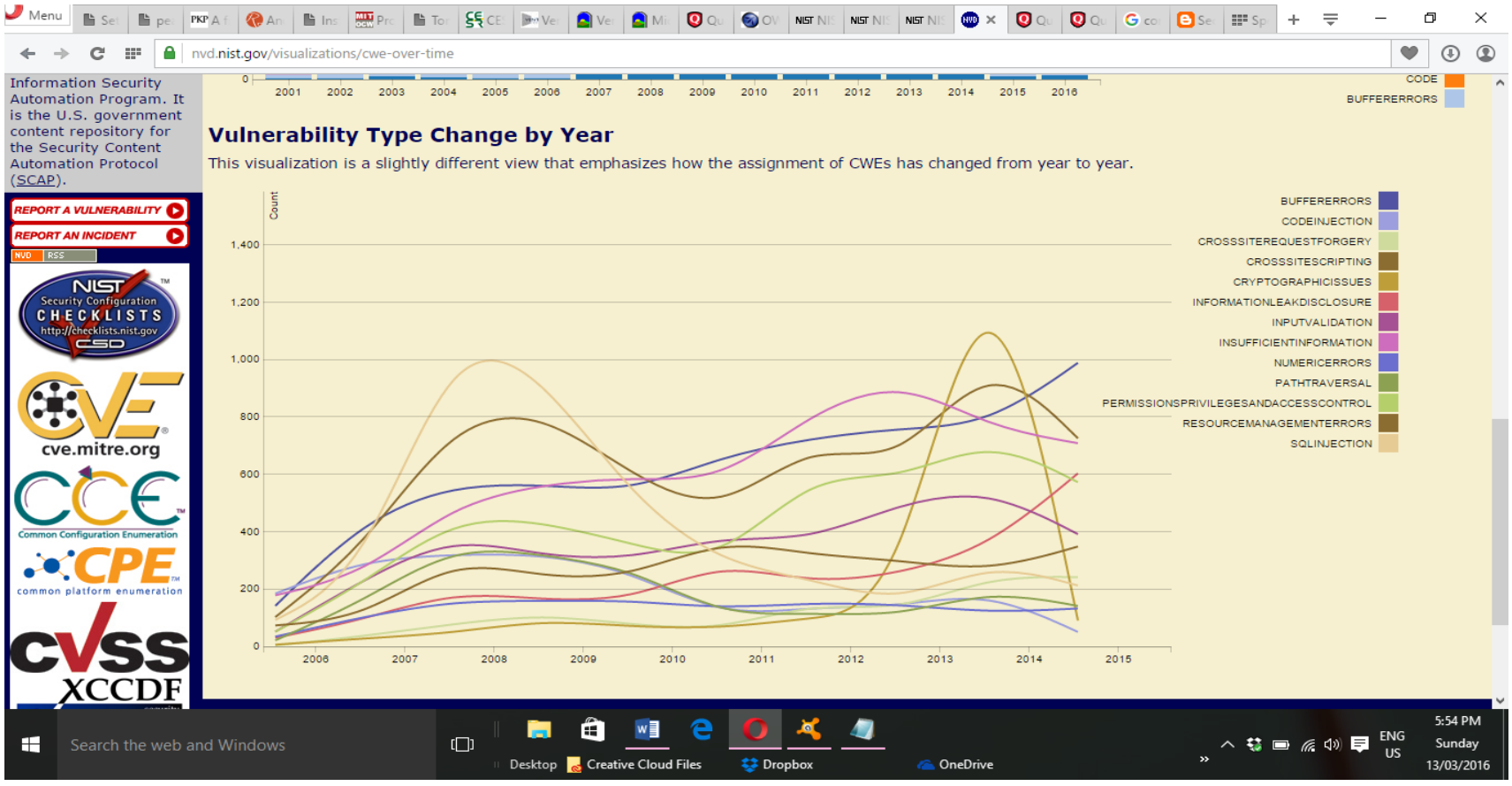
I had a situation with a client's website. The website was under heavy attack. The business website had been blocked by Google for containing malware. I spotted that the plugins supplied to the CMS were the main cause of the problem. This is on the left of the figure below. I contacted the plugin supplier (I later found that they supplied the slideshow plugin under different names to different theme suppliers). Their programmers said that the critical issues were not a concern, false positives. I disagreed. The sequel to this was that others found that there were major problems, the plugin supplier lost all their credibility and themes suppliers for CMS were in big trouble. The time from when the vulnerability is detected (that is, you are under heavy attack) to when a patch is supplied was months. The solution was a new static website (no plugins). Customer was happy but said why should I have to pay for their faults?



Gary McGraw summed up penetration testing well when he said, “If you fail a penetration test, you know you have a very bad problem indeed. If you pass a penetration test, you do not know that you don’t have a very bad problem”. However, focused penetration testing (i.e., testing that attempts to exploit known vulnerabilities detected in previous reviews) can be useful in detecting if some specific vulnerabilities are actually fixed in the source code deployed on the web site.

A Note about Web Application Scanners

Many organizations have started to use automated web application scanners. While they undoubtedly have a place in a testing program, some fundamental issues need to be highlighted about why it is believed that automating black box testing is not (or will ever be) effective. However, highlighting these issues should not discourage the use of web application scanners. Rather, the aim is to ensure the limitations are understood and testing frameworks are planned appropriately.



In parting, I believe that a great pen tester knows trends. One such trend is found at NVD. A human can remember such a graph and act accordingly.

Author: Bruce Williams

Bruce is a systems/telecommunications engineer with a Masters in Engineering Science from Sydney University Australia. He also has Diplomae in Adult Training and Sustainability.

He spent 20 years as Assistant Director for the Australian Department of Industry helping start-up companies in IT. He then switched to training and has spent 20 years training adult learners in business and IT at TAFE Sydney and Southern Cross University on the Gold Coast.

References:

http://www.ncsc.gov/nittf/docs/CNSSI-4009_National_Information_Assurance.pdf""http://www.ncsc.gov/nittf/docs/CNSSI-4009_National_Information_Assurance.pdf

https://en.wikipedia.org/wiki/Multitier_architecture"https://en.wikipedia.org/wiki/Multitier_architecture

https://www.owasp.org/index.php/OWASP_Secure_Software_Contract_Annex"https://www.owasp.org/index.php/OWASP_Secure_Software_Contract_Annex

https://nvd.nist.gov/full_listing.cfm"https://nvd.nist.gov/full_listing.cfm

http://malwarelist.net/2014/10/16/the-critical-vulnerability-in-the-web-content-management-system-drupal/"http://malwarelist.net/2014/10/16/the-critical-vulnerability-in-the-web-content-management-system-drupal/

https://www.qualys.com/enterprises/qualysguard/vulnerability-management/"https://www.qualys.com/enterprises/qualysguard/vulnerability-management/

http://csrc.nist.gov/publications/nistpubs/800-44-ver2/SP800-44v2.pdf"http://csrc.nist.gov/publications/nistpubs/800-44-ver2/SP800-44v2.pdf

http://csrc.nist.gov/publications/nistpubs/800-30-rev1/sp800_30_r1.pdf"http://csrc.nist.gov/publications/nistpubs/800-30-rev1/sp800_30_r1.pdf

http://www.iso.org/iso/catalogue_detail?csnumber=54534"http://www.iso.org/iso/catalogue_detail?csnumber=54534

https://www.owasp.org/index.php/Testing_Guide_Introduction"https://www.owasp.org/index.php/Testing_Guide_Introduction

http://www.drdobbs.com/security/beyond-the-badness-ometer/189500001"http://www.drdobbs.com/security/beyond-the-badness-ometer/189500001

https://www.owasp.org/index.php/Testing_Guide_Introduction"https://www.owasp.org/index.php/Testing_Guide_Introduction

https://nvd.nist.gov/visualizations/cwe-over-time"https://nvd.nist.gov/visualizations/cwe-over-time

HOW TO CREATE GOOD PENETRATION TESTING REPORT

ONLINE COURSE

4 modules

Sample reports

Course eBook

Click here

Approach by Alex Torres

In order to fully understand what needs to be done with the penetration testing report, several areas need to be discussed. These areas include, but are not limited to: the reason for the penetration testing report, keeping the reports secure, methods of remediating one's environment against this report, and a review of limitations of the report. A brief overview of the major penetration testing methodologies will also be discussed, followed by a discussion on how to measure a successful penetration testing program. Armed with this information, we should have the tools and techniques to properly manage a penetration testing report.

Reasons for the Penetration Testing Report

Organizations normally require penetration testing reports for either compliance requirements, to satisfy the implementation of a security framework, or to mitigate risk. “The purpose of the report is to assist the organization in its efforts to improve its security posture by identifying areas of potential risk that may need to be remediated” (PCI, 2015). To expand on this, we can look into reasons why these organizations conduct a penetration test to begin with. These include:

- To reduce risk
- To identify and remediate high risk vulnerabilities
- To test one's security posture and defenses
- To test new systems or security configurations
- To protect a company’s reputation
- To satisfy cyber insurance requirements
- To give management an overall view of a company’s exposure
- To provide evidence in support of budgetary increase (Basu, 2013)

To extend the need for the penetration test reports, understanding the various penetration tests will assist in understanding their overall value to organizations.

The various types of penetration test include, but are not limited to:

- Network penetration test – This can be either an internal or an external test, it can include wireless testing, telephone system/ VoIP test, and vulnerability scanning.
- Application penetration test – This can include web application, custom applications, mobile applications, industrial control systems (SCADA), and database testing.
- Website penetration test – This type of test can review server configuration problems, can test for additional attack vectors within the website, it can test for SQL injection and Cross-site scripting vulnerabilities to name a few.
- Physical penetration test – Can test an organization's physical security, to include lock-picking, impersonation, or bypassing any other security measure.
- Cloud penetration test – Given the push into cloud computing, this is a new area that needs to be understood and reviewed. It includes testing Amazon web services and Azure requirements.
- Social engineering – Testing methods may include phishing, tailgating, password resets, imposters, and other social engineering scams (Morella, 2015).

Keeping the Reports Secure

With the penetration test report in-hand, organizations need to assign the person or team in charge of this report. With this charge comes the responsibility to: secure, assign, remediate, and document. This document needs to be handled as if it were one of the most secure documents within an organization. After all, it contains the details on how to penetrate and circumvent an organization's current security posture. With this in mind, logs need to be kept as to who accesses this document and for what purposes. In larger organizations, it is not common to see one person in charge of remediating all company assets. Normally, different teams or departments are charged with remediating their environment.

The penetration test report may include details on different departments. Due to this, the custodian of the document should only relinquish portions of this document, as needed by the department. Again, careful logs should be maintained on who accesses the various portions of this document. The custodian should be working with the various departments to assign an individual point of contact or resource to assist with remediation efforts within that department. Verification or remediation efforts can take place by conducting an audit of the system, retesting the system and its components, and holding personnel accountable through documentation (NIST SP 800-115).

The responsibility of this custodian should also be to ensure that the various individuals assigned to that person remediate all findings on the penetration testing report. An important note on the penetration testing report is that it is a snapshot in time. Systems are in a constant state of flux; remediation teams should be thinking outside the box to ensure that all areas under their control is secure. The final penetration testing report should also be properly archived and compared to future penetration testing reports to ensure that all areas have been properly secured. When the organization has completed remediating the environment, the tester should perform a retest to validate the newly implemented controls mitigate the original risk (PCI, 2015).

Methods to Remediating One's Environment with this Report

The custodian of this report should create some type of log detailing who has access to the various portions of the report, and who is remediating the various sections. Handling remediation efforts in a project management format will assist in ensuring that all aspects of remediation are documented and implemented. Security is paramount for a successful remediation. Subject matter experts need to be brought into remediation efforts for completeness.

An example here is warranted. If Windows servers are found to have exploitable vulnerabilities, Certified or experienced Windows Server Administrators should be brought in, along with security professionals to discuss remediation techniques. The pen testing report probably did not find every Windows Server weakness or attack vector in an organization. By incorporating subject matter experts, discussions can be started along with following best practice methods by the various system manufacturers.

Project management tools can be incorporated in order to maintain the documents and to follow a systematic approach to managing the remediation project. These tools need to be on a system that is hardened with controlled access to limit its exposure. These tools can be further utilized to set time limits on remediation efforts, to provide an overall snapshot of the remediation efforts, and to assist with resource allocations.

Limitations of the Report

It should be important to note that organizations should not only rely on penetration reports to secure their environments; vulnerability assessments, system logs, firewall logs, and system alerts should also be constantly reviewed for areas of concern. The penetration test report should assist in filling the gap and looking for attack vectors that these processes where unable to find. As noted, this report can carry with it several limitations that need to be reviewed:

- limitations on the tester
- limitations on the environment or scope
- limitations on the results
- limitations on the tools and techniques utilized

One major issue or concern with penetration testing is that two people conducting a penetration test against the same environment will yield different results. This may be due to the experience or education of the penetration tester, limitations on the scope of the penetration test, or limitations on the time allotted for the test.

Organizations need to ensure that penetration tests are conducted against systematic documented formats that can be duplicated. The Institute for Security and Open Methodologies Open Security Testing Methodology Manual (ISECOM OSSTM) provides an excellent source for businesses to adopt to ensure the above objectives were met. There will always be differences in penetration reports, however, there should be areas of uniformity to ensure duplication.

Limitations in the scope of the penetration test itself can determine the overall effectiveness of the penetration test. There needs to be a give and take when it comes to penetration testing. Organizations need to loosen the vines and allow the penetration tester to test as much of an environment as possible. On the other hand, the penetration tester should make every effort not to intentionally harm or take down a network to show proof of concept.

The limitation of the results of a penetration test deal with not only the format of the report, but the content. The results of a penetration test should be in a format that can be understood and remediated. The technical aspects of the attack vector can be re-phrased in a format that is understood by the lay person. This report will be presented to upper management and may be utilized to increase staff or budgets. As such, it should be written at a high level. Once remediation efforts begin, a technical penetration testing report can be utilized for remediation efforts.

Finally, the limitations on the tools and techniques utilized deal with the applications, programs, or scripts utilized by the penetration tester. Every tool has its limitations and these limitations need to be briefly discussed within a report to ensure completeness. Google Dorks provides an excellent resource for external penetration tests, yet the tester needs to understand the various techniques as well as their ramifications on the network. Simply typing that a penetration tester ran a script or a line of code without explaining the script can result in limiting any remediation efforts. A full explanation needs to be given on what was done, why it was done, what it effected, what should have been the intended outcome, and how to fix the issue.

An effective penetration program should also not only be a yearly endeavor or only utilized to satisfy compliance requirements, penetration testing should be an ongoing activity.

Penetration Testing Methodologies

There are several penetration testing methodologies to assist organizations in both conducting and remediating penetration tests. These include the ISSECOM OSSTMM, The National Institute of Standards and Technology (NIST) Special Publication (SP) 800-115, The Open Web Application

Security Project (OWASP) testing guide version 4, The Penetration Testing Execution Standard, The Penetration Testing Framework and the PCI Data Security Standard.

OSSTMM is a comprehensive penetration testing methodology with a complete, repeatable, measurable, and quantifiable framework. It utilizes a standard Security Test Audit Report (STAR) as its foundation for this methodology allowing penetration testers to incorporate a systematic approach to penetration testing. “Its purpose is to serve as an executive summary of precise calculation stating the Attack Surface of the targets tested within a particular scope” (OSSTMM, 2010).

The NIST SP 800-115 is probably the most complete framework of all of the mentioned methodologies. It discusses every aspect of the penetration test from planning to remediation. The only issue or concern with this methodology is that it is not repeatable. Different penetration testers utilizing this methodology will yield different results. The reason for this is that there is too much room for interpretation. The authors of this framework have tried to cover all aspects of a penetration test within an enterprise level company or government agencies. Unfortunately, many businesses do not fall under this category, so they are left adjusting this framework for their needs. It is this adjusting that leaves too much room for inconsistencies. This framework is best used as a reference for the other frameworks.

The OWASP framework is an application and web testing framework. It is specific enough for a targeted penetration test, yet broad enough to cover many aspects of an application or web penetration test. This framework is best utilized in conjunction with other penetration testing frameworks due to its targeted attack vector of web applications.

The Penetration Testing Execution Standards provides an excellent resource as to the commands or tools utilized during a penetration test. It provides only basic documentation examples, and does not delve into remediation efforts. Similarly, the Penetration Testing Framework also provides similar technical details on conducting a penetration test. It goes a bit further in discussing the compliance requirements and pre-inspection efforts. Unfortunately, both of these methodologies lack in their remediation or repeatability efforts.

The Credit Card Industry has purposed, within their PCI compliance requirement, a penetration testing format. To assist with this, PCI included a penetration testing standard. Although not as comprehensive as OSSTMM or NIST SP 800-115, it is worth looking into. It can be viewed as an abbreviated form of NIST SP 800-115.

How to Measure the Success of a Penetration Testing Program

Unlike other security programs, like firewall configurations or network security policies, penetration testing programs can be difficult to measure. The effectiveness of any program needs to be quantifiable in order to be able to measure it. Once it can be measured, a comparative analysis can be conducted over time to show effectiveness. Looking at the various penetration testing methodologies, the one that stands out the most is the OSSTM. It provides for a measurement of exploitable vulnerabilities in a quantifiable format.

An effective penetration program should also not only be a yearly endeavor or only utilized to satisfy compliance requirements, penetration testing should be an ongoing activity. Previously, we named six

types of penetration tests: network, application, website, physical, cloud and social engineering. These penetration activities can be conducted at different times. The areas can further be divided by operation environment or by organizational departments. By constantly conducting penetration tests, organizations are provided with an environment that fosters security and remediation. Security professionals and system administrators can monitor their environment while penetration testers attack the various environments. This way, an updated penetration testing report can be maintained and remediated to ensure up to date security practices are maintained and any new attack vector is remediated.

Author: Alex Torres

Alex Torres is a Threat and Remediation Engineer at Hewlett Packard Enterprise specializing in Vulnerability Assessments and PCI compliance. He also has had over 10 years of experience as a security consultant for various healthcare businesses. Alex has an MS in Information Systems Security from American Military University, his CISSP as well as other security certifications.

References

Alharbi, M. (2010, April). SANS -writing a penetration testing report. Retrieved from https://www.sans.org/reading-room/whitepapers/bestprac/writing-penetration-testing-report-33343

Basu, E. (2013, October). What is a penetration test and why would i need one for my company. Retrieved from http://www.forbes.com/sites/ericbasu/2013/10/13/what-is-a-penetration-test-and-why-would-i-need-one-for-my-company/#4f53639242da

ISECOM. (2010). Open source security testing methodology manual. Retrieved from http://www.isecom.org/mirror/OSSTMM.3.pdf

Morella, R. (2015, August). Penetration testing methods and strategies. Retrieved from http://www.isaca.org/chapters3/Atlanta/AboutOurChapter/Documents/GW2015/081115-10AM-Pentesting.pdf

NIST Special Publication 800-115. (2008, September). Technical guide to information security testing and assessment. Retrieved from http://csrc.nist.gov/publications/nistpubs/800-115/SP800-115.pdf

Open Web Application Security Project. (2015, August). Testing guide version 4. Retrieved from https://www.owasp.org/images/5/52/OWASP_Testing_Guide_v4.pdf

PCI Security Standard Council. (2015, March). Penetration testing guidance. Retrieved from https://www.pcisecuritystandards.org/documents/Penetration_Testing_Guidance_March_2015.pdf

The Penetration Testing Execution Standard. (2014, August). Reporting - the penetration testing execution standard. Retrieved from http://www.pentest-standard.org/index.php/Reporting

VulnerabilityAssessment.co.uk. (2014). Report template. Retrieved from http://www.vulnerabilityassessment.co.uk/report%20template.html

Approach by

Paulo H., Juliano S., Mike G., Renato B., Thiago S. and Thiago F.

This article is part of research on invasion methods underway at the University Nove de Julho (UNINOVE, Brazil) under the coordination of Dr. Paulo Henrique Pereira. The project aims to carry out penetration tests for the analysis of vulnerabilities in servers, web applications and operating systems, including mobile. The research aims to analyze the potential that a cybercriminal could gather as tools of invasive and invasive attacks and that the attacker could do when she accesses their targets.

The project has two stages:

a) The first level is dedicated to obtaining the signature files saved on the server, using forensics and intrusion techniques.

b) The second level, which is not covered by this article, is dedicated to the use of invasive techniques to take control of the server.

Scenario for Penetration Test

The first stage of the penetration testing lab project consists of the server farm to find the preliminary signature of the file that is saved in the root level. For the realization of the project, two teams were created without the possibility of communication between them. One team aimed to create the laboratory to be invaded (Renato, Thiago), simulating a real corporate environment. The other team aimed to break into the server and capture a file that has sensitive data on that server (Juliano, Thiago and Mike). As a test lab, a small infrastructure was created (by Renato e Thiago) consisting of a router, switch, server, firewall and IDS (Figure 1). The target was a GNU / Linux server on which were installed the following tools for monitoring and analysis: Mysql, Apache2, Snort, htop, tcptrack, iptraf, goaccess, rkhunter, Lynis, lastcomm, ssh.

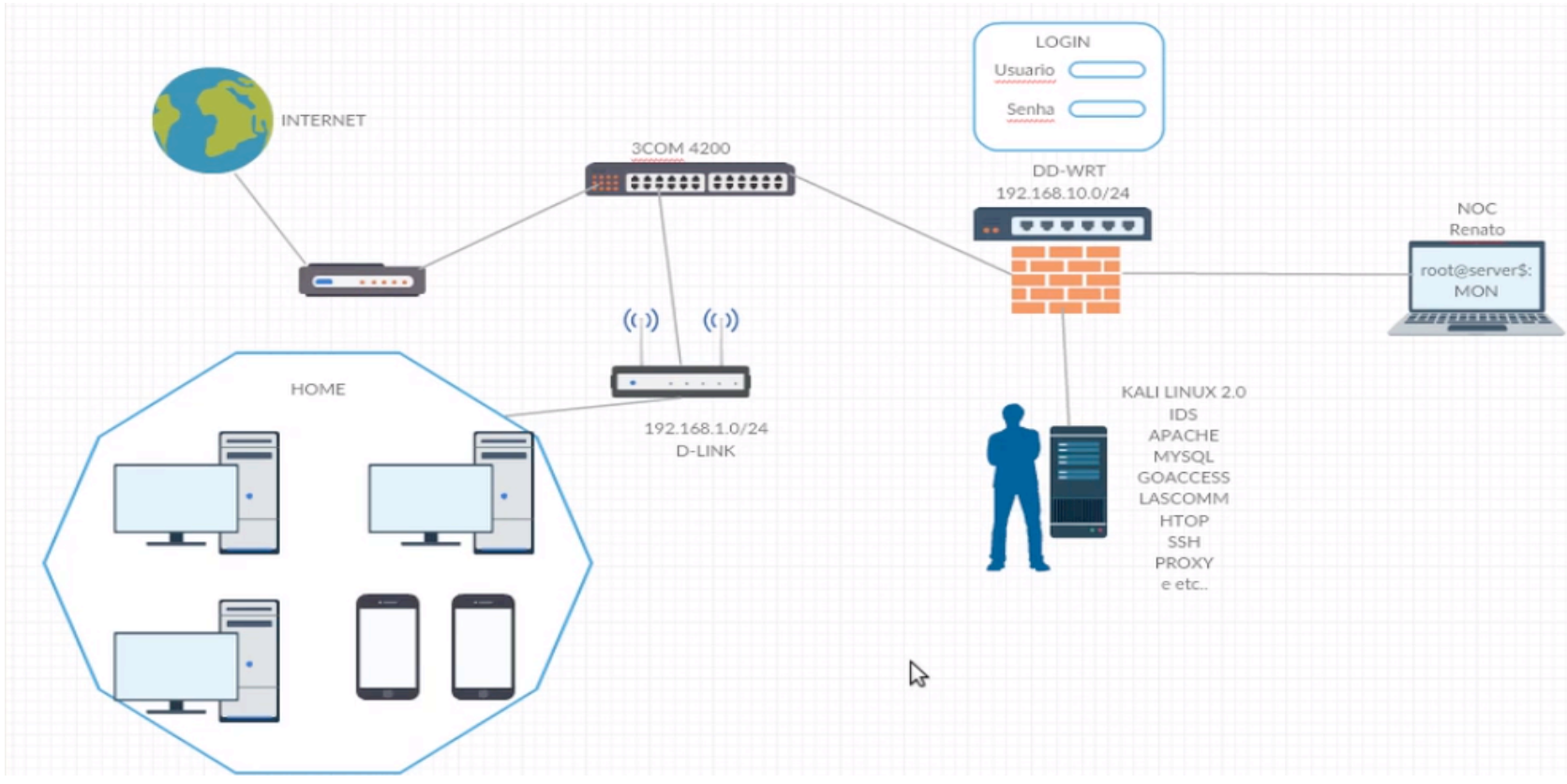


Figure 1: Network structure in the laboratory to Penetration Test.

Purpose of the PenTest

The main objective of the PenTest was to propose the capture of a file saved at root level on the server. The attacking team should find vulnerabilities in the network to access the web of the server or the attack on the network, access the server for remote access to the target server shell to find a private TXT file hidden behind an image. The attack methods the attacking team could use were not limited to the technical knowledge of hacking, but extended to social engineering vectors with the vectors of exploitation attacks with e-mail, telephone approach, and different polls (WATSON 2014).

Technical exploitation and methodologies

Two penetration testing methodologies were followed in this first stage: OWASP and Backtrack; testing the code fails on the server web page (OWASP) and attempts to using brute force attacks (Backtrack). Although Backtrack has been discontinued and in its place Kali Linux has emerged, the penetration testing methodology has not changed because many of the tools migrated and are available in Kali. The team of attackers used the black-box approach.

To identify which points would be potentially possible loopholes that the infrastructure of the network and the server had, an environment analysis for information to start the PenTest challenge using the technique of footprinting was carried out by following the sequence of layers to the server:

After a survey of information and documentation, open source tools were applied (relevant to each required activity) present in the Kali Linux, except `ISunShare` application, Shareware for the Windows platform.

The aim is discoverable by web `http://borbollanetwork.ddns.net` address. The first made access revealed a login page with PHP technology. The IP of this field obtained this date is 177.18.210.46, achieved by using the Ping tool:

```
ping borbollanetwork.ddns.net
```

A Google search with the keyword "ddns.net" shows that the target IP address has been appointed by the DNS service available in http://freeddns.noip.com that belongs to the site http://www.noip.com. A first questionable point is whether there would be relevant information to obtain access to the service through the password recovery feature available in http://www.noip.com/forgot-password.

After achieving the admin user access at ssh server was found on the server's internal structure would be any file that has the ability to raise suspicion that an end user would not see. Thus, a JPG file was found in the root directory. The team analyzed this file and realized that there was an image steganography hiding another file.

Tools	Port scan
Nikto	**Serviço de e-mail**
Owasp Zap	**Conexão remota via SSH**
Sqlmap	**NetBIOS Session Service**
Vega	**Squid Proxy Server**
Cookie Inspector	**Mini servidor Apache HTTPD**
Vega	**Microsoft Remote Procedure Call**
Hydra-GTK	**NetBIOS sobre TCP/IP e SMB/CIFS**

After using web crawler’s tools, an open directory was found (public access) and even had an important file for phpsysinfo application, the php.ini configuration file. Thus, we sought to examine the page source, and, even for an unexpected half, there was another vulnerability that compromised the user's password "admin" on the server web page. Through the analysis of the source code, it was discovered that the source contained the password "s3nh4Uni9" and instructions to make an SSH connection over port 2222 to the target server with the user "admin". After gaining access via SSH with the command (Figure 2):

```
ssh admin@179.178.177.160 -p 2222
```

view-source:borbollanetwork.ddns.netphpsysinfo/phpsysinfo.ini

```
; PSI Config File
;
; @category  PHP
; @package   PSI
; @author    RENATO BASANTE BORBOLLA
; @copyright 2016 phpSysInfo
; @license   http://opensource.org/licenses/gpl-2.0.php GNU General Public License
; @version   SVN: $Id: phpsysinfo.ini.new 705 2012-11-11 00:33:29Z namiltd $
; @link      http://phpsysinfo.sourceforge.net
; @SSH       SSH admin@borbollanetwork.ddns.net
; @PORTA     2222
; @AVISO     USER ADMIN (s3nh4Uni9)
; @AVISO1    Server 2016

[main]
; ********************************
;        MAIN PARAMETERS
; ********************************

; Turn on debugging of some functions and include errors and warnings in xml and provide a popu
; - false : no debug information are stored in xml or displayed
; - true : debug information stored in xml and displayed *be careful if set this to true, may i
;
DEBUG=false

; Turn on logging/unlogging of functions executeProgram() and rfts()
; example : executeProgram () and rfts () record the results to the "/tmp/phpsysinfo.log" file
; LOG="/tmp/phpsysinfo.log"
; example : executeProgram () and rfts () read the results from the "/tmp/phpsysinfo.log" file
; LOG="-/tmp/phpsysinfo.log"
; example : executeProgram () and rfts () read the results from the "/tmp/phpsysinfo.log" file
; LOG="+/tmp/phpsysinfo.log", if lack in the log file it runs normally
;
LOG=false

; Turn on/off compression for JavaScript file
; - JS_COMPRESSION=false  //no compression (recommended with slow processor)
; - JS_COMPRESSION="None"  //code minimizing
; - JS_COMPRESSION="Normal"  //code packing
;
JS_COMPRESSION="Normal"

; Additional paths where to look for installed programs
; Example : ADD_PATHS="/opt/bin,/opt/sbin"
;
ADD_PATHS=false
```

Figure 2: Access to data from the server.

As soon as we got the admin user and password login site, we also use the system, but got no success because the password was not the same. The attacking team used brute force tools using the password that was obtained on the site, but increasing some special characters to the invasion by ssh to succeed. It was soon completed the stage with the password: s3nh4Uni92016. (Figure 3).

```
s3nh4Uni91nv4s40
s3nh4Uni92016
s3nh4Uni92222
s3nh4Uni93nc0ntr4r
s3nh4Uni93ntr4r
s3nh4Uni94c3ss0
s3nh4Uni94c3ss4r
s3nh4Uni94dm1n
s3nh4Uni94dm1n@b0rb0ll4n3tw0rk.ddns.n3t
s3nh4Uni94rqu1v0
s3nh4Uni94rq|_|1v0
s3nh4Uni94ss1m
s3nh4Uni9Acessar
s3nh4Uni9Acesso
s3nh4Uni9Admin
s3nh4Uni9Admin@borbollanetwork.ddns.net
s3nh4Uni9Arquivo
s3nh4Uni9B0rb0ll4
s3nh4Uni9B4s4nt3
s3nh4Uni9Basante
s3nh4Uni9Borbolla
```

Figure 3: Brute-force Attack.

The team of attackers found the suspect file, a jpg called "imagem7.jpg" that was held by the root. The copy of the remote file to a local machine for forensic analysis was done. With the use of Foremost tool, it was identified that there was something behind the image. We obtained a new JPG image identical to that under review and a RAR file protected.

image7.jpg

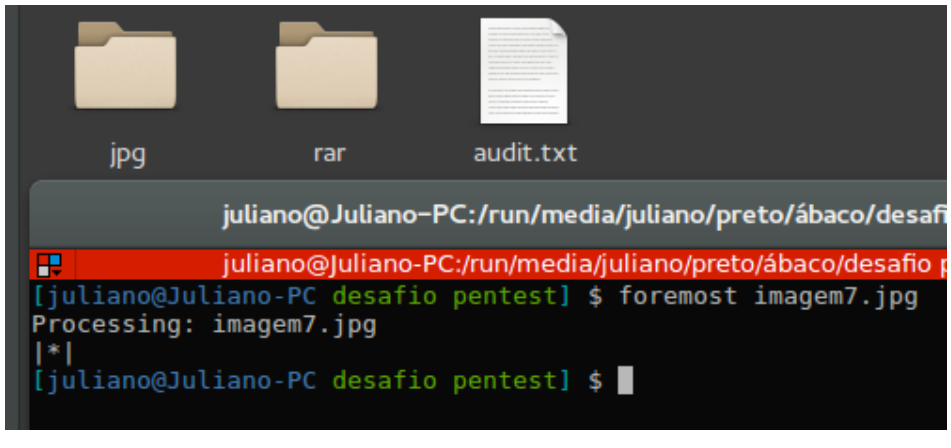


Figure 4: Using Foremost.

Steganography analysis performed with the toolkit that accompanies the utility stegbreak and stegdetect command in images obtained:

```
$ stegdetect 00000000.jpg

00000000.jpg : jphide(*)

$ stegdetect imagem7.jpg

imagem7.jpg : jphide(*) appended(1284)<[nonrandom][data][Rar!.....s......]>
```

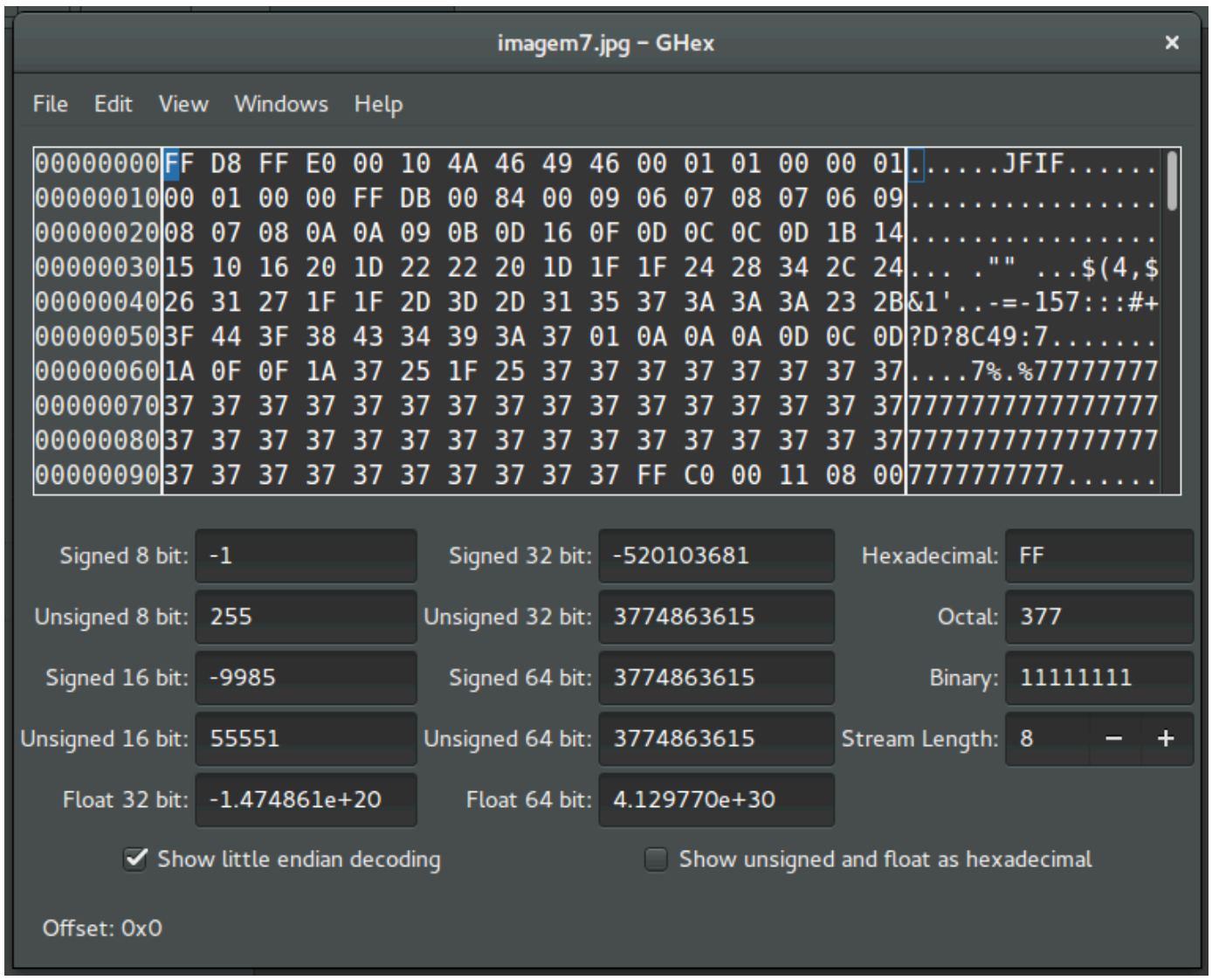
According to the server security team, the confidential file we seek is a TXT file. According to stegdetet tool, the search file may have been hidden in any one of two images with jphide tool. The attempt was to perform a brute force attack, but without any success against the files to try to extract the contents of the images by performing the reverse process of jphide tool with stegbreak. To use the stegbreak, you need a rules file; the file used in the test was obtained from:

https://raw.githubusercontent.com/poizan42/stegdetect/master/rules.ini

Initially, a wordlist was created with the crunch tool and run the tool stegbreak with this wordlist as the commands:

```
crunch 1 8 0123456789 -o wordlist2.txt

stegbreak -c imagem imagem7.jpg

stegbreak -c imagem 00000000.jpg

stegbreak -r rules.ini -f wordlist2.txt -t p imagem7.jph

stegbreak -r rules.ini -f wordlist2.txt -t p 00000000.jph
```

In search of some relevant information, without success, an analysis of the image files and RAR file with the hex editor GHex (Figure 5-6) was carried out, but still it is manually possible to separate the files. The signature was not found in the TXT file.



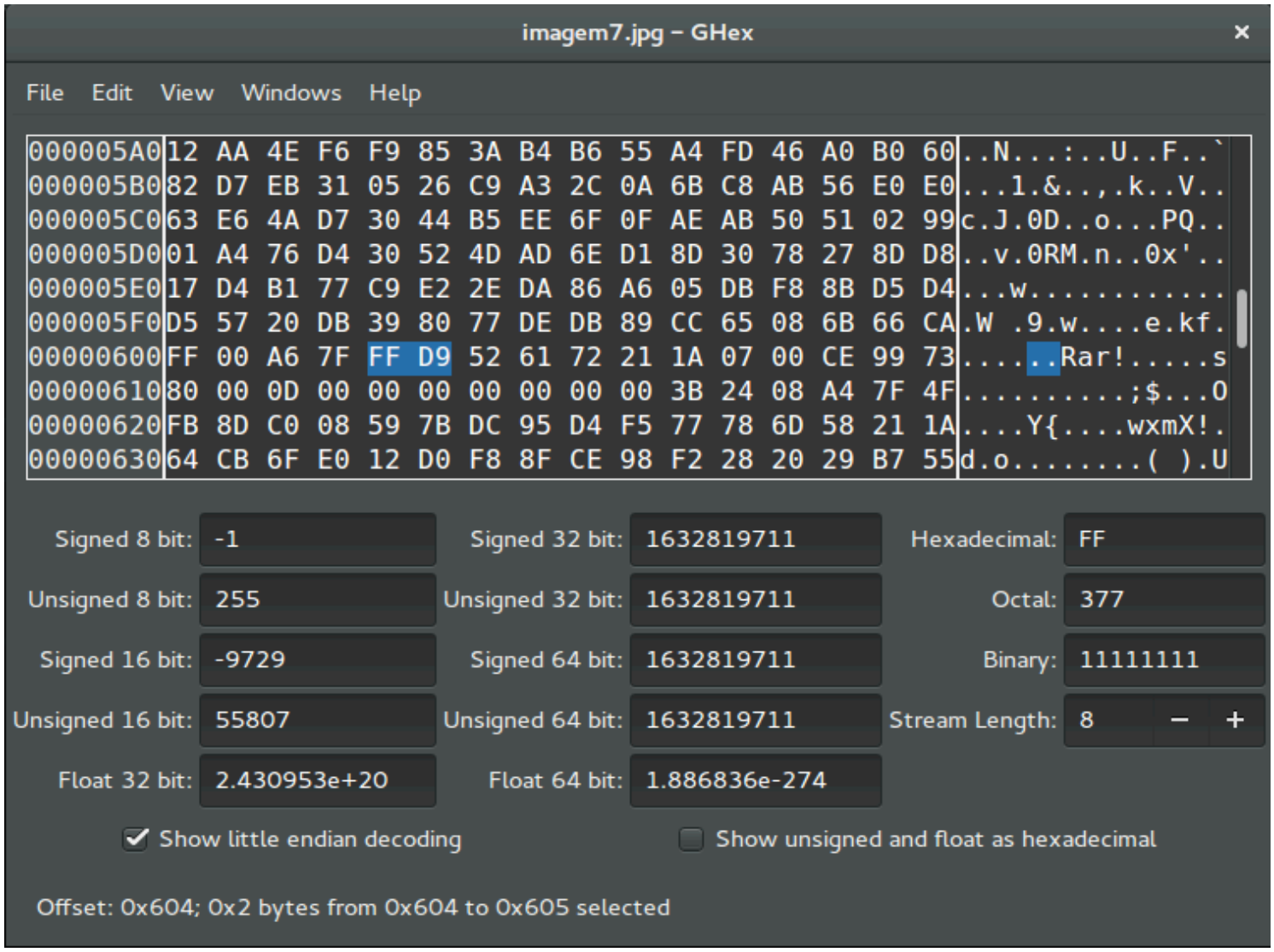
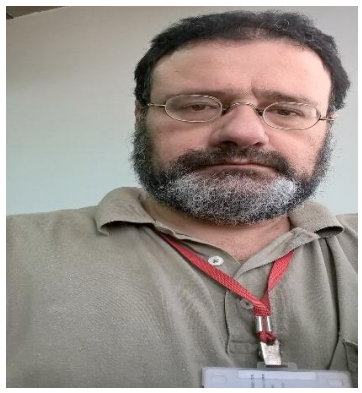


Figure 6: Getting the RAR signature of the file

Preliminary Conclusion

At this stage of the penetration testing project, it was found that even with a protective layer on the server, provided by the fact that the file was at root level and hidden behind an image, attackers manipulated this line of defense with exploration brute force attacking the ssh server. Every day, we are told about such attacks by the media. There are very recent cases in which the servers were attacked by this technique we tried to simulate, on a much smaller scale, in our server, and we have succeeded in the first stage.

Authors:



Paulo Henrique Pereira

Born in São Paulo, Brazil. He has a PhD in the area of analytical induction. Works with stochastic models and security algorithms. Researcher at the University Nove de Julho (UNINOVE) in the area of forensic and security (penetration testing). You are creating the so-called penetration testing platform Morpheus that will run on Raspberry PI.

	**Mike Almeida Garcia**	Born in São Paulo. With 23 years is Information Security student at Nove de Julho University, currently working as an intern in the technological part of the SEVEN BASE company. Junior Python programmer, has developed some own programs for Security Area information.
	**Thiago Geronimo Ferreira**	Born in São Paulo, Brazil. Graduated in Computer Science (UNINOVE) where he began to conduct research in information security area. Post-graduation in Security Data Information and auditing. Works as a network analyst, currently research on computer forensics and malware analysis.
	**Renato Basante Borbolla**	Born in São Paulo, Brazil. Degree in information security technology (UNINOVE). Works as administrator jr networks career with over 5 years in Information Technology. I taught some talks aimed at information security and currently studying on forensic and Pentest computing. Developed the server network structure and implemented all test and monitoring tools.
	**Thiago Silva**	28 years, Formed in networks by Senai. Is Information Security student at Nove de Julho University, Certificate in ITIL V3, Making sure in RHCSA. He worked for six years in Cloud Linux environment and is currently working on the implementation of high availability projects Linux environment, for large applications with a focus on safety and performance.
	**Juliano Sato**	23, Computer Technician by ETESP, studying Information Security at Uninove is systems analyst and Java developer with 3 years of experience in creating and sustaining systems for web. Since 2009, along with learning programming, came the desire to know how to manipulate, modify or bypass systems that were not created by him, was when he met the distro Backtrack 4 and entered as an amateur in the world of pen testing and hacking. Over the years investigating other people's systems, its goal today is to formalize this curiosity profession.

Approach by Vanshidhar

There is an old saying in the consulting business: "If you do not document it, it did not happen." (Read it somewhere in the library). A report, in its definition, is a statement of the results of an investigation or of any matter on which definite information is required.

A pen test report is useless without something substantial to give to an executive or a client. The target reader for the penetration testing report will vary; an executive summary for the senior management and the technical details for the IT and/or information security responsible people. In this article, I will discuss a conventional approach to develop a penetration testing report starting from collecting information, drafting the first report and ending with a professional report.

Let's start with the executive summary which will have the specific goals of the Penetration Test and the high level findings of the testing process. This section will consist of Background, Risk Ranking, General Findings, Recommendations and Strategic Roadmap.

If you do not document it, it did not happen.

Background will explain to the reader the exact purpose of the test, engagement details (terms of engagement) and testing goals. Practically, I have seen a change in objectives during the course of testing and I will mention them, as well, in the background section. This section should be written in such a way that the reader should get connected to test objectives and the relative results.

The introduction should be a narrative of the complete effectiveness of the test and the tester's ability to achieve the goals in a fixed time and include a brief description of the issues identified through the testing process and its potential impact to the business.

Now we come to the risk ranking profile where we give them grades (not the same that we had in our schools but something similar to that) on the basis of the business we are dealing with. (Ex: XSS vulnerability in a banking website and the same vulnerability in an internal college application where most people perform the job as per instruction manual and have a happy day.) Always keep in mind

“As priorities vary from person to person, similarly the risk ranking of vulnerabilities varies from business to business.”

Once we are done with the risk ranking, we move on to the general findings where we briefly represent the vulnerabilities found during the exercise. The best way I personally prefer to represent is the graphical way. The management people in most cases don’t bother much about the text written but as they normally see graphs/charts in day to day life (the company growth chart or the financial cost charts), it attracts their attention and they go through it and try to link it with the report. It is all up to the decision of the pen tester whether to include the monetary losses. I personally prefer to include them as most of the times the figures results in immediate action.

Finally, we arrive at the recommendation and plan section where we not only recommend the actions but also recommend the sequence in which the fixes should be applied. Again, the sequence will vary as per different business lines.

A pen test report is useless without something substantial to give to an executive or a client.

Once done with the tiring and boring section, as there were not many technical things to be mentioned, we arrive at our favourite part, “The Technical Section”. Here we are free to write the technical details, the aspects on which we agreed and success achieved. This section covers a detailed report on the scope, information gathered, assumptions, attack path, impacts and the remediation.

The scope section is the first and most important where we mention the grounds on which the pen test will be conducted. Practically, these are boundaries in which the pen test has to be conducted.

Now we move to the methodology section where we mention the steps followed to collect information, the methods used to analyse them, the risk rating used to calculate risk for every single vulnerability and you may also add the tools used in each step.

Now we move to the detailed finding section which is further divided into various sub parts. Here, one by one, all vulnerabilities are described in the simplest way. For every vulnerability there is a threat level, rating, analysis, impact, ease of exploitability and recommendations.

This section begins with the introduction of the person/team who conducted the test, the assets (systems on which the test was conducted) that were involved, approach methods and the final outcomes.

Starting with information gathering, it needs to be explained how the required information was gathered. This may be from passive intelligence (Google, information available publicly), active intelligence (port scanning, infrastructure assessment, etc.), corporate intelligence (mapping the business process and the physical assets identified in earlier stages) and finally personal intelligence (social engineering is the key here).

Now kicks in the vulnerability assessment section where the vulnerabilities found are described and threat classification levels are mentioned. Here we discuss the approach taken to find the vulnerability with the evidence of the vulnerability. Finally, we move to the classification section of the vulnerability where we classify them as technical (due to any misconfiguration, poor coding, poor error handling, etc.) and the logical section (application performing work that it was not supposed to do or the other way round). Finishing this section with the summary, we move to the exploitation part where we triggered the vulnerabilities to gain a specific level of access (don't forget to mention the level of access gained). Also, do mention the results where the access was gained and where it was not.

Here in the post exploitation section, we mention the impact of these vulnerabilities and the ability to take advantage of the flaw. This area should be covered with the step by step approach (I prefer screenshots to be attached for every step) and explain real world examples (explaining helps a lot). Also, mention the countermeasures that were in place and the incident response activities which were triggered during this phase.

Once done, we move to the detection capabilities of the vulnerabilities and explain how easy or how difficult it was to detect them in the system. There are certain vulnerabilities that show up only after a certain level of exploitation takes place, while some show up in the initial phase itself. If there are linked vulnerabilities, then always mention that Vulnerability A led to the finding of Vulnerability B and so on.

In the risk section, I personally prefer to give the direct impact the business may experience from the existing vulnerabilities, supporting them from my findings in the previous section. Always keep in mind that a risk which is high for one business may be low for another so be very careful while mentioning this in the report. Also, advise them about the frequency, the present controls and the skills required to exploit the vulnerability.

Last is the summary section where we mention the complete test in a nut shell and support the growth of our client security posture. Provide guidance and support and end it with a positive note.

Sample Format of a Pen Test Report

EXECUTIVE SUMMARY

SCOPE OF WORK

OBJECTIVES

ASSUMPTION

Author: Vanshidhar

Based in Pune, India; Vanshidhar has a keen intrest in information security and has been following these since the college days.

Started his career as Java Developer and moved to information security. He holds a bachelors degree in Electronics and Communication Engineering from Rajiv Gandhi Prodyogiki Vishwavidyalaya, Bhopal.

Approach by Junior Carreiro

Pentest reports are boring, but necessary

The least exciting part of a pentest certainly is to write the report.

If the pentest was performed in group, there is still the possibility of a draw to see who will create it.

The idea of this article is to help make this process more agile and less annoying.

Among the phases of a pentest, we can say that the report is the most important because it is the product that will be delivered to the client and the client side will not always have people with technical knowledge. For this reason, writing a good report, well written and providing evidence, is of maximum importance.

Various models and sites explaining how to make a good report can be found on the internet.

Using this base of information, I created a line to be followed when writing the report.

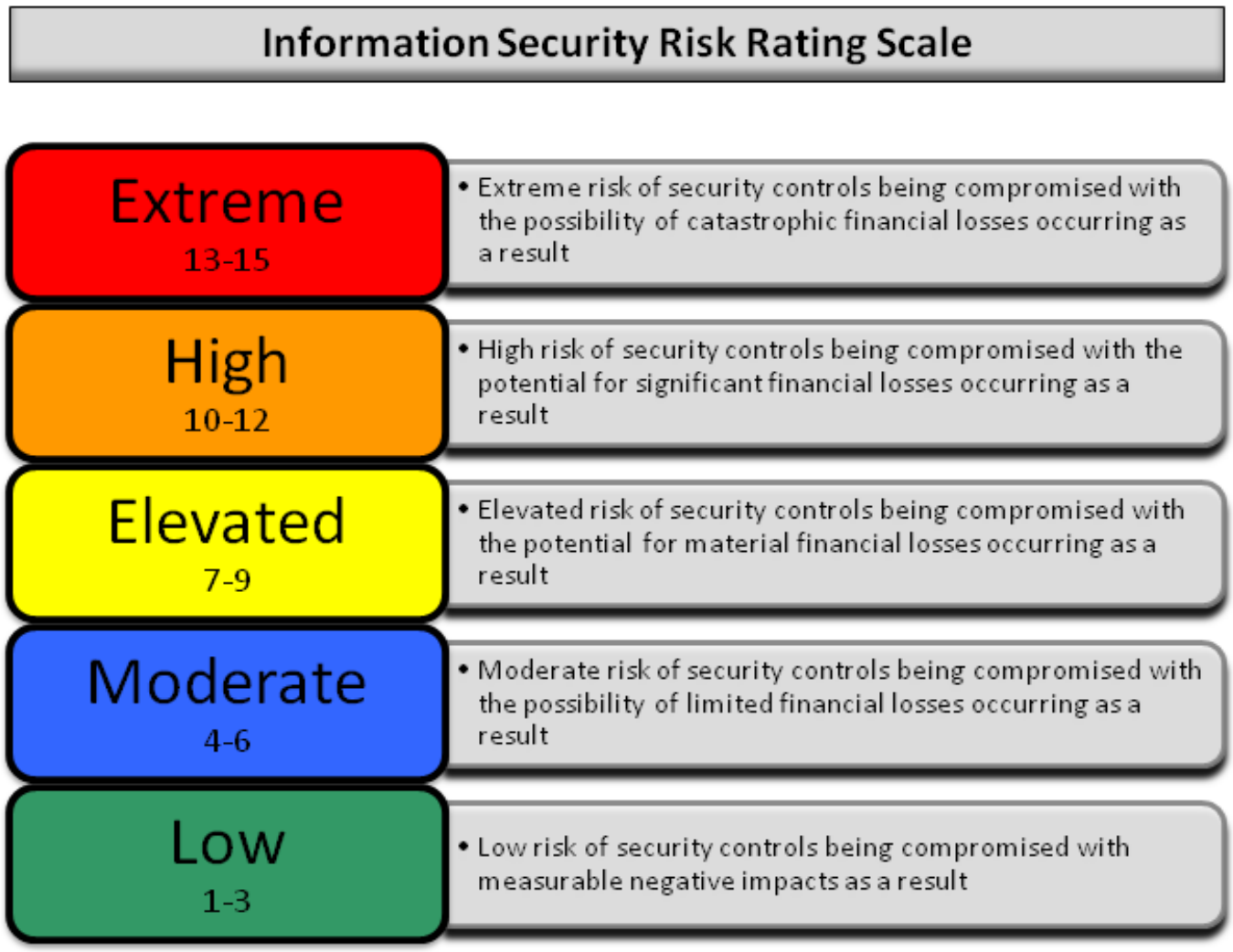
My idea is to provide some tips so that you can create your own tracks or follow mine and not have to create a step by step for each report you create.

Divide the actions

To facilitate the writing of the report, we can divide it into three phases:

Planning - Writing the draft - Review

Planning - The idea in the planning phase is to keep in mind all the tests that will be carried out and how they will be completed. To do this, create a checklist with the tasks; this will help you stay focused and not be distracted. Set scores for ease in measuring the risks and make clear to the customer the level of criticality of each breach found. Do not forget to put possible problems you may encounter and how to circumvent them. The picture below shows an example of the scores.



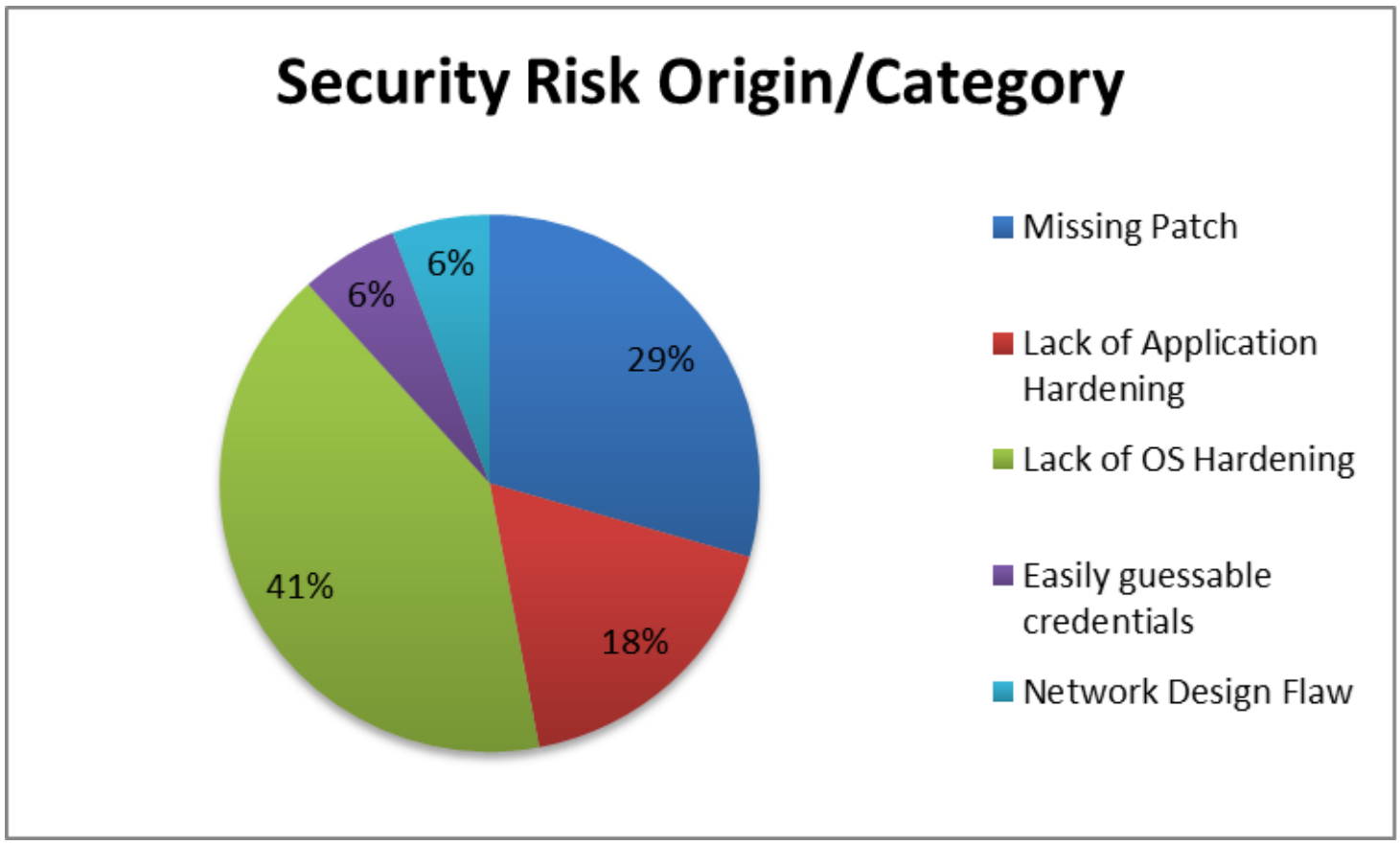
Draft - The idea of the draft is to create a file in Notepad and paste commands executed and their outputs. Take PRINT SCREENs when it's not possible to copy and save in a folder. Note here also, the found errors and difficulties and what was done to correct them. At the end of the pentest, you will see that the draft is already more than 60% of the report. And don't forget to highlight the most important information.

Review – Testing is finished and it's time to review everything that was noted in the draft and save. Carefully review all information and remove what is unnecessary so that the report does not include any junk. If the test was conducted by a team, ask everyone to review; if you have done it alone, ask a friend; the most important thing is to have a second opinion and be sure that it's all OK.

Conclusion

The report is as important as the pentest, so it is very important that it is very well written and detailed.

It will probably be read by people with technical knowledge and also people without that knowledge, so it needs to be written in a way that both types of people understand. Remember to make graphic representations of the targets tested, testing results, processes and other things that you think are important in the final outcome. That can help in the understanding of the end result. Remember that this is the final result that the directors and people of non-technical areas are likely to read. The picture below shows an example graph:



The pentest report is an important document and has standards to be followed. It should have a Cover Page, Table of Contents, Executive Summary, Scope of work, Project Objectives and other topics. I will include some links to examples of reports and websites with step by step explanations.

Author: Junior Carreiro

I have been working for about ten years with open source software, always redirecting my studies for the security area and security baselines. I try spend my free time with research and study issues facing security , security web application area and study English. When I'm not studying, I like to stay together with my wife or go to the mall, go to skateboard or play video game.

Member of DC-Labs Security Team
Member of Área31 Hackerspace
Staff on Bhack conference

Links and references:

https://www.offensive-security.com/reports/sample-penetration-testing-report.pdf

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

https://www.sans.org/reading-room/whitepapers/bestprac/writing-penetration-testing-report-33343

https://pentestmag.com/course/writing-an-effective-penetration-testing-report-w9-2/

https://darrylmacleod.wordpress.com/2012/03/26/penetration-testing-report-templates/

Approach by Eric Schultz

The database is dumped. The file server is served. The domain controller is controlled -- by you. It's official. The domain is yours. The euphoric high climaxes with a round of figurative high fives among the team for a job well done. Chairs lean back as the first breaths of relaxation enter lungs that have previously been fueled by caffeine and adrenaline. As the testing wraps up, other testers quickly get siphoned off to new tasks and you find yourself the sole survivor left to generate the report.

The euphoric high fully evaporates as you, a master of technology, realize you have to use one of the most unsophisticated tools of the trade and the favorite program of secretaries everywhere: the word processor. Although it's low on the technical totem pole, report writing is just as important as discovering findings. Knowing a few tips can make the process a lot easier and quicker so you can get back to testing.

Who Gets the Report?

One of the sad truths of penetration testing is that discovering more findings means more report writing, and not just any writing: complex, technical writing. Too few words, too many words, or choosing the wrong word can cause confusion and misinterpretation.

Reports with important findings can quickly be distributed to a hundred people, each with a differing technical understanding. In contrast, some reports only get read by a handful of developers. Knowing the technical proficiency of the readers can allow you to focus on the right areas.

-If you know the report is going to reach upper level management, more time needs to be spent on explaining the impact with simple terminology. Focus on what's impacted and what an attacker can do if the vulnerability is exploited.

-If the report is going to developers with with no security experience, definitions of security concepts should be included. Third party resources and relevant corporate polices should also be included.

-If the report is going to a team that regularly gets your reports, you can focus mainly on the technical details of the findings. Extra information, like basic definitions, should be avoided, if possible.

Know When to Keep It Brief

Penetration testing is from a black box, or gray box, perspective, so it's almost always impossible to know all the details of a system's architecture. If a report's recommendation section is too specific, it can cause developers to implement a solution that won't work in their environment. Now you've got an angry email in your inbox asking why the solution you provided didn't remediate the finding.

Know When to Lay on the Details

A week after you submit the report, you get an email that ruins your day. The developer refutes your claim that the server is vulnerable. You roll your eyes at their ignorance. Obviously, the server was vulnerable. Quickly, you check your notes, and realize you don't have the screenshots to prove the system was vulnerable. Major bummer!

You try to go back and get the information, but you no longer have access. Developers have applied fixes. Vulnerable parameters are no longer vulnerable. The software that never updated and had the vulnerability for the past four years has suddenly been updated. What should be an easy rebuttal can damage your credibility.

While this sounds far fetched, it does happen. Each finding should have:

1. Enough information that another tester can reproduce the finding.

2. Screenshots documenting the vulnerability.

3. Any factors that influence the risk rating (PII exposed, other vulnerabilities, firewalls, etc.)

4. Locations of vulnerable areas and parameters. If your screenshots show URLs, be a hero and provide a text version to make the developers' lives easier. No one wants to spend 10 minutes copying a URL from a tiny screenshot.

When Should You Not Write A Report?

Trick question. Anytime you're tasked with assessing the security of a system or application, you should write a report. Sometimes, what you're testing just isn't vulnerable and writing a report for an assessment with zero findings may seem useless, but the report is useful to the project owner. If the system ever gets audited, they can use the basic report to show their system is healthy and that it gets regular security assessments.

A basic report template stating that no findings were discovered should be used that lets the report writer quickly change the tested system's name, scope and dates. Create a template with placeholders for those variables (e.g. <APP NAME>) and then use the word processor's Replace All function (ctrl+h is the common short cut) to change the generic name with the actual value.

Time is Everything

Great report writing takes time -- time that may not always be available. It's not uncommon to get assigned a project scheduled to enter production in a day or two, or worse, sometime last week. All of the urgency to get the project into production is directed at the last hurdle, your security assessment. Feeling the pressure can result in mistakes or incomplete tests. In a small industry like security, having systems you've tested end up being exploited hurts your reputation.

The best way to alleviate the pressure is to tell the point of contact that you'll reach out immediately if you discover any show stoppers. In this case, an initial report can be key. It should be stripped down and informal. Only a handful of facts matter: your details (name, phone number, email address), the URL or IP addresses looked at, the finding title, finding description, finding location, initial estimated risk rating, and a brief one or two line recommendation. The goal is to notify the system owner and get back to testing as soon as possible.

How to Speed Up Report Writing

Use a template. As stated earlier, a report template can save a lot of time. It should have a title page, table of contents, and sections for the executive summary, scope, and a findings section. If the environment has frequent issues that appear in most reports, add them to the default template to save time.

Keep it simple. When using a template, it's easy to get carried away with the design and tweak font colors, sizing and spacing between sections. The more changes the template gets, the more effort is needed to make sure reports adhere to those changes.

Use boilerplate findings. Writing findings from scratch takes time and energy. Modifying findings from other reports takes time to search multiple reports. Instead, testers should get in the habit of maintaining a series of word documents that contain template findings that occur frequently and can be quickly modified to fit the current assessment.

Email templates. It's 4:58 PM on a Friday. Two minutes till you can officially start your weekend. As your coworkers disappear like alien abductions are the latest fad, you've just put the final fixes on your report and have to send it out before you can leave. Thankfully, you created an email template that has all the usual verbiage, and email recipients. Everything that changes in the email body (number of findings, URLs/hostnames, etc.) is conveniently colored in red. You quickly update the email body and subject, attach the report, and hit send before the clock hits 5.

Author: Eric Schultz

Eric Schultz is a senior security researcher with experience working on red teams, pen testing, web and mobile security assessments and source code reviewing for the United States Federal Government and Global Fortune 500 sector. In his spare time, Eric enjoys reading and writing about various security topics. Sometimes he even reads corporate security policies for fun. He also enjoys mentoring, developing, tinkering and avoiding use of the word cyber. When he isn't on the computer, Eric enjoys shopping at Walmart with his girlfriend Amber, battling chickens for eggs (its just one rooster, but he's sneaky), eating sushi at the local mall, and thinking of things to do once he returns to his computer.

Approach by Juan Pablo Quiñe

One of the first problems is to get the budget to start an evaluation. Some arguments to get budget to make an evaluation could be: regulation, audit, enforcement, risk management, a fusion with some corporate, to know where we are in security, a suspected fraud, the need of the new CSO/CISO, or who knows, maybe a business requirement.

First, let's share common language...

We are going to talk about Ethical Hacking as the activity of assessment that is made by an organization, team or individual, that is authorized to identify possible security holes in the IT infrastructure. An Ethical Hacker will try to bypass the security mechanisms with the objective of finding security breaches before an attacker.

Now let's talk about decisions...

Every time a person or company decides to execute an Ethical Hacking evaluation, a bunch of questions start in his head (or must have) in order to receive the most value for his investment.

Traditionally, some of the questions we start thinking are:

When to complete an Ethical Hacking evaluation?

What might be the scope?

What types of evaluation exist?

What is the profile an Ethical Hacker needs?

What methodologies should I consider?

Let's try to talk about each of them before talking about the report.

When to complete an Ethical Hacking evaluation?

Some people would say that it is part of a preventive security strategy, or to know the actual state of the security, and should answer the questions: Are we secure? How secure?

Others will say that this could be a part of a Detective Strategy, as a part of the operations, the SDLC or as compliance.

Other people will consider that it might be part of a reactive strategy, in case of need. In my humble opinion (IMHO), I believe it will depend on the budget, the time, and the speed of the organization, and these are some of the variables I will consider.

At the end, its all about expectations, and perception.

Now about the scope. Typically, “the best practices” consider the scope from the point of evaluation. It means that if the need is to get a perspective of the vulnerabilities from the Internet, it will be an external Ethical Hacking. If I want to know how an insider could get in to my network, then it will be considered an internal Ethical Hacking, and the same will happen with assessment options, like web, mobile, source code, physical security, SCADA, and so on.

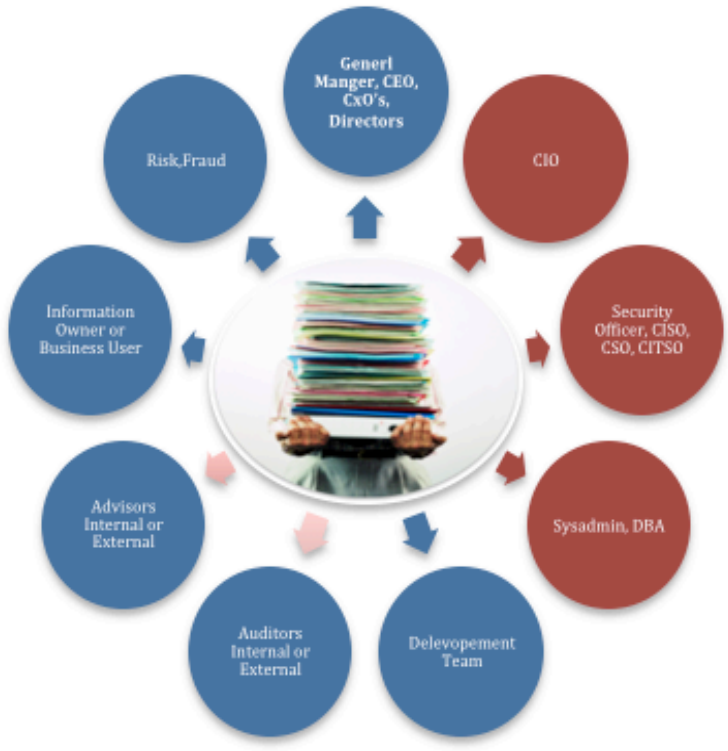
When we consider the type of evaluation, it will depend on the information you give to the evaluator, a black box, gray box or white box, and some other considerations like double black, or double gray, where you do not tell the monitoring team about the evaluation to see how they react.

Other considerations are related to the evaluator: the certifications, the “ethic”, the reputation, the experience, and the methodology that will be used.

What is missing then? (the problem)

Till now, I haven’t said much about the decisions and considerations; it’s more of the same. However, let’s consider this: in the end, you pay for a bunch of paper, you pay for the report. So no matter how good your pentester was, or the methodology used, or the many considerations, you are receiving a document, and will share it with many persons.

But who are they? Let me share my perspective in the following picture:



These people are the ones who probably will have access to the report. Some appear in blue and others in red. The red ones are probably who will make the request, and make all the previous decisions. Now I ask myself, how will the report meet the expectations of people that were involved in the requirement process?

You can add to this picture that the pentester may want to include information proving that he is THE expert, and that may not be what the people who read the report are expecting.

I found this situation to be so common in the field that I decided to analyze this. First, I started to document what they probably search for in the report, and what is in their agenda and found the following:

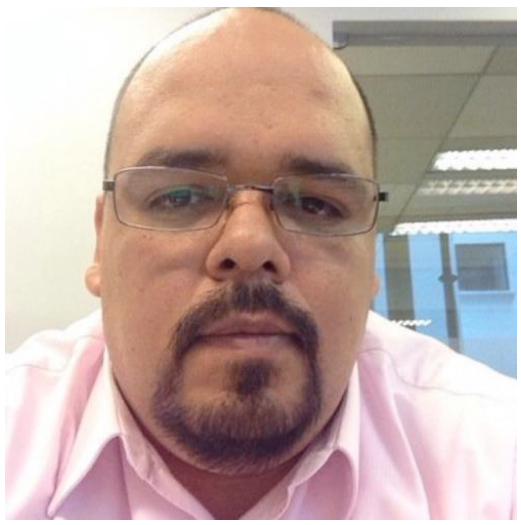
Who	What they search	What is in their agenda
General Manger, CEO, CxOs, Directors	Strategic Information	Profitability
CIO	Information that supports their decisions	Technological Support
Security Officer, CISO, CSO, CITSO	(In)Security Information	C.I.A. (Confidenciality, Integrity, Availability)
Sysadmin, DBA	Actions to execute	Availability
Development Team	Understanding the problem, not the vulnerability	Operability
Auditors Internal/External	The finding for follow up	Compliance

With the traditional decisions it’s almost impossible to please everyone with the evaluation. Each section of the report is normally applicable to a person, or a group of persons. It's common to have an “executive report” and a “technical report”, but this is not enough. We need to align the business needs with the security initiatives and this starts in the requirements, in the definition, in the expectations. If we involve all the people that will receive the document, and hear their needs, or what they are expecting, what will happen is that we could include their needs in the requirements, or explain to them that they must lower their expectations about the work. Always try to make it be a positive situation.

We need to make the executioner change their focus, so the result will give value to the business and make all the stakeholders involved in the process.

To do so, it’s important that the “sponsor” and the “executioner” have considerations before starting the evaluation, making clear the objectives and the expectations; during the evaluation, understanding the client, making sure that the objectives will be met, and after the evaluation, validating the results, learning from the experience, and helping all the stakeholders understand the findings and their implications.

At the end, its all about expectations, and perception.



Author: Juan Pablo Quiñe Paz

Information and IT Security Strategist with more than 15 years of experience in the field. Juan Pablo has worked for some renown Companies like Ernst & Young, IBM, Hewlett Packard always in the IT and Information Security Field. He has worked as an Specialist, Auditor, Consultant, Advisor, and he has also worked as Chief Security Officer for Perú’s Health Insurance Organization (Essalud).

CISSP, OSCP, OSWP, CPTE, ITIL