

How to Write a Killer Pentest Report

You may have l33t skillz on the command line, but can you communicate through the written word?

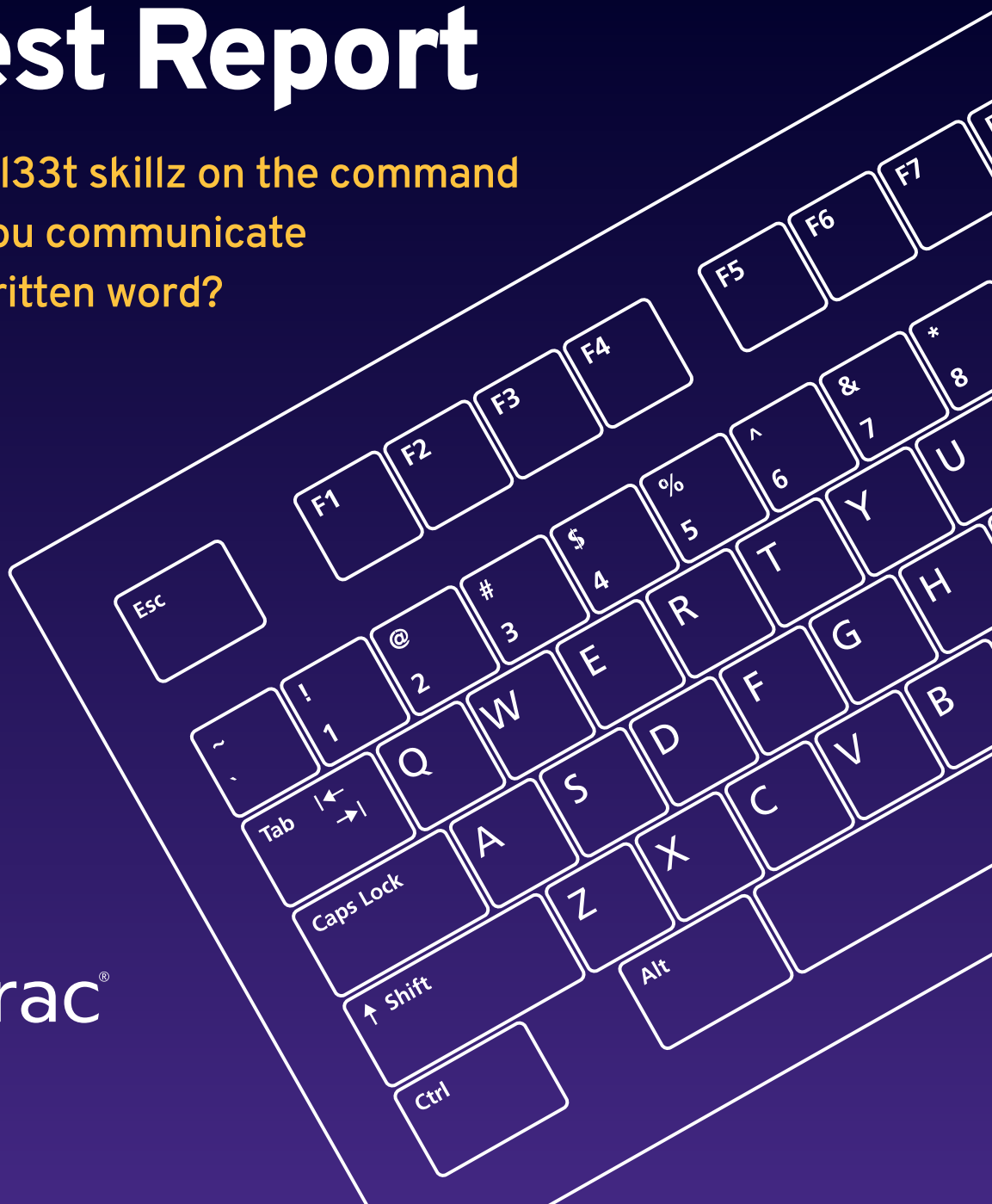


Table of Contents

- Introduction2**
- General Tenets of Report Writing.3**
 - Educate Your Audience. 3
 - Document Your Methodologies 3
 - Define and Respect the Scope 4
 - Guard Your Credibility 4
 - Remain Objective and Courteous at All Times 5
 - Acknowledge Areas of Strength 5
- Effective Report Structure.6**
 - Cover Page. 6
 - Table of Contents 7
 - Executive Summary 8
 - Nut and Bolts. 9
 - Scope 10
 - Methodology 11
 - Threat Model. 12
 - Rules of Engagement 13
 - Attack Narrative 14
 - Summary of Findings 15
 - Detailed Findings 16
 - Description 17
 - Vulnerability Overview 17
 - Vulnerability Education. 18
 - Vulnerability Technical Specifics 18
 - Score 19
 - Affected Assets 20
 - Recommendations 20
 - References. 21
 - Conclusions and Future Recommendations 21
 - Appendices 22
- This All Seems Like a Lot of Work24**
 - Final Thoughts 24

Introduction

You crafted an irresistible spear phishing email, which provided initial access to an unprivileged account. You rapidly gained persistence on your beachhead host, then escalated privileges through application shimming. You fired up a little Kerberoasting and grabbed the creds you needed to move laterally, picking your way through the network until you landed on the crown jewels. You dropped a calling card, erased your tracks, and then popped a cold beverage. Easy days' work — time to get paid ... but the most important part is still ahead of you.

Hacking is a blast, but many (okay, most) pentesters loathe writing the report. But like it or not, the report is why you were hired. It is the single document upon which you will be judged by your clients and indirectly by your future clients. Your ability to author an effective report is just as important as your hacking skillz when it comes to your bottom line. Yet very few pentesters spend even a fraction of the time honing their report writing skills as they spend learning and practicing new tactics.

At PlexTrac, we know a thing or two about reporting, both from experience in our roles as practitioners and from the extensive work we've done with our customers. We've seen a lot of reports in various templates and formats. We've also helped numerous customers convert their report templates into a PlexTrac compatible format. As a part of that work, we've seen a lot of great report formats, and we've also had the opportunity to provide recommendations to customers on areas for improvement in their templates. Based on that knowledge and expertise, we will highlight some of the good and bad that we've seen when it comes to reporting. This paper will begin by discussing a few tenets that should form the foundation for an amazing penetration test report. We will then discuss report layout, with an in-depth discussion of how content can be effectively presented to provide maximum knowledge transfer.



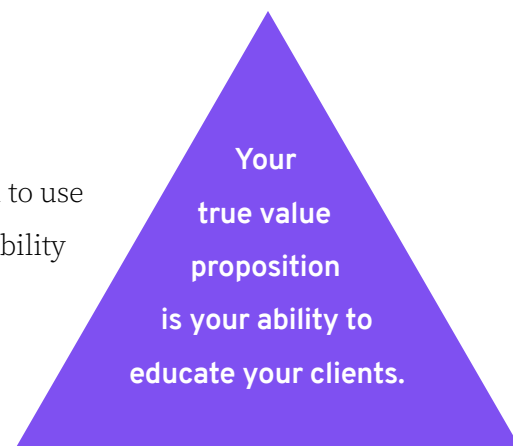
General Tenets of Report Writing

It's best to plan your report strategy before you ever start writing. Thinking in advance about the purpose, context, and audience of the penetration test results and the report you will write about will save you time and energy later.

Educate Your Audience

Your client isn't paying you to show off your l33tness — they are paying you to use your skills to identify risks and vulnerabilities. But as important as vulnerability identification is, your true value proposition is your ability to educate your clients. They not only need to know how to fix the issues you discovered, but also how to prevent them from popping up in the future.

Furthermore, your client isn't a person — it is an organization with many stakeholders of various degrees of technical competency. Your challenge (and duty) is to effectively educate all stakeholders at their level. And since you are only going to deliver one report, you need to communicate at multiple levels of technical competency within the same document.



Document Your Methodologies

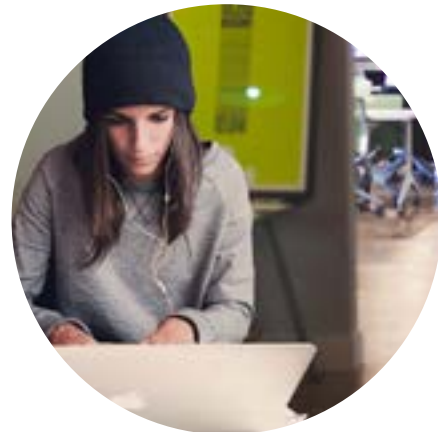
You may have breezed through your OSCP with time to make a pitcher of margaritas, but professional testing is about process — not wizardry. Though penetration testing is a relatively new discipline, the collective wisdom of the community has coalesced around standardized methodologies for performing testing. These methodologies help testers perform a comprehensive analysis of the environment instead of simply walking through the first open door they find.

There are numerous penetration methodologies available (search “penetration testing methodologies” at <https://www.owasp.org/index>), and some are surely more appropriate than others for your clients based on their industry vertical, regulatory requirements, maturity, etc. Documentation of your chosen methodology should include not only which tool you used, but why that methodology was the appropriate framework for this particular test.

Your methodology should not exist as a lonely paragraph isolated in the executive summary; it should be woven through your report. Each finding should have a direct and documented link to the methodology. This provides validation of the significance of the finding and reinforces the perception that you performed a methodical investigation.

Define and Respect the Scope

We live in the era of disappearing perimeters, fueled by the rapid adoption of EaaS (Everything-as-a-Service) in the cloud. You almost certainly were not hired to “boil the ocean,” or examine every aspect of every information system that your client uses. Thus your statement of work (SOW) should meticulously define which systems, applications, or third party services are in scope as well as anything that is specifically off-limits. However the SOW will be seen or read by far fewer people than the report, so it is vital for your report to reiterate the scope.



Each of your findings should directly point to an affected asset or location that is included within the scope. You may very well discover areas of concern outside of your scope — but resist the temptation to shoe-horn these findings into the main report. As a professional courtesy, you may consider including out-of-scope discoveries in a clearly-marked appendix or as a separate communication.

Guard Your Credibility

It is your responsibility to highlight gaps in your clients’ defenses. There may be many reasons why a gap exists, but it is inevitable that at some point a stakeholder will become defensive about a finding. This defensive reaction may often include an attempt to discount the validity of your work. If they are successful in raising doubt about a single finding, the credibility of your entire project is at risk.

You can mitigate the potential for this situation through meticulous documentation of your efforts and findings. It is not enough to simply throw artifacts into a report; those artifacts must be given the context necessary to support the narrative of your finding. Any assertions of vulnerability should be backed by reference to industry-recognized standards such as Common Weakness Enumerations (CWEs) or Common Vulnerabilities and Exposures (CVEs).

You can further enhance the credibility of a finding by searching out and acknowledging any mitigating controls that are already in place. In doing so, you elevate yourself from someone who simply catalogs vulnerabilities to a professional who helps the client identify true risk. And you take away an easy line of argument from any potentially defensive stakeholders.

Remain Objective and Courteous at All Times

Your report should always stick to the objective facts of what you found, avoiding any judgment of the people or processes that support the environment. Professionals build people up, not tear them down. You may be calling someone's baby ugly, but if you do it tactfully and respectfully, most clients will appreciate the work that you do and seek your counsel on how to do better moving forward. Exercising tact and respect is how you build positive relationships that will not only improve your clients' security posture, but will also likely land you additional contracts in the future.

Acknowledge Areas of Strength

Existing controls that frustrate your efforts are a good thing. Acknowledging the effectiveness of controls does not diminish your value as a tester — rather it reinforces behaviors that you want your clients to continue. If that expensive next-generation firewall (NGFW) thwarted an attack vector, the client should be aware so that they see the return on investment (ROI). A pentest is not a capture-the-flag contest; your role is to provide an assessment of the effectiveness of the clients' defenses. You won't make your client angry by giving them a thumbs-up where they are doing well.

But enough with the philosophical musings...

Let's get to the brass-tacks of report writing.

A pentest is not a capture-the-flag contest; your role is to provide an assessment of the effectiveness of the clients' defenses.

Effective Report Structure

There is no single correct way to structure a report, but there are many ways to structure one poorly. Remember that the primary purpose of the report is to educate, and the audience is likely diverse in their technical competencies.

The structure of a report supports the goal of education by

- Enabling consumers to rapidly find the information they need
- Delivering the information with the proper context and at the technical level of the audience
- Presenting similar data types in a consistent fashion
- Empowering readers to mitigate their risk with actionable next steps

We will discuss one method at a high level for structuring a report with these objectives in mind.

There can and should be some differences based on the type of testing. For example, a web application penetration test will have different subsections than an internal network pentest.

However the overall structure we present here is malleable to almost any circumstances. In general, the basic principles of effective communication still apply: Tell the audience what you're going to tell them, tell them what you want to tell them, and, finally, tell them what you told them.

Cover Page

Don't discount the value of the cover page — it is the first thing your client will see and is a representation of your brand. It should be professional yet distinctive; invest the time in creating a design that is congruent with your company's marketing schema. The color palette for any borders (should you use them) should be consistent with your logo. Customizing the cover page with your client's logo is an option, but be aware that this may result in color palette clashes with your existing design and your company's logo. In short, be thoughtful about visual design and enlist the help of your marketing department, if possible.

There is value in simplicity; complex graphics or busy images can crowd the space and distract from the relevant information displayed. Ideally, you will produce more than one report for any

client over time. Thus you want certain discriminators for this report to stand out, to include the following:

- Type of report (Web Application Security Assessment, Network Penetration Test, etc.)
- If for an application, include application name and version, if applicable
- If for an enclave in a larger organization, include a discriminator (e.g. HQ LAN)
- Date of report completion
- Primary point of contact for your team, to include email and phone number

Table of Contents

The table of contents is vitally important to your report's navigability. Modern word processors make generation much less painful than in days past, automatically updating pagination with a few simple clicks — if you use the feature (which you should!)

Automated updating relies upon the use of Styles in the word processing software. If you aren't familiar, now is the time to learn. A style is simply a collection of attributes for text, such as font, size, emphasis, spacing, color and alignment. By using Styles for each header and sub-header, you ensure uniformity throughout your report — and avoid a lot of unnecessary manual editing.

Styles are also vital to the automated updating of your table of contents (TOC). Generally, your TOC will only display those headers that are formatted with the Header 1, Header 2 or Header 3 Styles. You can go deeper, but that's generally a bad idea. If you have content that is so important that you think someone may wish to navigate to it directly, it shouldn't be buried inside the bowels of other sections.

When using Microsoft Word, we find it helpful to keep the Styles pane open. This provides instant visibility on what style you are currently using, and allows you to change the style you are using with a simple click.

Finally, the font family you use in the TOC should be consistent with the rest of your document.



Executive Summary

Your executive summary should begin with a brief introduction to the overall report. The introduction is the first impression your report will make to the client. The introduction should be concise and brief, highlighting what information the report will contain. It is the section of the report where you tell the client what you're going to tell them. The introduction sets the stage for the rest of the report.

We see two common mistakes in executive summaries. The most common is the “kitchen sink” error. You know that the executive summary may be the only thing a stakeholder reads, and thus you cram in so much information that it becomes 10 pages long. The second visits the other extreme: The content is limited to logistical information and perhaps a tally of the number of vulnerabilities discovered.

An executive summary should stand alone from the content it summarizes, providing a high-level overview of all the key concepts included in the main body of the report. Conversely, it should include nothing that is not discussed in greater detail elsewhere. Even though the executive summary is placed first in the report sequencing, it should be the last thing you author. This order ensures that you don't forget key points or concepts in the main body due to a perception that they have already been covered.

There are a few items that can be included in the executive summary and not referenced within the remainder of the report. These include brief logistics information (e.g. period of assessment, number of testers, location, etc.) and the engagement objectives. The objectives are different from the scope and ideally can be pulled directly from your SOW. Shared objectives are vital to ensuring that your actions meet your client's intent. “Perform a pentest” is not an objective. Objectives should be measurable, obtainable, and provide a clear vision of the desired end result of the engagement. For example:

- Identify the presence of any vulnerabilities in the application's front end which can be exploited to provide unauthorized access to data stored in backend databases.
- Identify publicly exposed ports and services which have documented vulnerabilities that can be exploited to bypass existing perimeter defenses.

Some testers include a summary of findings table that includes the title and severity of each finding. Such a table is important and should have a home in the report (and we do discuss this elsewhere); however, including a summary of findings table in the executive summary may become unwieldy on larger reports with numerous findings. An alternative approach is to only include a summary of critical findings. But remember your audience! Non-technical people are likely to read this. What is a critical finding, and what are the implications? You don't need to pen a paragraph, but a line or two of education can set the proper context. For example:

“PenTest Secure discovered 56 vulnerabilities in the assigned environment, of which four are considered critical. ACME corporation is strongly encouraged to remediate these critical findings within 15 days due to their high potential for exploit, which could result in a significant and immediate monetary impact.”

You may also choose to include some form of graphic that provides a tabulation of the total findings discovered by severity. This may be visually pleasing and add some interest to your report, but the value is debatable as a stand alone artifact. Naturally, the executive summary is what you plan to provide to an executive stakeholder, so it's okay to include additional graphs and metrics, but always make sure these artifacts accurately depict the message you're trying to convey and help answer questions rather than create new ones.



Nut and Bolts

The next section of your report is a superset of all the modules that provide the context for how you arrived at your findings. We will use the term “Nuts and Bolts” to describe this grouping, but that's not a term that you should use as a section header! Each module should include its own header to ensure proper inclusion in the table of contents. At a minimum, these modules should include:

- Scope
- Methodology

Depending on the engagement type, desired level of effort, and sophistication of your testing team, you may wish to include modules such as:

- Threat Model
- Rules of Engagement
- Attack Narrative
- Summary of Recommendations

Each of these sections are discussed separately below. Note that you may have additional or different sections in your report, which is okay. We have just seen these sections often and find them to be valuable in achieving the objective of a high quality report.

Scope

A properly written scope section can prevent misunderstandings between you and your client on what was expected versus what you delivered. As a pentester, time is money — and the scope dictates how much time you will devote to a project.

It is unlikely that most clients will understand the level of effort necessary to meet the objectives they define. Therefore, it is your responsibility to accurately and honestly define that level of effort through the scoping process. This includes working with your client to determine what activities will provide the greatest return on investment within their budget.

This section should largely reiterate your statement of work, as the scope of an engagement is an integral part of your pricing model. The scope needs to be clear and unambiguous; this is the time to discuss specific enclaves, subnets, and perhaps even hosts. Tests against applications should include the version number as well as the hosting environment, especially if the application inherits controls that are unique to that environment.

Your scope section may also include limitations on your level of effort in terms of hours committed to the engagement or section of the engagement. For example, efforts to gain initial access through phishing may be scoped at two mass campaigns and a single unique spear-phishing attempt crafted for nine executives. You may include the number of hours devoted to particular phases of the attack life cycle.

If you are contracted to perform multiple “flavors” of pentest (e.g. Physical, Web App, External, Internal, etc), you should include separate subsections that address the assets and level of effort respective to each flavor.

To learn more about the scoping process, a comprehensive resource is available at http://www.pentest-standard.org/index.php/Pre-engagement#Introduction_to_Scope

Methodology

The methodology section highlights the general progression of activities you conducted to complete the assessment. You may follow an industry standard methodology or follow your custom (and documented) testing process. Regardless, the goal of this section is not to make the readers experts on your chosen methodology but rather to educate them so that they can answer these questions:

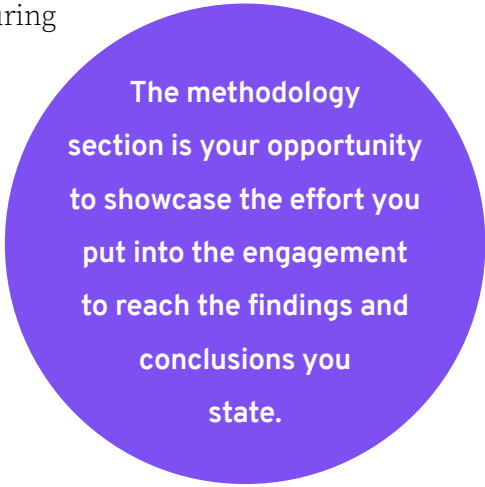
- Why was this methodology chosen?
- Is it considered an industry standard for my vertical?
- Do I have regulatory requirements that align with this methodology?
- Is it appropriate for the scope of this engagement?
- How was this methodology tailored for this engagement?
- Were aspects more heavily weighted than others to meet our objectives?
- Were aspects omitted due to scope?
- Were techniques and tactics aligned with our projected adversaries?

If you initiated the test with a level of access provided, a brief discussion of why this level of access was given may be warranted. For example:

“We began the engagement with unprivileged access to a single host in the marketing department. Over the past year, three incidents occurred within ACME corp in which initial network entry was achieved via phishing into this enclave. Repeating the actions necessary to achieve this initial access was deemed unnecessary and outside the scope of this engagement.”



The methodology section is your opportunity to showcase the effort you put into the engagement to reach the findings and conclusions you state. Take time to walk through each phase of your methodology and highlight what activities were conducted during that phase. This is particularly valuable when you find yourself having devoted a significant amount of time to the engagement but may not have found many issues to report. That's great for your client but can make you feel self-conscious that they paid you a fair sum and have so few findings to show for your hard work. The methodology section is your opportunity to showcase the level of effort that went into the engagement.



The methodology section is your opportunity to showcase the effort you put into the engagement to reach the findings and conclusions you state.

Threat Model

Threat modeling seeks to identify the profile of the attacker that may target your client, to include the most likely vectors and which assets are most coveted. This process enables the pentesting team to choose tactics, techniques, and procedures that are representative of the adversary profile.

If you choose to engage in threat modeling and include it in your report, this section should answer the following questions from the perspective of the adversary:

- What are the high-value assets that are most enticing?
- What methods can I use to bypass controls?
- How much effort will an attacker put into obtaining access? (i.e. How dedicated are they?)

The answers to these questions may vary widely based on whether the profile is that of a targeted or opportunistic attacker; therefore, you may wish to differentiate between these profiles in your report. In either case, you should include a discussion of how your threat model impacted your choice of methods and tools for the test.

Distinct methodologies have been developed for threat modeling (e.g. STRIDE, P.A.S.T.A., OCTAVE) and executing the formal process can easily grow to become a distinct project (or at least be a line item in your statement of work). If you use a formal threat modeling

methodology, you should identify which one you used. But don't discount the value of more informal threat modeling that can be accomplished in consultation with your client by asking

- What type of incidents have you experienced in the past?
- Do you have direct competitors in nations with environments that are permissive of cyber crime?
- What kinds of attacks are your competitors or others in your vertical experiencing?

Include whatever information you gather as the basis for your model in this section. Ultimately, the threat model is your justification for why you chose to focus on specific areas throughout the pentest. As everyone knows, you have a limited amount of time to conduct a pentest, and the threat model can help dictate where to focus your efforts while justifying those choices to your clients.

Rules of Engagement

Your scope identifies the assets that you will assess, but it doesn't necessarily cover what actions you are permitted to perform on those assets. Like scope, the rules of engagement (ROE) should be determined before the engagement begins — but as most stakeholders will not have been privy to those agreements, the ROE should be included as a section in the report.

The line between what is allowed and what is prohibited is not always black and white; a well-developed ROE will include the conditions under which some actions are permitted. A method used by testers in the U.S. military is to identify Pre-Approved Actions and Pre-Coordinated Actions.

Pre-Approved Actions (PAAs) may be performed by the testing team at any time during the engagement without client coordination. Examples may include:

- External port scanning
- Transmission of phishing emails with non-destructive payloads
- Social engineering efforts to gain access



Pre-Coordinated Actions (PCAs) may be performed by the team once specific conditions are met. These may include:

- Notification of the client's SOC
- Obtaining approval from a client POC
- Performance during specific time frames, such as off-duty hours

In addition to enumerating what you were allowed to do, your report should identify any specific restrictions placed on your efforts. These may include specific assets that are off-limits, prohibitions of certain attacks (e.g. DoS on production servers), etc.

What is most important from a report perspective is to ensure that your ROE description is logically organized. The list of approved and prohibited actions may become excessively long for inclusion in the main body of your report. One technique is to limit the content included in this section to restricted activities while providing the approved activities list as an appendix

Attack Narrative

The attack narrative is simply the story of how you achieved whatever level of access you obtained. This section is your opportunity to be a storyteller ... but not in the same way you might do over adult beverages at DefCon. A professional narrative starts with your methodology. How did you conquer and move on from each phase of the attack life cycle? The attack narrative should iteratively answer the following questions for each phase:

- How did I survey the environment?
- What did I discover during the survey?
- Why did I choose to attempt an action?
- If the action failed and caused us to re-assess the strategy, why?
- If the action succeeded, why?
- What did I gain from the action? Did I move into a new phase?

You will need to get somewhat technical to provide your narrative effectively, but this isn't the place for calling out specific CVEs, providing code snippets or dropping screen shots — those

will be included later in the detailed findings. When you are done writing, it should read like a story that someone reasonably familiar with security can follow — and enjoy!

This section is an excellent place to highlight existing controls that caused your team headaches. If you hit a wall that you were never able to overcome, use this opportunity to highlight all the hard work you did in your attempt to move the ball. It will provide your client with positive feedback on their defensive efforts and demonstrate your diligence as a testing team.

Summary of Findings

The goal of this section is to be the quick reference of all items that need to be remediated. This section is likely visited more often after the items are in process, weeks or even months after initial report delivery. Ideally, you would deliver your final report electronically through a platform like PlexTrac, and your client could immediately dive into their remediation workflow of assigning tasks to analysts and tracking their progress in real-time. But if you do not have that luxury, you must keep a few things in mind when presenting your summary of findings.

Many of your clients will take your findings and manually copy and paste them into a workflow tool like Jira or ServiceNow. Perhaps they are still living in 1998 and opt to create an Excel spreadsheet for task assignment and tracking. In either situation, they will want a single place to get the minimum information needed to generate a ticket:

- Finding title
- Finding severity
- Numbering / Unique ID system

If your test includes multiple flavors (e.g. External Network, Web App, etc), you will want to logically group and sort your findings. You should assign a numbering convention to your findings here to provide the responsible analyst a reference to the full detailed finding later in the report. Using a tool like PlexTrac can ease this process by automatically generating the numbering in a custom export template, ensuring that last-minute changes don't result in mindless renumbering.

In the example shown, shading is used in the title row and the numbering column. If shading colors are used, they should be congruent with the color palette used for your branding throughout the report.

Detailed Findings

And now we are finally on to the “red meat” of the report – the

detailed findings. Now is your opportunity to drop all the details on each vulnerability you found, arming your client with all the knowledge they need to begin effective remediation. This seems like a great place to use a table to organize all this data, right?

| Finding # | Severity | Finding Title |
|-----------|---------------|--|
| 1 | Critical | Default or Guessable SNMP community names: public |
| 2 | High | WebSocket hijacking (reflected DOM-based) |
| 3 | Medium | Open redirection (stored) |
| 4 | Low | Cookie manipulation (reflected DOM-based) |
| 5 | Informational | CIS 16.4: Encrypt or Hash all Authentication Credentials |

No.

Tables are visually appealing, and they do provide a tool for systematically displaying data in a way that is easy to comprehend. But they suffer from one huge drawback.

Ever try to copy the text in a cell? Sure, you can do it ... if you are really, really careful, using a 19,000 dots-per-inch gaming mouse and didn't have more than two beers the night before. Trying to copy and paste out of a table can be a horrific experience. Inevitably, you will highlight the entire cell (and usually adjacent cells as well) — not just the contents — ruining your ability to CTRL+C and forcing you to reposition the mouse and try again. And again. And again. Side note: if you have enlightened clients who are using PlexTrac to receive, remediate, and track their vulnerabilities, then this isn't an issue.

Tables are wonderful for providing visual cues. Generally, when we see a new table, we know that we have moved on to the next thing — whatever that thing is. So we propose a middle ground. We suggest using tables to display the metadata of a finding: the finding number, the finding title, the severity, the CVSSv3 score, etc. With the exception of the title, your client won't be doing a lot of copying-and-pasting of those items. But when it comes time to provide the narrative components of a finding, you should ditch the table. Your clients will thank you when they begin the cut and paste.

To be perfectly clear, this is the layout we propose you use to display detailed findings: A small, two to three line table that includes only metadata, followed by a text display of the associated narrative fields. The small table will still be there to provide eye candy and a visual cue that a new finding is being displayed, but the client can now easily copy and paste from fields like Description and Recommendations as in the example shown here.

The content in the above example is abbreviated for demo purposes, and you probably won't want to use the pumpkin-spice color theme. But the concept is sound — use tables sparingly when presenting data that will likely be copied, and don't be afraid of text.

But enough on layout — let's talk content. Your target audience for this section is the analyst who will remediate the vulnerability, but other stakeholders will certainly read some findings — likely those rated with a critical or high severity.

Description

The findings field that you will need to most tailor for a diverse audience is the “Description,” or whatever title you choose for the portion where you narrate the issue.

One method for structuring this field is to mentally view it as three subsections:

Vulnerability Overview

Provide 2-3 sentences that summarize the vulnerability and potential impact. Basically you want to clearly identify what the issue is and why it's a problem. The language should be

| Finding # | Risk Rating | External Findings |
|-----------|-------------|---|
| 1 | HIGH | Server-side Template Injection Exploitable on www.contoso.com/login |

Description:
The template available at [www.contoso.com/login](#) allows attackers to inject commands to the server. Using a series of injected commands, our testers were able to obtain unauthorized root access to the server.
Server-side template injection occurs when user input is unsafely embedded into a server-side template, allowing users to inject template directives. Using malicious template directives, an attacker may be able to execute arbitrary code and take full control of the web server. The severity of this issue varies depending on the type of template engine being used. Template engines range from being trivial to almost impossible to exploit. The following steps should be used when attempting to develop an exploit....

(Exhibit 3. You Don't Want To do This...)

```
function generateInputField($type, $name, $value = '') {  
    return '<input type="' . $type . '" name="' . $name . '" value="' . $value . '" />';  
}  
  
function generateForm($method, $action) {  
    return '<form method="' . $method . '" action="' . $action . '" />';  
}  
  
function setTitle($title) {  
    return '<title>' . $title . '</title>';  
}  
  
function generateButton($text, $type) {  
    return '<button type="' . $type . '">' . $text . '</button>';  
}  
  
if ($logged_in == false) {  
    $html .= '<div class="login-form">';  
    $html .= generateForm('POST', 'login.php') . generateInputField('text', 'user', getUserNameFromCookie()) .  
    generateInputField('password', 'pass') . generateButton('login', 'submit') .  
    '</div>';  
}  
echo $html;
```

CVSSv3 Score: 8.7

Affected Assets:
[www.contoso.com/login](#) hosted at 10.12.23.34, 12.34.45.56

Recommendations:
Wherever possible, avoid creating templates from user input. Parsing user input...

References:
CVE-2018-14716

understandable by a non-technical person with some mild help from Google for specific terminology if needed.

“The customer portal login page at <https://www.contoso.com/portal> allowed our testers to gain unauthenticated access through the use of SQL injection. When accomplished in conjunction with valid user names obtained through open-source research, we were able to view account details and had the required access to make account detail changes.”

Vulnerability Education

Here your goal is to educate the person who will be remediating the issue. You should assume that the person is a junior security analyst or mid-level IT engineer. The reader may have some familiarity with the class of vulnerability but lacks your team’s level of expertise. This is an opportunity to not only give them the knowledge they need to fix this issue today but also to prevent future recurrences. This subsection should be generic — it should provide the background education necessary to understand the context of the finding, but it should not directly address the specific instance in the client environment. Why? Because you want to reuse this portion every time you encounter the same vulnerability! PlexTrac allows you to easily save any finding into the WriteupsDB, so on the next engagement when you encounter the vuln again, you can simply import it into the new report.

You will need to polish the first and last subsections of the description to reflect the specifics of that environment, but you will have saved yourself a ton of time researching and paragraphs of writing. You should feel free to make this area as long as necessary; three to four paragraphs is not uncommon. An example of the start of this subsection:

“Insecure Direct Object Reference (IDOR), also known as Forced Browsing, is a class of information disclosure vulnerabilities that allow an unauthorized user to bypass access controls through the manipulation of certain parameters.”

Vulnerability Technical Specifics

This last subsection is written squarely for the analyst performing the remediation. Now that they are armed with the contextual education, you can dive into the technical aspects of the environment and how it was exploited. Using this subsection as a guide, a junior analyst should be able to recreate the exploit step-by-step. This is also the place to include screenshots and code samples as corroborating artifacts. Better yet, if using a platform like PlexTrac, you can include videos that show you executing the exploit. If a picture is worth a thousand words, a video can be worth a thousand screenshots!

Score

Providing an industry-standard scoring to a vulnerability demonstrates objectivity in your severity ranking for a given finding. Automated scanning tools normally provide such a score, such as Common Vulnerability Scoring System version 3, and PlexTrac will automatically import these scores into the appropriate fields. But you should also consider scoring for vulnerabilities discovered through manual testing. Industry scoring systems can seem daunting due to the number of variables included, however tools are freely available to simplify this process such as that found at <https://www.first.org/cvss/calculator/3.1>. With a little bit of analysis and a few clicks in a browser, you can greatly enhance the professionalism of your report.

The screenshot displays the 'Base Score' configuration interface in PlexTrac. At the top right, a yellow badge shows the calculated score: **6.8 (Medium)**. Below this, the interface is organized into two columns of input fields, each with a title and several selectable options.

| Field | Selected Option | Other Options |
|--------------------------|-----------------|---------------------------------------|
| Attack Vector (AV) | Network (N) | Adjacent (A), Local (L), Physical (P) |
| Attack Complexity (AC) | Low (L) | High (H) |
| Privileges Required (PR) | Low (L) | None (N), High (H) |
| User Interaction (UI) | Required (R) | None (N) |
| Scope (S) | Unchanged (U) | Changed (C) |
| Confidentiality (C) | Low (L) | None (N), High (H) |
| Integrity (I) | Low (L) | None (N), High (H) |
| Availability (A) | High (H) | None (N), Low (L) |

Affected Assets

Anyone who reads the description should have a clear understanding of what assets are impacted by this finding, but it is helpful to have this data displayed discreetly for easy reference. Note that the asset type may vary considerably; you may have domains, subdomains, individual hosts, subnets — or even people in a social engineering engagement!

Automated scanning tools will normally include the affected asset as a field in their report; however, many, such as Nessus, will create an individual finding per asset. We recommend grouping the assets together under a single vulnerability to eliminate confusion and allow for easier reading. Then in your analytics and ticketing systems, you may split the tickets by asset depending on who is responsible for implementing the fix.

Recommendations

The overarching goal here is to provide methodical, step-by-step guidance on how to mitigate the vulnerability, written to the level of a junior analyst or IT engineer. Line-by-line checklists are always appreciated if command line instructions are included. Again, they will be even more appreciated if you DON'T place them in a table which would make copying and pasting a pain! If a resource exists that is authoritative (e.g. A Microsoft Knowledge Base (KB) article), don't recreate the wheel! Provide the link and be confident that your clients will appreciate you acknowledging the value of expert guidance.

If at all possible, conclude your recommendation with guidance for how to validate or confirm remediation of the vulnerability. If your description subsection is well documented, it may be as simple as stating, "Confirm remediation through re-test using exploit procedures documented in the finding description." Your clients will never be angry at you for including too much guidance as long as it is professionally presented in a clear, correct, and concise fashion.

References

This is not the place for you to simply drop links and hope that your clients will gather the information to fix their problems. This is where you can help your client's junior analyst take their skills to the next level by providing additional background education on the root vulnerability or vulnerability class.

Automated scanning tools will spit out CVEs, CWEs, and occasionally an article or two with nothing more than a listing of their reference number or a URL. You can demonstrate additional value to your clients by selecting references that are tailored to their environment and by simply providing a few words of context before the reference. For example, if the vulnerability was SQL injection and your client's environment includes Drupal 7, the following reference would be appropriate:



**“For information specific to preventing SQL injection in Drupal 7 environments:
<https://befused.com/drupal/sql-injection>”**

When it comes to references, more is not better. Everyone is busy, and you should feel lucky if a client analyst actually looks up any of the references you provide. So prioritize quality over quantity; if a young analyst is happy with a single link provided for one finding, they might just follow another in a different finding.

Conclusions and Future Recommendations

At the end of the detailed findings section, your client may well feel like you slammed them across the face with a 2x4. But you can make them feel a little better by wrapping up everything with a nice bow in your conclusions and future recommendations section. This section is geared to summarize everything you just reported. It's the “tell them what you told them” section.

You should briefly conclude with the activities you conducted to get to this point and finish off with a general idea of how the client's security posture ranks based on the issues identified. Once you've established how bad it looks (or maybe good), then you should summarize the key takeaways and immediate next steps.

This is an opportunity to address more of the global issues that may have been recognized during the engagement. For example, if you identified that multiple systems were missing critical patches, the future recommendation is to improve their overall patch and vulnerability management processes. Regardless, this section is the last true impression you will leave with the client, so you should clarify what their next steps are and what yours will be too. If you offer validation testing or remediation consulting, then this is where you mention how that will be handled or addressed.

For example, in PlexTrac you can collaborate with the client on individual findings through the status tracker. This feature can show the value you provide after the initial delivery and develop that "stickiness" factor. So you may wish to clarify that remediation and validation testing can be coordinated through the status tracker in PlexTrac. Regardless, the client needs to be clear on which balls are in whose court.

Finally, it's best to end with a cordial statement saying that you appreciated working with them and look forward to additional collaboration in the future. After all, your goal is to develop a long term relationship with the client, and here is where you plant those seeds.

Appendices

Consider the appendices as the relief valve for your ego. During the course of the engagement, you have undoubtedly collected a ton of very interesting data and artifacts that any security engineer would drool over. But as we mentioned at the beginning of this paper, you have a diverse audience of stakeholders. You will lose their interest and generate frustration if you attempt to shoehorn all the data you collected into the main body of the report.

It's perfectly okay to not include everything you learned on an engagement in the main body of the report. In fact, we don't recommend that you do. Your client has a limited attention span,

driven by conflicting priorities and limited budget. If you drown them in information, the critical items may be lost in the noise. During the course of your engagement, you will develop an understanding of your client's capacity to remediate findings. You can and should make professional decisions about how much to put on their plate at one time. If you know that they won't be able to fix all the critical and high severity issues within the next year, is it prudent to include 50 informational findings in the main body?

Appendices are a lovely tool for still providing the client with useful information, but segregate it from what you really want them focusing on today. Because there is an explicit understanding that items in the appendices are optional, you have freedom to let your educational spirit fly. That being said, most people aren't going to determine where the appendices begin when they hit the print button. Don't irritate your clients by including pages upon pages of information of minimal value.

So what are candidates for inclusion in an appendix? Consider the following:

- Detailed finding information for those findings classified at a severity of “informational”
- Biographies and certifications of the testers assigned to the engagement
- Detailed information on your chosen formal methodology
- Lengthy artifacts that would otherwise interrupt the flow of a finding
- Complex risk calculations that are not based on industry standards



This All Seems Like a Lot of Work

Writing a robust and professional penetration test report is not easy, nor should it be given the financial investment your clients are making. For the most part in this paper, we have focused on the brain work — the things you need to communicate effectively and methods for doing so. The report is the most important deliverable for a pentest and it's important to get it right.

It's important to make it look professional, but you need to consider the manner in which your client will interact with the report as well. For a static document (Word/PDF), they will likely be copying and pasting information to another tool. Have you made it easy to do that copy/paste operation? How do they get screenshots and exhibits into those other systems? Do they have a good place to show progress on fixing their issues compared to the original findings?

These are all important considerations as you continue to write and deliver your reports. So yes, it is a lot of effort, but worth the investment.



The report is the most important deliverable for a pentest and it's important to get it right.

Final Thoughts

While all of what we just described is a lot of work, we will conclude with a shameless plug. Unfortunately, for many pentesters too much time is consumed with searching, formatting, adding graphics or visuals, and copying/pasting during the report writing process. Men and women who have spent the majority of their adult lives becoming experts in information security are spending countless hours fighting the very productivity tools that were designed to empower them. Furthermore, their clients don't always know what to do with a large document of reported findings.

Don't get us wrong, we love Microsoft Office products; in fact, we can't imagine a modern business operating without them. But just as we all know that Microsoft Excel is inadequate

for tracking financials and operational ticketing, we must acknowledge that we need custom tools for performing the complex tasks associated with managing our cybersecurity assessments and programs.

Enter PlexTrac.

PlexTrac was built by pentesters to ease the pains we all feel when the fun part is done and the report must be written, delivered, and tracked. We don't claim to make the process easy — you still need to provide the analytical brainpower. But let PlexTrac help reduce the time spent on mindless and tedious tasks. We all deserve tools that empower us to focus on our skills and remove barriers for ultimate efficiency.

Visit us at www.plextrac.com to learn how PlexTrac can help your organization focus more time on the things that bring value to your customers.

PlexTrac is the premier penetration test reporting, collaboration, and management platform designed to automate planning, documentation, communication, and remediation tracking, allowing service providers to enhance margins and client outcomes and enterprises to demonstrate the value of internal pentesting efforts and improved security posture.

www.plextrac.com

© 2023 PlexTrac, Inc., all rights reserved.

